

# Dokumentation

Abdullah Alrefai: 677243 | Julian Ludwig: 677202 | Joanis Palaskas: 677208

Projektname: Modern-Walking



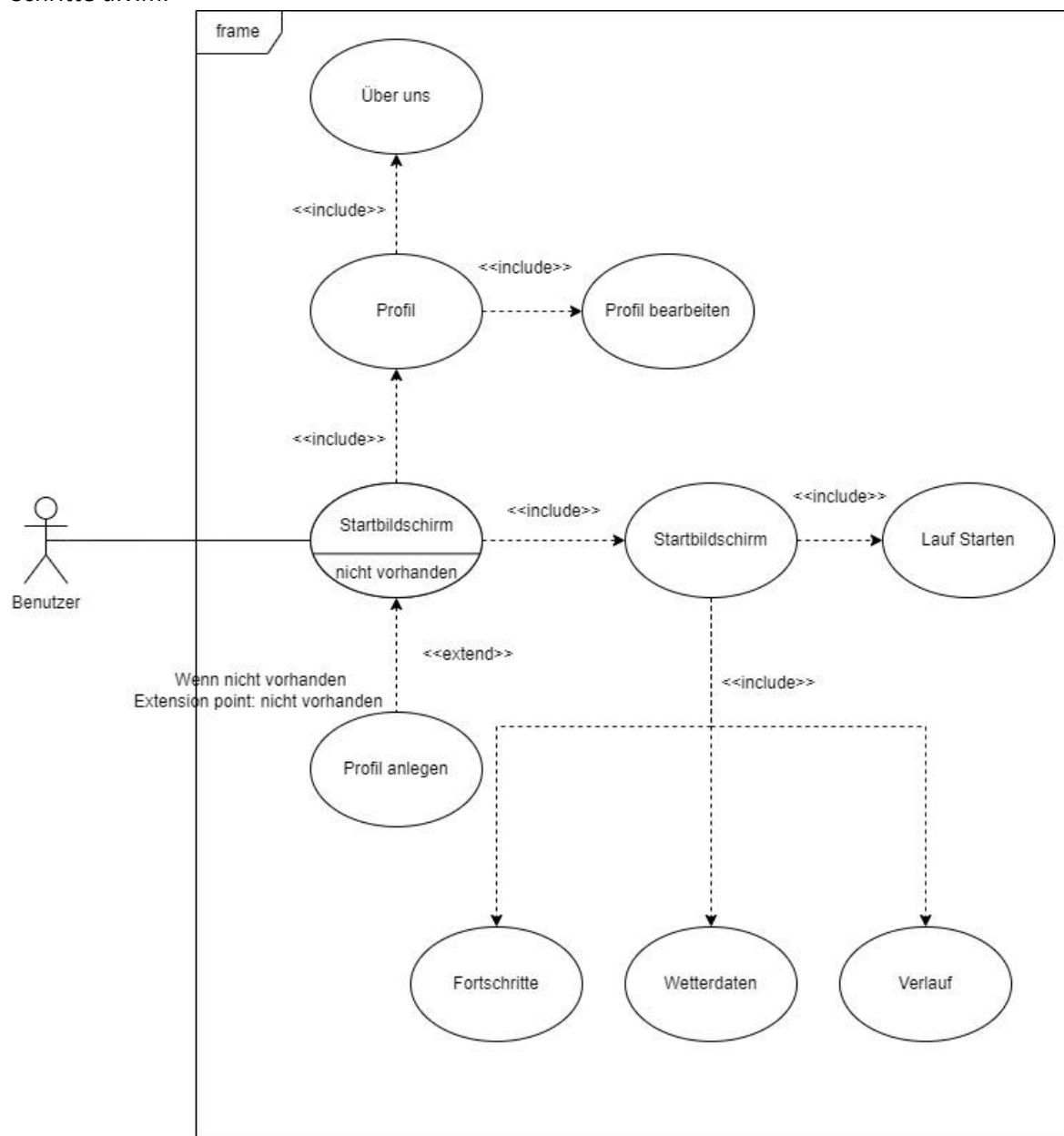
# Inhalt

Ziele und Inhalt des Projektes.....	3
1. .... Funktionalität	4
1.1. .... Profil Anlegen	4
1.2. .... Lauf Starten	4
1.2.1. .... Standortdaten auslesen und anzeigen	4
1.3. .... Laufdaten speichern und Anzeigen	4
1.4. .... Grafische Darstellung der Daten	5
1.5. .... Profil Seite	5
1.6. .... Profil bearbeiten	5
1.7. .... Wetterdaten	5
2. .... Projektarchitektur	6
3. .... Projekt-Package-Struktur	7

## Ziele und Inhalt des Projektes

Das Ziel des Projekts war das Erlernen wichtiger Fähigkeiten und Kenntnisse in der App Entwicklung, um diese in einer praktischen Abgabe anwenden zu können. Hierbei wurde eine Android App anhand des Erlernten sowie einiger Vorgaben konzipiert und entwickelt.

Die App mit dem Titel „Modern Walking“ soll durch spielerische Mechaniken und Visualisierungen für Motivation beim Spaziergehen sorgen. Laut der Weltgesundheitsorganisation sind zweieinhalb bis fünf Stunden moderater Bewegung für einen gesunden Lebensstil empfohlen. Mit „Modern Walking“ sollen Anwender Ihre Spaziergänge protokollieren können, um wichtige Zusatzinformationen zu erhalten, wie: die zurückgelegte Strecke, verbrannte Kalorien, gelaufene Schritte u.v.m.



# 1. Funktionalität

## 1.1. Profil Anlegen

Wenn der Nutzer zum ersten Mal die Applikation startet, wird er aufgefordert ein Profil anzulegen. Bei der Angabe muss der Benutzer sein Geschlecht, Körpergröße, Gewicht, Alter und Name angeben. Die Daten werden in einer Datenbank gespeichert, welche die restlichen Seiten der Anwendung mit diesen Informationen versorgt (Singleton).

## 1.2. Lauf Starten

Wenn die Berechtigung für die Standortdaten gegeben sind, wird durch das Drücken auf den "Start" Knopf der Startseite nicht erneut nach der Berechtigung gefragt, sondern der Lauf gestartet. Der gewünschte Lauf kann auf dieser Seite erstellt werden, indem die Distanz, Dauer oder Kalorien, die verbrannt werden sollen, festgelegt werden. Genauer zu den Läufen wird in den nächsten Schritten detaillierter erläutert.

### 1.2.1. Standortdaten auslesen und anzeigen

Nachdem der Benutzer der Anwendung die Rechte zur Erfassung der Standortdaten erteilt hat, kann die Applikation das Smartphone bis auf wenige Meter direkt orten. Diese Standortdaten werden für den Benutzer auf einer Karte angezeigt, damit er sich bei seinen Spaziergängen besser orientieren und das Ziel einfacher finden kann.

Als Grundlage dient der FusedLocationProvider welcher durch die Google Play Dienste bereitgestellt wird.

## 1.3. Laufdaten speichern und Anzeigen

Nach einem abgeschlossenen Lauf werden die Daten des Laufs gespeichert und in einer Karte auf der "Verlauf" Seite angezeigt. Auf der "Verlauf" Seite kann der Anwender seine abgeschlossenen Läufe einsehen, es werden die verbrannten Kalorien, Dauer, Distanz, Schritte, durchschnittlicher Puls und die Geschwindigkeit in km/min angezeigt.

Der Benutzer kann durch nach oben Wischen die Karten mit den Laufdaten nach seinen vergangenen Läufen durchsuchen. Für eine bessere Übersicht ist die „Verlauf“ Seite auch für den Landschaftsmodus angepasst.

## 1.4. Grafische Darstellung der Daten

Die abgeschlossenen Läufe können in grafischer Darstellung auf der “Fortschritte” Seite angezeigt werden. Hierbei werden die letzten sieben Läufe (sofern vorhanden) in einem Balken Diagramm dargestellt, zusätzlich wird auch ein Durchschnittswert angezeigt. Der Benutzer kann selbst auswählen nach welchem Wert die Daten dargestellt werden sollen, zur Auswahl stehen: Distanz, Kalorien, Zeit und Schritte.

Für die Darstellung des Balkendiagramms wurde eine Library verwendet (<https://github.com/tehras/charts>)

## 1.5. Profil Seite

Auf der “Profil” Seite kann der Nutzer sein aktuelles Level, Profilbild und seine Erfahrungspunkte sehen. Der Levelfortschritt bis zum nächsten Level wird in einer kreisförmigen Fortschrittsanzeige dargestellt, in der Mitte dieser Anzeige wird der Levelfortschritt in Prozentzahlen angezeigt. Ein neues Level wird für jeweils 100 gesammelte XP erreicht. Des Weiteren kann der Anwender sein Profil bearbeiten, den Dark Mode für die Anwendung ein oder ausschalten, die Sprache der Anwendung ändern und mehr über die Entwickler erfahren. Diese Funktionen werden über die Links in der unteren Hälfte der Profil Seite ermöglicht.

## 1.6. Profil bearbeiten

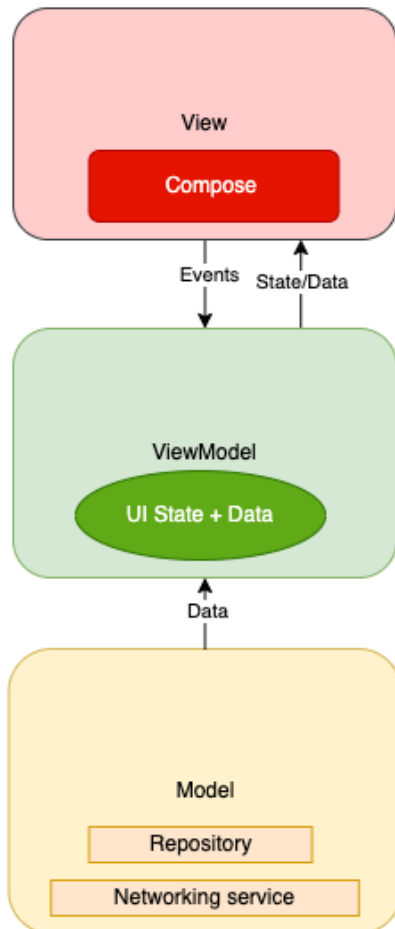
Falls der Nutzer seine angegebenen Daten überarbeiten muss, kann er dies auf der “Profil bearbeiten” Seite tun. Dort hat er die Möglichkeit den Benutzernamen, das Alter, Höhe und Gewicht zu ändern. Darüber hinaus kann er durch Drücken auf die runde Karte in der oberen Hälfte des Screens kann er ein neues Profilbild aus seinem lokalen Speicher auswählen. Durch das Drücken des “Speichern” Knopfes werden die geänderten Werte übernommen und in die Datenbank gespeichert, falls die Seite über den “Abbrechen” Knopf verlässt, werden die geänderten Werte verworfen.

## 1.7. Wetterdaten

Auf der “Wetter” Seite wird in der oberen linken Ecke des Bildschirms der Standort angezeigt. In der Mitte des Bildschirms wird das aktuelle Wetter an dem obigen Standort angezeigt. Am unteren Rand der Seite findet sich eine detailliertere Ansicht des Wetters, das am heutigen Tag am Standort zu erwarten ist.

## 2. Projektarchitektur

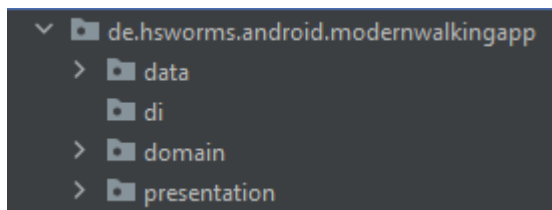
Für die App wird das MVVM-Pattern verwendet. Ein Hauptbeweggrund hierfür waren die Vorteile für die kollaborative Zusammenarbeit und die Wartbarkeit durch die klare Trennung von Model, View und Viewmodel. Ein zusätzlicher Vorteil ergibt sich durch die Möglichkeit der Code Wiederverwendung nach dem DRY Prinzip, wodurch Views auf die gleichen Komponenten zurückgreifen können, sofern nötig. Aus der Einhaltung des DRY Prinzips in Verbindung mit dem MVVM-Pattern ergibt sich auch eine hohe Kohäsion der einzelnen Komponenten, wodurch die App in vielerlei Aspekten und durch wenig Anpassung erweiterbar ist.



Der Model-Layer enthält die Datenquellen, auf die über ein Repository zugegriffen wird. Der ViewModel Layer enthält die States und die App-Logik und gilt als Vermittler zwischen der Datenquellen und der View-Layer. Im der View-Layer (UI) befindet sich den Jetpack Compse-Code, der zur Erstellung der UI dient.

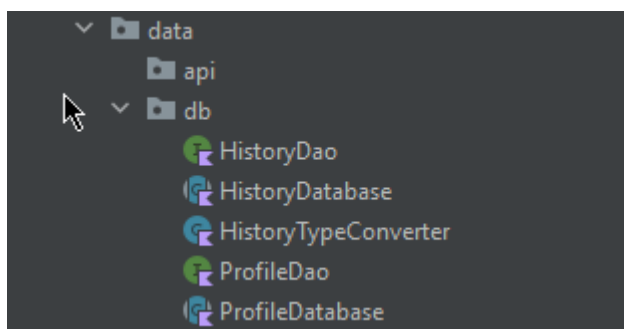
### 3. Projekt-Package-Struktur

Die Projektstruktur hat folgende Layers:

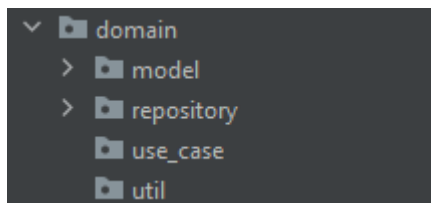


#### **Data-Layer:**

Dieser Layer enthält die Datenbanken und die entsprechenden DAO-Klassen, die in der Repository verwendet werden.



**Domain-Layer:** Enthält die Logik der App sowie die Model-Klassen und die Repositories und die Use-Cases.



**Presentation-Layer:** Enthält alle Screens, Composable-Functions und die Main-Aktivität. In diesem Layer wird die Navigation definiert.

