

Projektbericht

**Fahrtenbuch-App**

**Modul „Android-Praktikum: Anwendungsentwicklung“ (CS2012)**

**bei Steffen Vaupel – Sommersemester 2022**

vorgelegt von

Martin, Holstein  
Matrikelnummer 576677

Nuno, Morais Ferreira  
Matrikelnummer 5328457

Felix, Dörmer  
Matrikelnummer 5331839

Technische Hochschule Mittelhessen (THM), Gießen  
Fachbereich Mathematik, Naturwissenschaften  
und Informatik (MNI)

11. Juli 2022

**Zusammenfassung und Hinweise**

Inhaltsverzeichnis

[1 Einleitung 3](#_Toc72497954)

[1.1 Fachlicher Hintergrund (Problem) und Zielsetzung 3](#_Toc72497955)

[1.2 Anforderung (Zusammenfassung) 3](#_Toc72497956)

[1.3 Ausgewählte Technologien/Architektur 3](#_Toc72497957)

[1.4 Organisation und Vorgehensmodell 3](#_Toc72497958)

[2 Anforderungen 4](#_Toc72497959)

[2.1 Grundlegende Funktionsbeschreibung 4](#_Toc72497960)

[2.2 Funktionale Anforderungen 4](#_Toc72497961)

[2.3 Nicht funktionale Anforderungen 4](#_Toc72497962)

[2.4 (Benutzer) Schnittstellen / Ein- Ausgabeformate 4](#_Toc72497963)

[2.5 Fehlverhalten 4](#_Toc72497964)

[2.6 Abnahmekriterien 4](#_Toc72497965)

[3 Entwurf 5](#_Toc72497966)

[3.1 Technische Funktionen (und Funktionsabhängigkeiten) 5](#_Toc72497967)

[3.2 Datenmodell 5](#_Toc72497968)

[3.3 Strukturmodell (Aufbau) 5](#_Toc72497969)

[3.4 Funktionsmodell (Verhalten) 5](#_Toc72497970)

[3.5 Testfälle 5](#_Toc72497971)

[4 Umsetzung 6](#_Toc72497972)

[5 Qualitätssicherung (Testprotokoll) 7](#_Toc72497973)

[6 Evaluation 8](#_Toc72497974)

[7 Anwenderdokumentation 9](#_Toc72497975)

[Glossar 10](#_Toc72497976)

[Anhang 11](#_Toc72497977)

# Einleitung

## Fachlicher Hintergrund (Problem) und Zielsetzung

Unser Ziel ist es, eine App zu gestalten, welche die Gesamtkosten eines Fahrzeugs dokumentiert.

Ausgaben für Benzin, Versicherung, Steuern und Werkstatt werden manuell eingetragen und gespeichert. Fahrten werden automatisch aufgezeichnet mit Hilfe des Bluetooth-Beacons im Auto -> Fahrtanfang: Kopplung im Auto mit Bluetooth, Fahrtende: Entkopplung. Falls ein Standort neu auftritt, an dem eine Fahrt endet wird nachgefragt um welche Kategorie es sich handelt (beruflich oder privat) und wird gespeichert. Sollte dieser Standort erneut als Ziel ermittelt werden, wird die Fahrt automatisch der gleichen Kategorie zugeordnet. Manuelle Einträge sollen auch ermöglicht werden.

Zudem sollen monatliche/jährliche Statistiken erstellt werden, welche die gesammelten Daten anschaulich aufbereiten.

## Anforderung (Zusammenfassung)

* Automatische Erkennung von Fahrten
* Manuelle Eingabe von Kosten
* Manuelle Eingabe von Fahrten
* Statistiken zu allen Daten

## Ausgewählte Technologien/Architektur

Das Fahrtentracking soll mithilfe des Bluetooth Beacons im Auto erfolgen und mit der GPS-Funktion des Smartphones. Das Fahrtentracking soll so im Hintergrund funktionieren.

Die Datenbank soll lokal auf dem Smartphone sein, weshalb jeder nur seine eigenen Daten speichert und auch nur seine eigenen sehen kann.

Die App im Vordergrund zu benutzen kann der User für 2 Fälle:

1. Manuelles Eintragen von Kosten/Fahrten
2. Betrachten der Statistiken, welche seine Fahrten und Kosten repräsentieren

## Organisation und Vorgehensmodell

Wir treffen uns wöchentlich mind. 1-mal alle zusammen, um zu besprechen, was wir aktuell machen bzw. machen müssen. Wir teilen die Aufgaben, welche wir aktuell machen sollen in Arbeitspakete auf, die wir allein, zu zweit oder auch zu dritt erledigen wollen.

Arbeitspakete:

* Datenbank aufsetzen
* Grundlegende Layouts
* Statistiken
* GPS-Tracking
* Automatisierung von bestimmten Datenerhalten

# Anforderungen

## Grundlegende Funktionsbeschreibung

Basisverlauf 1:

1. User öffnet App
2. User bekommt seine letzten Fahrten angezeigt
3. User drückt auf Statistiken
4. User bekommt Statistiken angezeigt
5. User drückt auf Ausgaben
6. User bekommt seine letzten Ausgaben angezeigt
7. User schließt App

Basisverlauf 2:

1. User setzt sich in sein Auto
2. Bluetooth Beacon des Autos verbindet sich mit seinem Handy
3. User fährt
4. Standorte werden regelmäßig gesammelt
5. Zurückgelegte Strecke wird kontinuierlich berechnet
6. User beendet seine Fahrt
7. Bluetooth Beacon trennt sich vom Smartphone
8. Standorte werden nicht mehr berechnet
9. Zurückgelegte Strecke dieser Fahrt wird gespeichert zusammen mit Datum und Zeit
10. User bekommt Benachrichtigung zur bearbeitung der Fahrt.

Alternative Anwendungsverläufe:

1.2.1. User drückt auf „+“

User bekommt ein Formular zum Eintragen einer Fahrt

User trägt eine bereits getätigte Fahrt manuell ein

User drückt auf Speichern

Die Fahrt wird in der Datenbank gespeichert

1.2.2 User drückt auf Einstellungen

User bekommt Einstellungen angezeigt

User drückt auf Bluetooth-Beacon wählen

User bekommt seine gekoppelten Geräte angezeigt

User wählt sein Auto aus

MAC-Adresse des Autos wird gespeichert

## Funktionale Anforderungen

Das System soll Fahrten automatisch aufzeichnen und in einer Datenbank Datum, Zeit und Strecke der Fahrt speichern.

Das System soll manuelle Eingaben bei Fahrten und Ausgaben ermöglichen.

Das System soll die Fahrten und Ausgaben des Users darstellen.

Das System soll es dem User ermöglichen bestimmten Kategorien Standorte zuzuweisen.

Das System soll es dem User ermöglichen seinen Bluetooth-Beacon seines Autos auszuwählen.

## Nicht funktionale Anforderungen

Das System soll die Fahrten und Ausgaben des Users übersichtlich und geordnet darstellen.

Die App soll intuitiv zu bedienen sein.

## (Benutzer) Schnittstellen / Ein- Ausgabeformate

Bisher ist keine Funktionalität vorgesehen, wo der User einen Dateiimport/-export tätigen kann.

Daten werden entweder automatisch gesammelt oder der User muss ein Eingabeformular ausfüllen.

## Fehlverhalten

Der Nutzer fährt zu lange durch einen Tunnel oder Ort mit schlechtem GPS-Empfang.

Auf dem Gerät ist kein Speicher mehr für die Datenbankeinträge vorhanden.

## Abnahmekriterien

Fahrten werden strukturiert angezeigt im Fragment Fahrten.

Ausgaben werden strukturiert angezeigt im Fragment Ausgaben.

Passende und visuell schöne Statistiken werden im Fragment Statistiken angezeigt.

In den Einstellungen kann der Bluetooth-Beacon des eigenen Autos ausgewählt werden.

In den Einstellungen können Kategorien Standorte zugeordnet werden.

Fahrten und Ausgaben können manuell hinzugefügt werden.

Fahrten werden automatisch erkannt, sofern der Bluetooth-Beacon ausgewählt wurde.

# Entwurf

Daten, welche gesammelt werden, werden in einer eigenen SQLite Datenbank gespeichert.

Design:

Es ist eine 4-geteilte Activity geplant.

1. Anzeigen von Fahrten
2. Anzeigen von Ausgaben
3. Anzeigen von Statistiken
4. Einstellungen

In einer Auswahl-Bar unten am Bildschirm kann einfach zwischen den 4 Fragmenten hin und her gesprungen werden.

Die Einzelnen Fragmente können weitere Fragmente aufrufen, um weitere Funktionalität aufzurufen. Beispielsweise kann man im 1. Fragment auf ein „+“-Button drücken um zu einer neuen Ansicht zu gelangen, in der man ein Formular ausfüllt und so eine neue Fahrt gespeichert werden kann.

Die manuellen Eingaben für die Fahrten und Ausgaben erfolgen über zwei Separate Fragmente. Die Distanz und Ausgaben werden über Texteingabefelder, die Kategorie und das Ausgabenintervall über Spinner erfasst. Das Datum und die Uhrzeit werden über das von Google bereit gestellte Datums- und Uhrzeit Auswahl eingeben werden.

Die Datensätze werden als Recycler List angezeigt.

Standorte werden gespeichert, indem man in über eine Kartenansicht einen Standort für eine Kategorie auswählt. Davon werden die Längen- und Breitengrade gespeichert. Die Start- und Endpunkte werden mit den gespeicherten Standorten verglichen. Dadurch wird eine Fahrt automatisch zu einer Kategorie zugeordnet.

Die App ist für API Version ab „29“ konzipiert. Zudem wird ein GPS und Bluetooth Modul benötigt.

## Datenmodell

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

## Strukturmodel

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

3.4 Funktionsmodell

## Der Nutzer kann manuell die Fahrten und Ausgaben über die Benutzerschnittstelle einfüge oder bearbeiten. Zusätzlich werden durch die Fahrtenerkennung mit dem GPS-Modul Fahrten erkannt und auch eingefügt. Die Daten werden vor der Speicherung kontrolliert und formatiert. Anschließend werden die Datensätze in eine interne Datenbank gespeichert. Von da aus werden die Daten aufbereitet und in einer Statistik dargestellt.

## Testfälle

Als Testfälle wird das manuelle Einfügen von Fahrten und Ausgaben genützt. Zudem wird im Moment für Fahrten bei jedem Start der App mehrere hundert Fahrten neu zufällig über einen Zeitraum von 3 Jahren generiert. Ausgaben werden auch für jede Kategorie generiert. Zusätzlich können auch Daten manuell eingetragen werden.

Statistiken werden abhängig von den gegebenen Daten erzeugt und sind so auch öfters getestet worden.

In den Eingabefeldern für die Distanz und Ausgabemenge wird darauf geprüft, ob Zahlen eingegeben werden.

# Umsetzung

Für das moderne Layout haben wir viel die CardView benutzt. Mit der hatte man den großen Vorteil Elemente zu gruppieren und diese mit Schatten und verschiedenen Vordergrund Farben besser abzuheben. Mit dem setCornerRadius von den CardViews entstannt zudem der Google Material-Design look.

Eine Spezialität der Umsetzung in der App war der Umgang mit den Datumobjekten in den Fahrten und Ausgaben. Da SQL Lite keine Date Objekte als Datentyp akzeptiert, sind die Daten zum Zeitpunkt der Erstellung als Millisekunden seit 1970 gespeichert. Die Millisekunden werden bei Verwendung entweder mit einen DateTimeFormater oder in SQL Abfragen mit der strftime Methode, die für SQL Lite die Daten in einen lesbaren Datumsstring umwandelt, formatiert.

Für die Auswahl der „Gespeicherten Orte“, also Orte die bei der Fahrtenerkennung als Geofences agieren sollen, um dann die jeweilige Kategorie der Fahrt zu ermitteln, wurde in den Settings die GoogleMaps API verwendet. Mit dieser kann der benutzer auf der Karte mit einem Pin in der Mitte, den Standort von der jeweiligen Kategorie ermitteln.

Die Geofences wurden, bei der Fahrterkennung jedoch nicht ermittelt, da die Google API dafür, Gebühren für die API-Calls verlangen. Die Voraussetzungen dafür sind jedoch vorhanden.

Die Fahrtenerkennung war aufwendig, da Android einem das Leben mit den ganzen Restriktionen beim Standort und den Hintergrund Services schwer macht. Damit die App auch im nicht laufenden Zustand auf eine Bluetooth Verbindung mit dem Beacon reagieren kann, wurde ein Broadcaster Receiver auf die ACTION\_ACL\_CONNECTED reagiert. Daraufhin wird ein Foreground Service mit einer bestehenden Benachrichtigung gestartet, die den Standort trackt und in ein Array packt. Bei einer Trennung, der Bluetooth Verbindung mit dem Broadcast Receiver ACTION\_ACL\_DISCONNECTED wird die Summe der Distanzen zwischen allen Punkten im Array berechnet. Dies ist dann die Fahrtdistanz. Darauf wird die Fahrt in die Datenbank gespeist und eine Benachrichtigung zur Überprüfung der Werte und der Kategorie dem Benutzer geschickt. Bei Klicken der Benachrichtung kommt man auf die „Fahrt Bearbeiten“ Seite von der jeweiligen Fahrt.

# Qualitätssicherung (Testprotokoll)

Für die Testfälle wurden die Daten in der Datenbank, bei starten der Activity neu randomisiert. So konnten anomalien Entdeckt werden.

Zudem wurde die App durch exportieren, der APK Datei an Komillitonen weitergegeben um Fehler zu finden. So wurde z.B. entdeckt, dass die App bei zu großen Eingaben in der Fahrterstellung, abstürtzt. Auch gab es Fehler, wenn versucht wurde Daten für die Statistik zu bekommen, wenn es in dem gewünschten Zeitraum keine Fahrten oder Ausgaben gab.

# Evaluation

## Das Hauptziel war für uns das automatische Fahrtenerkennen und das richtige einteilen der Fahrten in Kategorien. Das haben wir, wenn auch mit anfänglichen Schwierigkeiten unserer Meinung gut erledig. Die anderen Ziele, das manuelle Eingeben von Daten und die Darstellung in den Statistiken, haben wir auch gut bewältigt. Mit dem erlernten Wissen, welches wir während der Entwicklung angeeignet hat, würde man die Anwendung in manchen Aspekten anders Entwickeln. Wenn auch Möglichkeiten zur Verbesserung existieren, erfüllt die Anwendung alle gesetzten Ziele zu voller Zufriedenheit.

# Anwenderdokumentation

### Grundeinstellungen für die automatische Fahrtenerkennung:

1. Öffne die App
2. Gehe zu Einstellungen (unten bei der App Bar, ganz rechts)
3. Öffne die Bluetooth-Einstellungen in der App
4. Finde deinen Bluetooth-Beacon deines Autos und wähle ihn aus.

### Einfaches benutzen der App:

* App-Bar 1. Von links: „Fahrten“
  + Anzeigen lassen von Fahrten mit Strecke, Datum, benötigter Zeit und Kategorie
* „+“-Symbol bei „Fahrten“
  + Hinzufügen einer Fahrt mit Strecke, Datum, benötigter Zeit und Kategorie
* App-Bar 2. Von links: „Ausgaben“
  + Anzeigen lassen von Ausgaben mit Kosten, Datum und Kategorie
* „+“-Symbol bei „Ausgaben“
  + Hinzufügen einer Ausgabe mit Kosten, Datum und Kategorie
* App-Bar 2. Von rechts: „Statistiken“
  + Anzeigen lassen von unterschiedlichen Statistiken, welche die Fahrten und Ausgaben aus unterschiedlichen Blickwinkeln darstellen
* App-Bar 1. Von rechts: „Einstellungen“
  + Push-Benachrichtigungen erlauben/verbieten
  + Bluetooth-Beacon wählen
  + Standort einer Kategorie zuweisen (per Google-Maps Interface)

# Glossar

* Vorlesung
* Stackoverflow
* <https://github.com/blackfizz/EazeGraph>
* <https://www.geeksforgeeks.org/how-to-display-bluetooth-paired-devices-programmatically-in-android/>
* <https://developer.android.com/guide/navigation/navigation-ui>
* <https://www.vogella.com/tutorials/AndroidActionBar/article.html>

# Anhang