

Auf der Reise zum fertigen Softwarepaket

06.12.2025

Zwischenbericht Team 3:

Leonel Konla

Patrick Rahn

René Pettelkau

Marty Jabbusch

Leo Ogger

Gliederung

1. Einführung in unser Projektthema
2. Requirements Engineering und Use-Case-Modellierung
3. UML-Modellierung und Design
4. Implementierung und Versionsverwaltung
5. KI-Werkzeuge im Entwicklungsprozess
6. Herausforderungen und Lösungen

Einführung

BHT

Berliner Hochschule
für Technik

Studiere Zukunft

Projektidee und Zielgruppe

- Fitnessstracker hilft Nutzerinnen und Nutzern dabei, ein ausgewogenes und gesundes Leben zu führen
- Soll Aktivitäten erfassen und Trainingspläne erstellen können
- Ziel: Gewohnheiten sichtbar zu machen, Motivation zu fördern und langfristig die gute Balance zu ermöglichen

Zielgruppe:

Menschen mit erhöhtem Gesundheitsbewusstsein und –bedarf

Technologiestack

Komponente	Technologie	Begründung
UI	JavaFX	Pures Java, einfach umsetzbar, gut für Desktops
Datenbank	SQLite	ohne Installation nutzbar, einsteigerfreundlich
Build-Tool	Maven	Einsteigerfreundlich und gut dokumentiert
Testing	JUnit	De-facto Standard für Java

Geplanter und realisierter Funktionsumfang

- Plan:
 - Nutzbares System
 - 2 oder 3 funktionierende Use-Cases
 - Intuitives User-Interface
 - Anbindung Datenbank
 - Testing
- Realisiert:
 - Grundkonstrukt der Anwendung
 - Einloggen Use-Case
 - Code für Testing geschrieben

Requirements Engineering und Use- Case-Modellierung

BHT

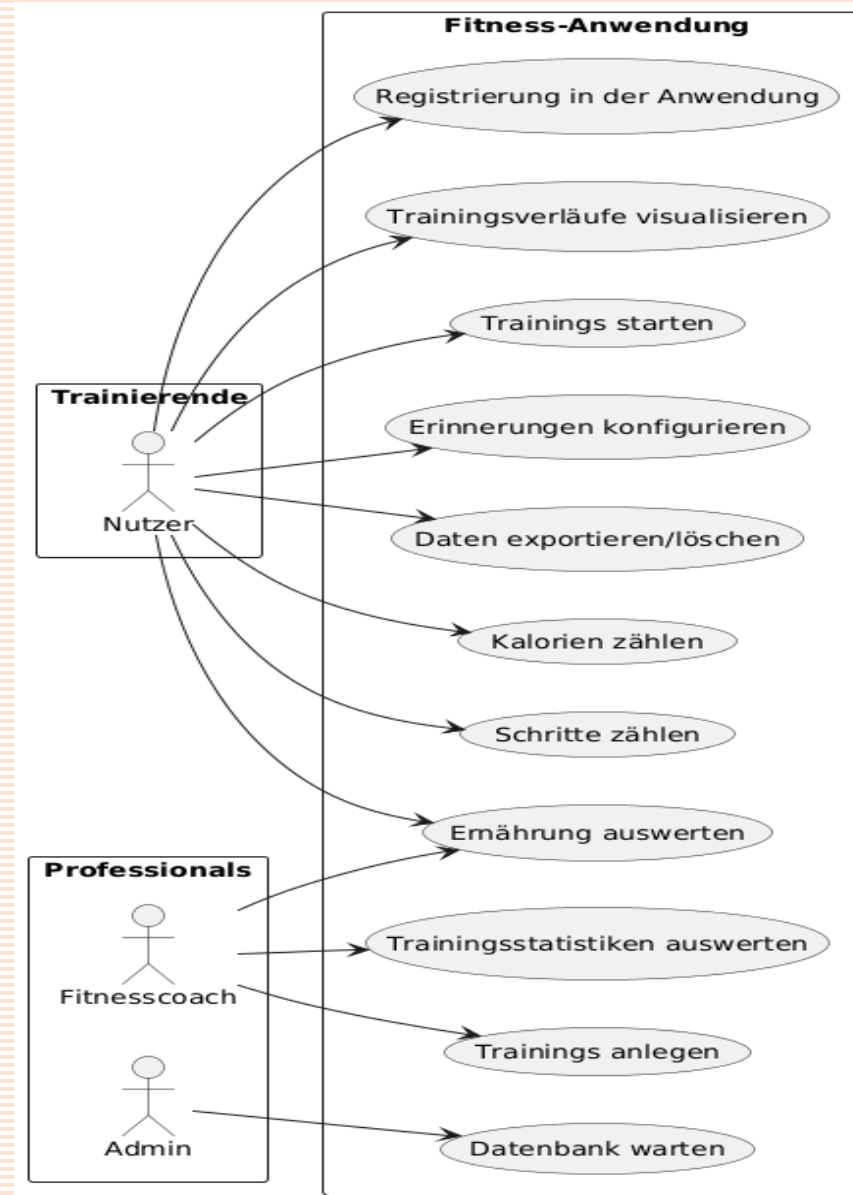
Berliner Hochschule
für Technik

Studiere Zukunft

Requirements Engineering

ID	Anforderung	Priorität	Abnahmekriterium
F1	Nutzer kann ein Konto anlegen (Registrierung) und sich anmelden.	A	Registrierung, Login und Logout funktionieren; Daten sicher gespeichert.
F2	Nutzer kann tägliche Werte zu Bewegung, Ernährung, Schlaf, Wasser und Erholung eingeben.	A	Eingaben validiert, gespeichert und Datum korrekt zugeordnet.
F3	System berechnet automatisch den Life-Skill-Score anhand der täglichen Eingaben.	A	Score-Berechnung liefert reproduzierbare Werte nach definierter Formel.
F4	Dashboard zeigt aktuellen Score und Statusanzeige (z. B. Ampelfarben).	A	UI zeigt Score korrekt an; Anzeige aktualisiert sich nach Eingaben.
F5	Nutzer kann Verlauf und Trends (7/30 Tage) visualisieren.	A	Diagramm zeigt korrekte Aggregation; Werte mit Zeitreihe übereinstimmend.

Use Case Diagramm



Zentrale Use Cases

UC-01: Nutzer registrieren

- Kurzbeschreibung: - Neuer Nutzer möchte sich im Programm registrieren
- Akteur(e): - Unregistrierter Nutzer
- Auslöser: - Nutzer hat Programm installiert und möchte sich registrieren
- Vorbedingungen: - Nutzer hat Programm installiert
- Registrierung ist verfügbar
- Normalablauf: - Nutzer öffnet Programm
- System zeigt Registrierungsformular
- Nutzer wählt Mailadresse und Passwort
- System prüft, ob Mailadresse bereits vergeben wurde
- Nutzer bestätigt Eingabe
- System speichert Daten und zeigt Bestätigung
- Ergebnis: - Nutzer ist im System registriert und kann sich anmelden
- Alternative Abläufe: - Nutzer bricht ab → System verwirft Eingabe
- Mailadresse bereits vorhanden → Nutzer kann andere angeben

Zentrale Use Cases

UC-02: Schrittzähler

- Kurzbeschreibung: - Der Nutzer möchte seine täglichen Schritte protokollieren
- Akteur(e): - Registrierter Nutzer
- Auslöser: - Nutzer hat Programm geöffnet und wählt den Punkt „Schrittzähler“
- Vorbedingungen: - Nutzer ist eingeloggt
- Anzahl der am Tag bewältigten Schritte ist bekannt
- Normalablauf: - Nutzer öffnet Programm
- System öffnet Startansicht
- Nutzer wählt „Schrittzähler“
- Nutzer gibt Anzahl der Schritte an
- Nutzer bestätigt Eingabe
- System speichert Daten und zeigt Bestätigung
- Ergebnis: - Nutzer kann Schrittzahl protokollierter Tage auswerten
- Alternative Abläufe: - Nutzer gibt keine Schritte an → System speichert leeren Tageswert
- Nutzer möchte vergessene Werte ergänzen → Nutzer kann Tag auswählen und Daten ergänzen

Methodik

Wie haben wir unsere Use-Cases ermittelt?

1. Zum einen aus eigener Erfahrung, also persönlicher Erfahrungen unserer Teammitglieder
2. Zum anderen mit Hilfe von KI, um auch einen Blick über unsere eigenen Erfahrungen hinaus zu erhalten

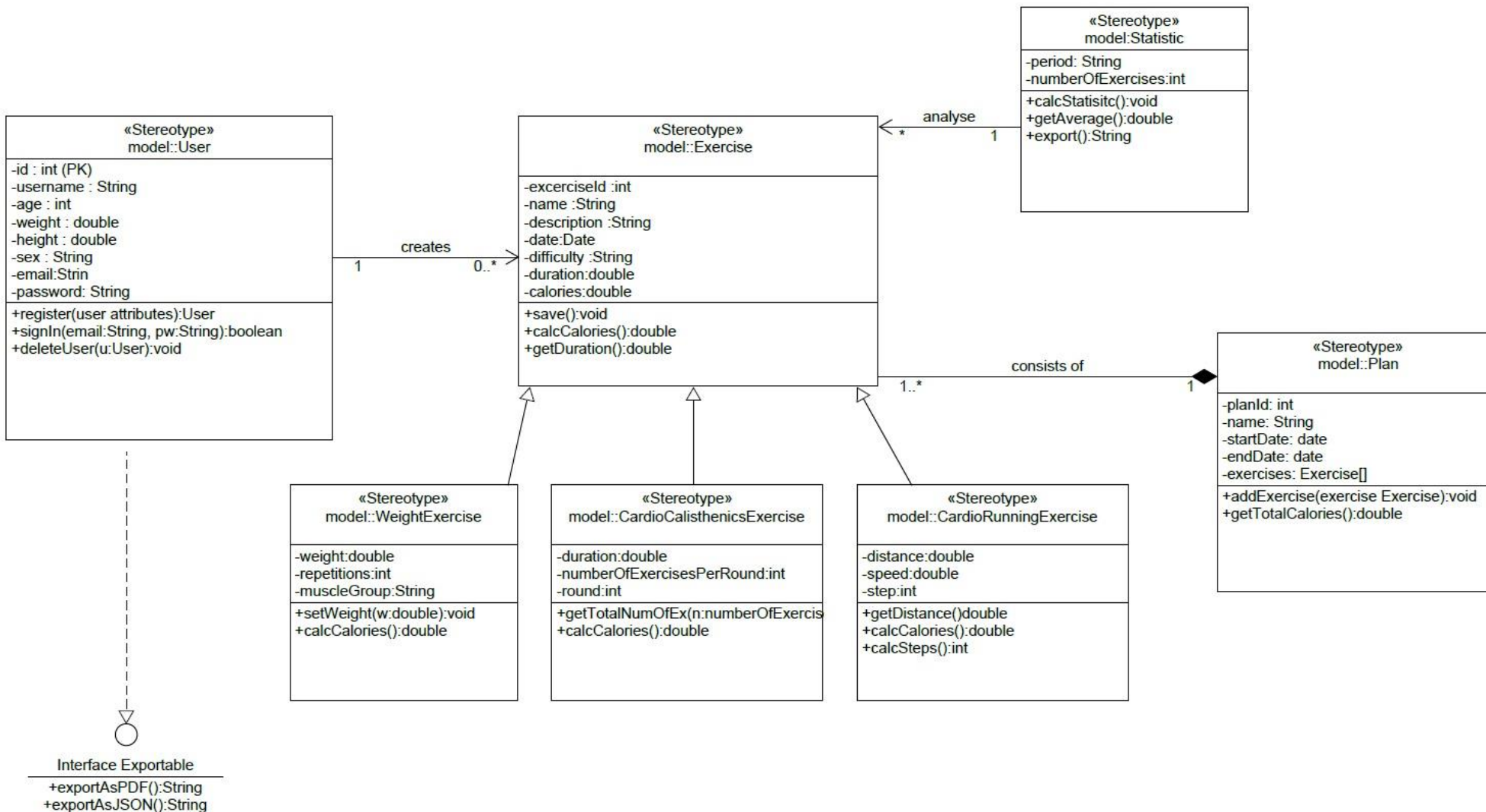
UML-Modellierung und Design

BHT

Berliner Hochschule
für Technik

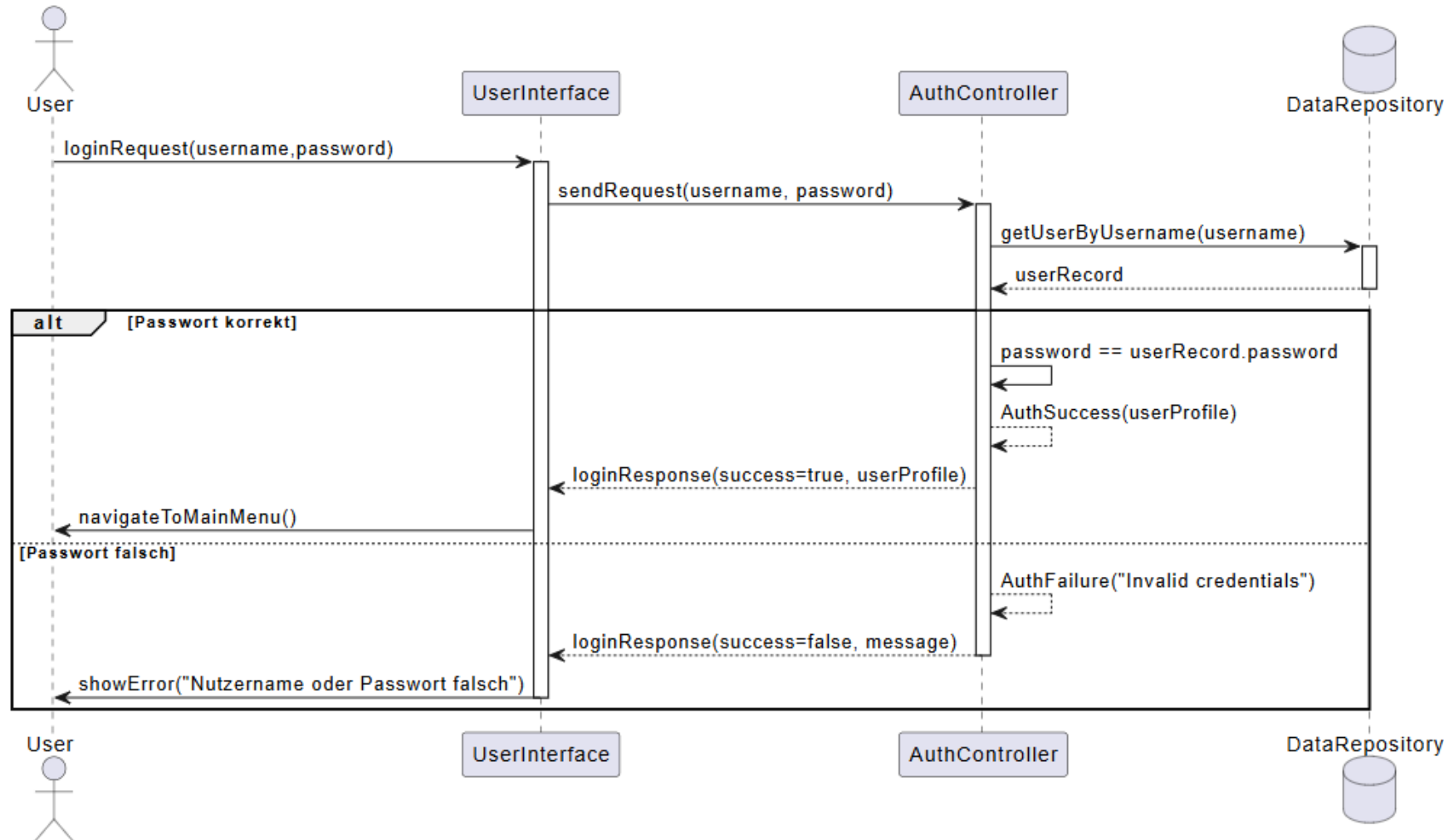
Studiere Zukunft

Klassendiagramm



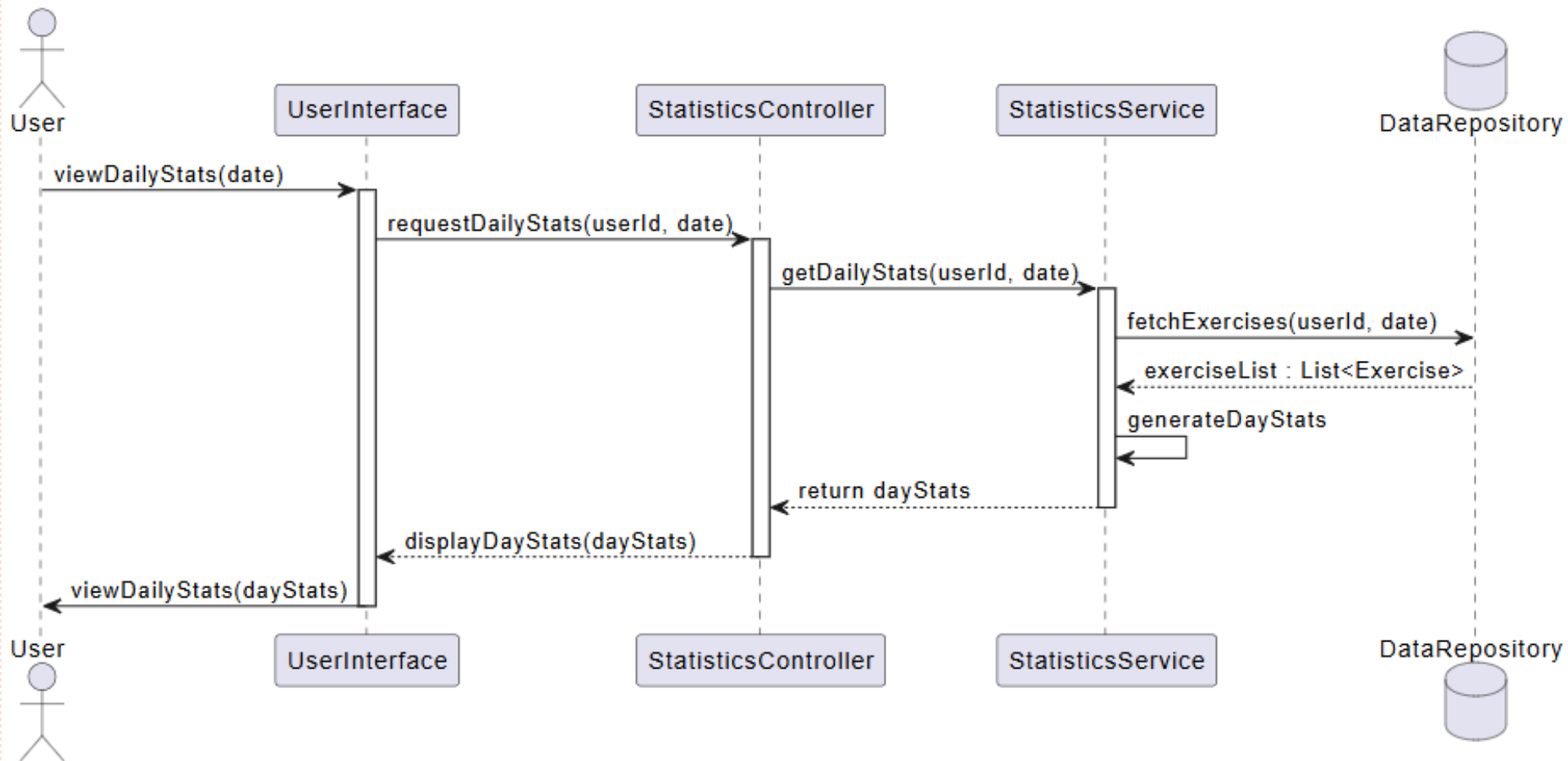
Sequenzdiagramme

Sequenzdiagramm 1 - Einloggen eines Nutzers



Sequenzdiagramme

Sequenzdiagramm 2 - Abruf einer Tagesstatistik



Implementierung und Versionsverwaltung

BHT

Berliner Hochschule
für Technik

Studiere Zukunft

Projektstruktur, Codebeispiel, Git-Workflow und Testing

KI-Werkzeuge im Entwicklungsprozess

BHT

Berliner Hochschule
für Technik

Studiere Zukunft

Einsatzbereiche

- Aufsetzen von Git
- Nutzung der Git-Befehle
- Java FX bei IntelliJ einrichten
- Coding
- Nachfragen zum Inhalt des Moduls
- Stakeholderansicht
- Use Case Empfehlungen
- Funktionsumfang

Promptbeispiele

- Promptbeispiel Funktionale Anforderung:

"Stell dir vor, du wärst ein 45 jähriger Fitnessenthusiast und benutzt eine neue Fitness App. Welche Funktionen wären für dich wichtig?"

- Promptbeispiel Requirements Engineering:

"Du bist ein erfahrener Requirements Engineer. Ich entwickle eine Fitness-Tracking-App, die nicht nur die sportlichen Aktivitäten einer Person, sondern auch dessen Wohlbefinden im Allgemein umfasst. Ernährungs-Tracker-Scores sollen die Anwender motivieren, gesunde Gewohnheiten zu entwickeln, Routinen zu verbessern und eine nachhaltige Balance zwischen Körper und Geist zu finden. Finde 8 funktionale Anforderungen (Was soll das System tun?) und 6 nicht-funktionale Anforderungen (Wie soll das System arbeiten?)."

Bewertung der KI-Nutzung

- Projektalltag:
 - Dokumente und Meetingnotizen zusammenfassen
 - Fachliche Erläuterungen
 - Ideenimpulse
 - Bewertung und Verbesserung der eigenen Arbeit
- Risiko:
 - Vertrauliche Projektdaten an Dritte
 - Unvollständige oder missverständliche Prompts

Herausforderungen und Lösungen

BHT

Berliner Hochschule
für Technik

Studiere Zukunft

Technische Herausforderungen

- Git: allgemeine Nutzung, Commits, Branching Strategien
- Umlet zur Gestaltung der UML-Diagramme
- Maven:
 - Building Tool
 - Dependency Manager
 - Project Manager
- Lösungsansätze:
 - Schrittweise Erarbeitung
 - Von Null auf anfangen

Teamorganisation

- Heterogenes Team: Alter, Familienstand, Arbeitsverhältnis
- Wenig überschneidende verfügbare Zeit
- Keine nennenswerte Programmiererfahrung
- Lösungsansätze:
 - Wöchentliches Meeting nach der Vorlesung
 - Aufgaben nach Stärken und Interessen verteilt

Danke für eure Aufmerksamkeit!