GPS-Tracking-App zur Aufzeichnung von Outdoor-Aktivitäten

Markus Suchalla
Matrikelnummer: 30355671
Fachhochschule Südwestfalen

Schriftliche Ausarbeitung im Rahmen der Projektarbeit Software Engineering

12. Mai 2025

Inhaltsverzeichnis

1 Einleitung

Im Rahmen dieses Projekts wurde eine mobile Android-Applikation entwickelt, die darauf abzielt, Streckenaufzeichnungen für Outdoor-Aktivitäten wie Fahrradfahren, Wandern oder Spazierengehen zu ermöglichen. Die Anwendung nutzt GPS-Daten, um Routen in Echtzeit zu verfolgen, relevante Aktivitätsdaten wie Distanz, Dauer, Geschwindigkeit und Höhenmeter zu berechnen und diese lokal auf dem Gerät zu speichern. Ziel war es, eine benutzerfreundliche, datenschutzorientierte und funktionale Applikation zu schaffen, die es Nutzern ermöglicht, ihre Aktivitäten visuell darzustellen, detaillierte Statistiken einzusehen und vergangene Strecken jederzeit abrufbar zu machen.

Die Entwicklung konzentrierte sich auf eine intuitive Bedienung, eine performante Umsetzung und die Einhaltung moderner Datenschutzstandards durch lokale Datenspeicherung. Die Applikation wurde so konzipiert, dass sie sowohl für Gelegenheitsnutzer als auch für sportlich ambitionierte Anwender geeignet ist. Im Folgenden werden die Anforderungen, die technische Umsetzung, die Herausforderungen sowie ein Ausblick auf mögliche Weiterentwicklungen detailliert beschrieben.

2 Anwendungsfälle

Die GPS-Tracking-App richtet sich an Nutzer, die ihre Outdoor-Aktivitäten wie Fahrradfahren, Wandern oder Spazierengehen aufzeichnen und auswerten möchten. Die wichtigsten Anwendungsfälle lassen sich wie folgt beschreiben:

- **Streckenaufzeichnung starten:** Der Nutzer startet die Aufzeichnung, wobei die App die GPS-Daten kontinuierlich erfasst.

- **Live-Anzeige von Aktivitätsdaten:** Die zurückgelegte Strecke wird dem Nutzer auf dem Bildschirm angezeigt. Ebenfalls werden Distanz, Dauer, Geschwindigkeit und Höhenmeter in Echtzeit angezeigt.
- **Streckenaufzeichnung pausieren:** Der Nutzer kann die Aufzeichnung manuell pausieren, falls er eine Unterbrechung einlegen möchte.
- **Fotos aufnehmen**: Der Nutzer kann während der Fahrt Fotos aufnehmen, welche mit der Strecke abgespeichert werden.
- **Speicherung der Aktivität:** Nach dem Beenden wird die Strecke lokal auf dem Gerät gespeichert, inklusive aller relevanten Messwerte und Fotos.
- **Abruf vergangener Aktivitäten:** Der Nutzer kann seine bisherigen Aktivitäten einsehen, inklusive Karte, Statistiken und ggf. Fotos.
- **Gesamtstatistiken anzeigen:** Die App bietet eine Übersicht aller aufgezeichneten Aktivitäten mit zusammenfassenden Werten wie Gesamtdistanz oder Durchschnittsgeschwindigkeit. Diese kann er auch auf eine Fahrzeugkategorieëinschränken.

3 Anforderungsanalyse

3.1 Funktionale Anforderungen

Auf Grundlage der Projektzielem, der Produktvorstellung sowie den geplanten Nutzungsszenarion werden die folgenden Anforderungen an die Applikation gestellt:

- **Erfassung und Anzeige von Bewegungsdaten:** Die App muss in der Lage sein, GPS-Daten kontinuierlich zu erfassen und auf einer Karte in Echtzeit darzustellen.
- **Messung relevanter Aktivitätsdaten:** Während der Aufzeichnung sollen wichtige Parameter wie Distanz, Dauer, Geschwindigkeit und Höhenmeter berechnet und angezeigt werden.
- **Steuerbare Aufzeichnung:** Der Nutzer soll die Möglichkeit haben, die Aufzeichnung jederzeit zu starten, zu pausieren oder zu speichern.
- **Lokale Speicherung:** Speicherung aller aufgezeichneten Strecken und zugehörigen Daten lokal auf dem Gerät, um Datenschutz und Offline-Nutzung zu gewährleisten.
- **Abruf vergangener Strecken:** Möglichkeit für Nutzer, gespeicherte Strecken einzusehen und Details wie Route, Statistiken oder verknüpfte Fotos zu überprüfen.
- **Statistische Auswertungen:** Die App soll einfache Gesamtstatistiken über alle bisherigen Aktivitäten bereitstellen, um Fortschritte und Aktivitätslevel nachvollziehen zu können.

3.2 Nicht-funktionale Anforderungen

Neben den funktionalen Anforderungen wurden folgende nicht-funktionale Aspekte berücksichtigt:

- **Plattform:** Entwicklung ausschließlich für Android-Geräte, optimiert für verschiedene Gerätetypen.
- **Performance:** Ressourcenschonende Implementierung, insbesondere hinsichtlich Akku- und Speicherverbrauch.
- **Datenschutz:** Lokale Speicherung der Daten, um die Weitergabe sensibler Informationen an Drittanbieter zu verhindern.
- **Benutzerfreundlichkeit:** Intuitive und einfache Benutzeroberfläche, die eine niedrige Einstiegshürde für alle Nutzergruppen bietet.
- **Kompatibilität:** Einhaltung der Richtlinien des Google Play Stores, um eine Veröffentlichung zu ermöglichen.

3.3 Requirements Diagram

Hinweis: Ein Use-Case-Diagramm oder Datenflussdiagramm zur Visualisierung der Interaktionen zwischen Nutzer, App und Datenbank wäre hier sinnvoll. Eine konkrete Beschreibung oder Datei des Diagramms kann diesen Abschnitt präzisieren.

4 Technische Umsetzung

4.1 Verwendete Programmiersprache: Kotlin

Die Entwicklung erfolgte mit **Kotlin**, einer modernen, von JetBrains entwickelten Programmiersprache, die sich als Standard für Android-Entwicklung etabliert hat. Kotlin bietet gegenüber Java folgende Vorteile:

- **Kürzere Syntax:** Reduzierung von Boilerplate-Code, was die Lesbarkeit und Wartbarkeit verbessert.
- **Null-Safety:** Integrierte Mechanismen zur Vermeidung von NullPointerExceptions erhöhen die Stabilität.
- **Extension Functions:** Flexible Erweiterung bestehender Klassen zur besseren Code-Strukturierung.
- **Interoperabilität:** Volle Kompatibilität mit Java, was die Integration bestehender Bibliotheken erleichtert.

Trotz anfänglicher Unkenntnis der Sprache erleichterte die Ähnlichkeit zu Python und Java den Einstieg. Besonders nützlich waren Features wie **Koroutinen** für asynchrone Programmierung sowie Schlüsselwörter wie `var`, `val` und `fun`, die Boilerplate-Code reduzieren.

4.2 Kartendarstellung mit OSMdroid

Die Kartendarstellung basiert auf der Open-Source-Bibliothek **OSMdroid**, die OpenStreetMap-Daten nutzt. Im Vergleich zum Google Maps SDK bietet OSMdroid folgende Vorteile:

- **Kostenfreiheit und Open Source:** Keine Lizenzgebühren und transparente Nutzung.
- **Datenschutz:** Keine Abhängigkeit zu Google-Diensten, wodurch keine Nutzerdaten an Drittanbieter übertragen werden.
- **Flexibilität:** Umfangreiche Anpassungsmöglichkeiten der Kartenansicht und Funktionen.

OSMdroid ermöglichte eine zuverlässige Darstellung der Strecken und des Live-Standorts, wobei die Performance auch auf älteren Geräten gewährleistet wurde.

4.3 GPS-Ortung mit FusedLocationProviderClient

Die GPS-Daten werden über den **FusedLocationProviderClient** aus den Google Play Services bezogen. Diese Komponente kombiniert verschiedene Ortungsquellen wie GPS, WLAN, Mobilfunkdaten und Bewegungssensoren, um präzise und energieeffiziente Ergebnisse zu liefern. Die Vorteile umfassen:

- **Hohe Genauigkeit:** Intelligente Kombination der Datenquellen auch in schwierigen Umgebungen.
- **Geringer Energieverbrauch:** Optimierte Ortungsintervalle reduzieren den Akkuverbrauch.
- **Regelmäßige Updates:** Kontinuierliche Standortaktualisierungen bei korrekter Konfiguration.

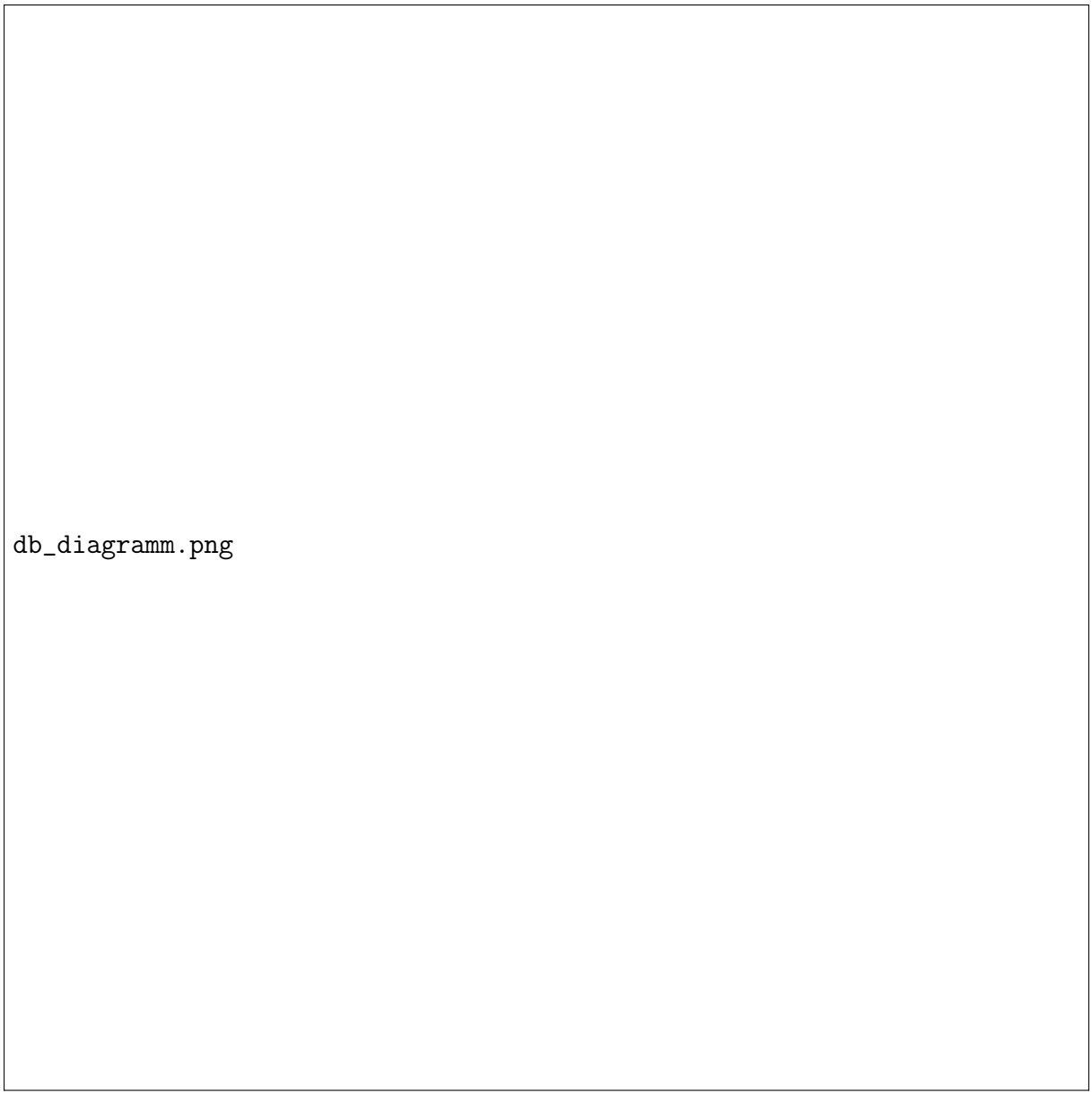
Die Implementierung erforderte jedoch eine sorgfältige Abstimmung, um Genauigkeit und Energieeffizienz zu balancieren.

4.4 Datenbankstruktur mit Room (SQLite)

Die lokale Speicherung erfolgt mittels der **Room-Bibliothek**, einer Abstraktionsschicht über SQLite, die typensichere und komfortable SQL-Abfragen ermöglicht. Die Datenbank umfasst drei Tabellen:

- **Tracks:** Speichert Metadaten wie Startzeit, Gesamtdistanz, Dauer und Durchschnittsgeschwindigkeit.
- **TrackPoints:** Enthält GPS-Punkte mit Zeitstempel, Koordinaten (Breiten- und Längengrad) und Höheninformationen.
- **Photos:** Speichert Fotos, die während der Aufzeichnung gemacht wurden, optional mit Geokoordinaten.

Datenbankdiagramm:



Hinweis: Ein ER-Diagramm mit den Beziehungen zwischen den Tabellen wäre hier sinnvoll. Bitte liefere Details, um diesen Abschnitt zu konkretisieren.

4.5 User Interface Design

Die Benutzeroberfläche wurde mit **XML-Layouts** in Android Studio entworfen, wobei der integrierte Layout-Editor eine interaktive Gestaltung ermöglichte. Das Design ist modern und funktioniert sowohl im **Light Mode** als auch im **Dark Mode**. Es wurde Wert auf folgende Aspekte gelegt:

- **Intuitive Bedienung:** Klare Navigation und leicht zugängliche Funktionen.
- **Übersichtliche Darstellung:** Strecken, Statistiken und Fotos werden klar präsentiert.
- **Visuelle Datenanalyse:** Die Bibliothek **MPAndroidChart** von Philipp Jahoda wurde für die Darstellung von Diagrammen (z. B. Höhen- oder Geschwindigkeitsverläufe)

verwendet.

Das Design wurde eigenständig entwickelt, mit konsistenter Farbgebung und Schriftarten für ein professionelles Erscheinungsbild.

4.6 Herausforderungen

Die Entwicklung brachte folgende Herausforderungen mit sich:

- **GPS-Genauigkeit:** Die Zuverlässigkeit der GPS-Daten war von Geräteeinstellungen (z. B. Energiesparmodus) und Umgebungsbedingungen abhängig. Umfangreiche Tests und Anpassungen waren nötig.
- **Datenstruktur und Visualisierung:** Die Speicherung musste präzise genug sein, um eine genaue visuelle Darstellung zu ermöglichen, was eine sorgfältige Datenbankmodellierung erforderte.
- **Einarbeitung in Kotlin:** Als neue Sprache erforderte Kotlin eine Lernphase, die durch Ähnlichkeiten zu Java und Python erleichtert wurde.
- **Datenvisualisierung:** Die Darstellung von Diagrammen war aufgrund verrauschter GPS-Daten anspruchsvoll und erforderte Datenbereinigungen.
- **Fotoqualität:** Die gespeicherten Fotos waren unscharf, was bis Projektende nicht vollständig behoben wurde. Dies bleibt ein Punkt für zukünftige Updates.

4.7 Testen der App

Die Applikation wurde auf einem realen Android-Smartphone getestet, da die Streckenaufzeichnung echte Bewegungen erfordert. Die Tests umfassten:

- **Funktionstests:** Überprüfung von GPS-Tracking, Datenbankoperationen und Benutzeroberfläche.
- **Alltagsszenarien:** Tests in verschiedenen Outdoor-Situationen (z. B. Fahrradfahren, Wandern) zur Validierung der Robustheit.
- **Fehlerbehebung:** Regelmäßige Tests zur Identifikation und Korrektur von Schwachstellen.

Die Kombination aus automatisierten Unit-Tests und manuellen Tests gewährleistete eine hohe Stabilität.

5 Fazit und Ausblick

Die Applikation erfüllt, mit Ausnahme der Unschärfe bei Fotos, alle Anforderungen und bietet eine zuverlässige, benutzerfreundliche Lösung zur Streckenaufzeichnung. Die lokale Speicherung gewährleistet Datenschutz, während die intuitive Benutzeroberfläche ein positives Nutzererlebnis schafft.

Zukünftige Erweiterungen könnten umfassen:

- **GPX-Export:** Export von Strecken im GPX-Format für andere Anwendungen.

- **Integration mit Fitness-Plattformen:** Anbindung an Dienste wie Strava oder Google Fit.
- **Cloud-Synchronisation:** Sicherer Sync mit einem privaten Server.
- **Routenplanung:** Navigation entlang vorgegebener Strecken.

Die Entwicklung bot wertvolle Einblicke in Android-Entwicklung, GPS-Technologien und Datenbankdesign. Mit den genannten Erweiterungen könnte die App noch vielseitiger werden.