



# Entwicklung einer App zur Steuerung und Datenauswertung einer Drohne zur Luftqualitätsmessung

**Studienarbeit**

an der Dualen Hochschule Baden-Württemberg Stuttgart

von

**Julian Riegger**

04.06.2018

**Bearbeitungszeitraum**  
**Matrikelnummer, Kurs**  
**Ausbildungsfirma**  
**Betreuer**

xx Wochen  
1577610, STG-TINF15-ITA  
Robert Bosch GmbH, Stuttgart  
Thilo Ackermann, Rene Lasse

# Erklärung

Ich erkläre hiermit ehrenwörtlich:

1. dass ich meine Studienarbeit mit dem Thema *Entwicklung einer App zur Steuerung und Datenauswertung einer Drohne zur Luftqualitätsmessung* ohne fremde Hilfe angefertigt habe;
2. dass ich die Übernahme wörtlicher Zitate aus der Literatur sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der Arbeit gekennzeichnet habe;
3. dass ich meine Studienarbeit bei keiner anderen Prüfung vorgelegt habe;
4. dass die eingereichte elektronische Fassung exakt mit der eingereichten schriftlichen Fassung übereinstimmt.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Stuttgart, 04.06.2018

---

Julian Riegger

## **Abstract**

Logfiles beinhalten eine große Menge an Daten, deren Analyse bei der Suche nach Fehlern und der Überwachung einer IT Infrastruktur eine große Hilfe sind. Dabei stellen sich mehrere Herausforderungen. Die Erste ist, dass Logfiles textbasiert sind. Der Nachteil hierbei ist, dass im Vergleich zu einer Datenbank oder einer XML Datei textbasierte Dateien keine klar auslesbare oder durchsuchbare Struktur besitzen. Die Zweite ist, dass Systeme so viele Informationen wie möglich loggen und dadurch die nützlichen bzw. wichtigen Informationen erst herausgefiltert werden müssen.

Für die Analyse von textbasierten Daten eignet sich sehr gut das MapReduce Modell. Außerdem lässt sich das Modell sehr einfach skalieren und auf mehrere Programmläufe verteilen (Master-Worker). Das Apache Hadoop Projekt stellt sowohl für MapReduce, als auch für die Verwaltung von mehreren Programmläufen, ein Basisframework bereit, mit welchem die Entwicklung eines Analyseprogramms durchgeführt werden soll.

Ziel dieser Arbeit ist die Entwicklung einer prototypischen Anwendung zur formatunabhängigen Analyse von Logfiles unter Zuhilfenahme von Apache Hadoop MapReduce. Die Anwendung soll die bisher vorhandenen Monitoring Systeme innerhalb der Infrastruktur ergänzen, wodurch Informationen über den Zustand des Systems schneller erhoben werden können. Des Weiteren sollen aufkommende Fehler besser erkannt werden, um die Reaktionszeit auf diese zu optimieren.

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>I</b>
<b>Abbildungsverzeichnis</b>	<b>II</b>
<b>Tabellenverzeichnis</b>	<b>III</b>
<b>Listings</b>	<b>IV</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Problemstellung . . . . .	1
1.2 Aufgabenstellung . . . . .	2
<b>2 Theoretische Grundlagen</b>	<b>3</b>
2.1 Luftqualität . . . . .	3
2.1.1 Indices . . . . .	3
2.1.2 Schadstoffe . . . . .	4
2.2 Hardware . . . . .	7
2.2.1 DJI Phantom 3 Standard . . . . .	7
2.2.2 iPad 3 . . . . .	9
2.3 iOS-Appentwicklung . . . . .	10
2.3.1 Model View Controller (MVC) . . . . .	10
2.3.2 Cocoa . . . . .	11
2.3.3 Xcode . . . . .	14
2.3.4 SWIFT . . . . .	15
2.3.5 DJI-Software Development Kit (SDK) . . . . .	15
<b>3 Die iOS App</b>	<b>17</b>
3.1 Aufbau . . . . .	17
3.2 Graphical User Interface (GUI) . . . . .	17
3.2.1 MainView . . . . .	17
<b>Literatur</b>	<b>i</b>
<b>Anhang</b>	<b>ii</b>

# Abkürzungsverzeichnis

<b>API</b>	Application Programming Interface
<b>AQI</b>	Air Quality Index
<b>CAQI</b>	Common Air Quality Index
<b>CO</b>	Kohlenstoffmonoxid
<b>CO<sub>2</sub></b>	Kohlenstoffdioxid
<b>EPA</b>	Environmental Protection Agency
<b>IDE</b>	Integrated Development Environment
<b>GUI</b>	Graphical User Interface
<b>GPS</b>	Global Positioning System
<b>MVC</b>	Model View Controller
<b>NO<sub>2</sub></b>	Stickstoffdioxid
<b>NO<sub>x</sub></b>	Stickoxide
<b>O<sub>3</sub></b>	Ozon
<b>SDK</b>	Software Development Kit
<b>SO<sub>2</sub></b>	Schwefeldioxid
<b>OS</b>	Betriebssystem

# Abbildungsverzeichnis

2.1	DJI Phantom 3 Standard . . . . .	7
2.2	DJI Phantom 3 Standard - Fernbedienung . . . . .	8
2.3	iPad 3 . . . . .	10
2.4	MVC Design-Pattern Diagramm <b>mvc1</b> . . . . .	11
2.5	Externe Bibliotheken in Xcode . . . . .	13
2.6	Xcode-Oberfläche . . . . .	14

# Tabellenverzeichnis

2.1	<a href="#">AQI</a> Übersicht . . . . .	4
2.2	DJI Phantom 3 Standard - technische Daten . . . . .	9

# Listings

2.1	Podfile Beispiel . . . . .	12
3.1	Statusleiste . . . . .	18



# 1 Einleitung

In den letzten Jahren bekam das Thema der Luftqualität immer mehr Aufmerksamkeit und gewinnt immer mehr an Bedeutung in der Tagespolitik sowie in der Industrie. Hier ist vor allem die Automobilindustrie in den Fokus gerückt, da die Verbrennungsmotoren in einer sehr emotional geführten Debatte für einen Großteil der schlechten Luft in Großstädten verantwortlich gemacht werden. Nun trägt nicht nur der Verkehr sondern auch die Industrie mit verschiedenen Fabriken, wie auch andere Faktoren, wie zum Beispiel das heizen mit Holz im Winter, zur Verschlechterung der Luftqualität bei. Es wurden in den letzten Jahren immer mehr Messstationen in großen und kleineren Städten platziert, um die Luftqualität zu überwachen.

Zum Thema Luftqualität stellen sich folgende Fragen, welche in der folgenden Arbeit teilweise beantwortet werden sollen.

- Was ist Luftqualität?
- Kann man die Luftqualität messen?
- Was sind Faktoren für die Luftqualität?
- Was sind für den Menschen gefährliche Faktoren in der Luft?

## 1.1 Problemstellung

Von den im Kapitel Einleitung genannten Messstationen ist in der Region Stuttgart die Messstation am Neckartor die bekannteste. Diese misst die Luftqualität aber nur an einer Stelle. Hierbei kann man diskutieren, ob dieser Wert überhaupt aussagekräftig ist oder nicht. Es könnte sein, dass die Wahl für den Ort der Messstation missglückt ist und die gemessenen Werte deshalb nicht aussagekräftig sind.

Ein weiterer Aspekt, welcher zu berücksichtigen ist, sind die Auswirkungen des Wetters auf die Luftqualität.

## 1.2 Aufgabenstellung

Um die genannten Probleme zu umgehen, soll eine Air-Quality-Drone erstellt werden. Hierbei soll eine bereits existierende Drone mit Sensoren ausgestattet werden, welche klassische Werte zur Beurteilung der Luftqualität und zur Beurteilung der Umgebung, wie zum Beispiel die Luftfeuchtigkeit und Temperatur erfassen können. Eine Drone ist agil und kann an verschiedenen Orten und in verschiedenen Luftschichten Messungen durchführen. In dieser Arbeit soll ein Prototyp für eine Drohne zur Messung der Luftqualität erstellt werden. Zu der Drone soll eine App erstellt werden, über die die Drohne bedient werden kann.

## 2 Theoretische Grundlagen

Für die Erstellung der App, sowie für die Auswahl der Sensoren und Erstellung des Messaufbaus ist verschiedenes Wissen notwendig. Diese theoretischen Grundlagen werden im folgenden erläutert.

### 2.1 Luftqualität

Die Luftqualität gibt den Gütegrad der Luft an. Dabei handelt es sich um die Bewertung und den Einfluss von Verunreinigung durch Fahrzeugabgase oder Kraftwerke. Bestandteile der ausgestoßenen Luft, wie beispielsweise Stickoxide (**NO<sub>x</sub>**) oder Feinstaub, sind für die Luftverschmutzung verantwortlich und werden in Kapitel? näher erläutert. Um die Luftqualität möglichst hoch zu halten und den Menschen wenig zu schaden werden Richtlinien mit Grenzwerten eingeführt sowie fortlaufend verfeinert. Diese Richtlinien geben die Höhe sowie deren Eintrittshäufigkeit pro Zeiteinheit an. Das überschreiten der Grenzwerte hat Sanktionen zur Folge.

#### 2.1.1 Indices

Zur allgemeingültigen Bewertung der Luft gibt es diverse Luftqualitätsindices, die basierend auf den gemessenen Werten von Feinstaub (PM<sub>2,5</sub> und PM<sub>10</sub>), bodennahem Ozon (**O<sub>3</sub>**), Stickstoffdioxid (**NO<sub>2</sub>**) und Schwefeldioxid (**SO<sub>2</sub>**), den Gütegrad der lokalen Luft bestimmen. Zur Messung dieser Werte werden derzeit meist stationäre Sensoren, oftmals an viel befahrenen Straßen, eingesetzt.

Es gibt Länder- und Regionenspezifische Indices. Im folgenden werden die Indices für Europa und für die Vereinigten Staaten erläutert.

#### Air Quality Index (**AQI**)

Der sogenannte Air Quality Index (**AQI**) bildet für die Vereinigten Staaten eine Skala von 0 bis 500 ab. Mit steigendem Wert wird die Luftqualität, bezogen auf den Tag, schlechter

und die Risiken für den Menschen schwerwiegender. Die folgende Tabelle stellt die Zusammenhänge von AQI-Wert und den Folgen sowie deren Bedeutung dar. Laut Environmental Protection Agency (EPA) ist ein Wert von maximal 100 als gesundheitlicher Standard angesetzt.

Air Quality Index Levels of Health Concern	Numerical Value	Meaning
Good	0 to 50	Air quality is considered satisfactory, and air pollution poses little or no risk.
Moderate	51 to 100	Air quality is acceptable; however, for some pollutants there may be a moderate health concern for a very small number of people who are unusually sensitive to air pollution.
Unhealthy for Sensitive Groups	101 to 150	Members of sensitive groups may experience health effects. The general public is not likely to be affected.
Unhealthy	151 to 200	Everyone may begin to experience health effects; members of sensitive groups may experience more serious health effects.
Very Unhealthy	201 to 300	Health alert: everyone may experience more serious health effects.
Hazardous	301 to 500	Health warnings of emergency conditions. The entire population is more likely to be affected.

Tabelle 2.1: AQI Übersicht

## Common Air Quality Index (CAQI)

Europäische Länder haben eine eigene, angepasste Skala zur Bewertung der Luftqualität: den Common Air Quality Index (CAQI). Man unterscheidet hierbei kurzfristige (stündlich oder täglich) von langfristigen (jährlich) Luftqualitätsindices. Die stündlich und täglich aktualisierte Luftverschmutzung wird als relatives Maß aus den für Europa bedeutendsten Schadstoffen. Dazu gehören der Feinstaub (PM<sub>2,5</sub> und PM<sub>10</sub>), NO<sub>2</sub> und O<sub>3</sub>. Bei entsprechend verfügbaren Daten können ebenfalls Kohlenstoffmonoxid (CO) sowie SO<sub>2</sub> einbezogen werden. Zur Berechnung der Index-Klasse unterscheidet man zwischen dem Verkehrs- sowie dem Hintergrundindex. Ersteres soll die Verkehrsbelastung anhand Messsensoren in direkter Nähe von vielbefahrenen Straßen und letzteres die allgemeine Belastung einer Stadt darstellen.

Über den für ein ganzes Jahr hinweg berechneten Index lässt sich der Abstand zu den EU-Grenzwerten darstellen. Der Schwellwert ist dabei 1. Ein höherer Wert des Luftqualitätsindex zeugt von einer Überschreitung eines oder mehrerer Schadstoffwerte. Die Vergleichswerte entsprechen einer Empfehlung durch die Welt Gesundheitsorganisation und dienen dem Schutz der Gesundheit.

Die aktuellen Luftqualitätswerte der Messstationen Europas werden von der EUA und der Europäischen Kommission online unter

### 2.1.2 Schadstoffe

Im Folgenden soll auf messbare und für die Luftqualität bzw. die menschliche Gesundheit entscheidende Schadstoffe eingegangen werden. Neben der Erläuterung der luftverunreinigenden Teilchen wird auch auf deren Konsequenzen für den Menschen eingegangen.

### Feinstaub

Ein für die Gesundheit des Menschen entscheidender Schadstoff ist der Feinstaub oder auch Schwebstaub genannt. Diese kleinen, für das menschliche Auge nicht sichtbaren Teilchen, die nur langsam zu Boden sinken, sind zum größten Teil menschlicher Herkunft. Dazu gehören die Emissionen, die durch Kraftfahrzeuge, Öfen und Heizungen in Innenräumen sowie durch die Metallerzeugung, auftreten. Dabei wird der durch Kraftfahrzeuge entstehende Feinstaub nicht nur aus dem Motor emittiert, sondern auch Bremsen- und Reifenabrieb sowie die Aufwirbeln des Straßenstaubes verunreinigen die Luft. Die eben genannten Faktoren tragen zum primären Feinstaub bei. Der sekundäre Feinstaub hat seine Herkunft in der Landwirtschaft. Hierbei entstehen in der Tierhaltung gasförmige Schadstoffe durch die Ammoniakemissionen.

Dabei unterteilt man die Partikel nach ihrer Größe. Alle Teilchen mit einem aerodynamischen Durchmesser kleiner 10 Mikrometer werden als PM<sub>10</sub>, wobei man diejenigen mit einem Durchmesser von kleiner 2,5 Mikrometer als PM<sub>2,5</sub> bezeichnet. Eine weitere Aufschlüsselung innerhalb der eben genannten Grenzen ergibt die Begrifflichkeiten Grobfraktion, für alle Partikel zwischen 2,5 und 10 Mikrometer, sowie die Feinfraktion, für Partikel mit einem Durchmesser kleiner 2,5 Mikrometer. Die aller kleinsten Partikel, aerodynamischer Durchmesser kleiner als 0,1 Mikrometer nennt man ultrafeine Partikel. PM<sub>10</sub> wandern in die Nasenhöhlen, PM<sub>2,5</sub> gelangen über die Atemwege in die Bronchien sowie Lungenbläschen und setzen sich dort ab. Die ultrafeinen Partikel können bis in die Blutgefäße eindringen können. Als Folgen können abhängig von der Partikelgröße Atembeschwerden entstehen sowie die Gefahr eines Herzinfarkts oder einer Lungenerkrankung steigen.

### NO<sub>x</sub>

Stickstoffoxide sind die Verbindung aus unterschiedlich vielen Stickstoff- und Sauerstoffatomen. Die wichtigsten Vertreter dieser Gruppe sind das Stickstoffmonoxid und das Stickstoffdioxid. Diese Oxide entstehen bei unerwünschten Nebenreaktionen während der Verbrennung von Benzin, Öl, Gas oder Kohle. Sie stellt einen sehr reaktionsfreudigen Stoff dar, wodurch es nicht nur zur Ozonbildung, sondern auch zur Feinstaubbelastung beiträgt. Stickstoffdioxide können vor allem bei Asthmatikern zur Bronchienverengung führen.

### SO<sub>2</sub>

Das farblose aber stark riechende Schwefeldioxid wird vor allem bei der Verbrennung von Kohle oder Öl erzeugt und liegt im Normalfall als Gas vor.

Für den Menschen hat dieser Schadstoff Schleimhaut- und Augenreizungen sowie Atemwegsprobleme zur Folge.

Heutzutage ist die Belastung durch  $\text{SO}_2$  jedoch nicht mehr kritisch und die Gesundheitsrisiken akut nicht vorhanden.

Zudem entstehen aus Schwefeldioxid Sulfatpartikel in der Atmosphäre, die die  $\text{PM}_{10}$  Belastung verstärken.

### **O<sub>3</sub>**

Bodennahes  $\text{O}_3$  gilt als sekundärer Schadstoff, da es erst durch photochemische Prozesse entsteht und nicht direkt emittiert wird. Es ergibt sich vor allem aus  $\text{NO}_x$  sowie flüchtigen organischen Verbindungen. Diese beiden sogenannten Vorläuferstoffe werden überwiegend vom Menschen erzeugt. Während  $\text{NO}_x$  von Kraftfahrzeugen emittiert werden, entstehen flüchtige organische Stoffe bei der Verwendung von Lösemitteln, wie zum Beispiel in Farben, Klebstoffen, Reinigungsmitteln oder durch die Verbrennung von Kraftstoff.

Zu den gesundheitlichen Folgen gehören eine geringere Lungenfunktion sowie Atemwegsbeschwerden. Diese Wirkungen treten vor allem bei körperlicher Belastung sowie bei besonderes anfälligen oder vorgeschädigten Personen auf.

### **CO**

Das gasförmige, farb- und geruchslose  $\text{CO}$  wird bei der unvollständigen Verbrennung von Kraftstoffen freigesetzt. Der Grund hierfür ist Sauerstoffmangel, der bei extrem niedriger Dosierung zu einer  $\text{CO}$  Konzentration führt und in einem solchen Fall als Atemgift wirkt. Das liegt an der Beeinträchtigung der Sauerstoffaufnahme und zieht negative Konsequenzen für das zentrale Nervensystem mit sich.

Des Weiteren ist  $\text{CO}$  ein Bestandteil bei der Bildung von bodennahem  $\text{O}_3$ .

### **Kohlenstoffdioxid**

$\text{CO}_2$  ist ein farb- und geruchsloses Gas. Es ist als Treibhausgas für das auf der Erde entstehende Klima verantwortlich, indem es einen Teil der Wärme, die ins Weltall ausgestrahlt wird zurück auf die Erde emittiert. Neben den großen Vorkommnissen im Weltall wird es ebenfalls bei der Zellatmung von Menschen und vielen Tieren ausgeschieden. Ein weiterer Entstehungsort ist der Verbrennungsvorgang von Öl, Holz oder Kohle. Ein Problem von  $\text{CO}_2$ , ist dass sich dieser Schadstoff nicht selbstständig wieder abbauen kann. Die einzige Möglichkeit zur Reduzierung des  $\text{CO}_2$ -Gehaltes ist die Photosynthese oder die physikalische Speicherung in Gewässern. Dabei entsteht Glucose und Sauerstoff.

Die folgende Tabelle zeigt die Auswirkungen von diversen CO<sub>2</sub>-Konzentration auf den Menschen.

## 2.2 Hardware

Um die Luftqualität in verschiedenen Luftschichten messen zu können ist diverse Hardware notwendig.

Für die Messung werden unterschiedliche Sensoren benötigt, die die wichtigsten Aspekte der Luftqualität, wie zum Beispiel den Feinstaub, messen.

Um die Daten zu verarbeiten und auszulesen ist ein  $\mu$ -Controller notwendig. In dieser Arbeit wurde ein XDK der Firma Bosch verwendet.

Um agil messen zu können wird eine Drone der Marke DJI eingesetzt. Die Modellbezeichnung lautet Phantom 3 Standard.

### 2.2.1 DJI Phantom 3 Standard

Der Aufbau der Drone ist in folgender Abbildung zu sehen. Die Drone hat vier Propeller, welche die Antriebskraft leisten. Um Bilder und Videos aufzunehmen ist auf der Unterseite der Drone ein Gimbal mit einer Kamera befestigt. Für eine sichere Landung und einen guten Stand hat die Drone zwei Beine.



Abbildung 2.1: DJI Phantom 3 Standard

Die Drone lässt sich auf verschiedenen Wegen steuern. Es gibt die Möglichkeit der klassischen manuellen Steuerung über die Fernbedienung oder man lässt die Drone autonom fliegen.

Allgemein ist eine App notwendig, um die Drohne zu steuern und alle Features, welche mit der Drohne kommen, zu nutzen. Hierfür stellt die Firma DJI eine eigene App, die DJI GO App zur Verfügung. Es wird aber auch ein Software Development Kit ([SDK](#)) bereitgestellt, mit dem sich eigene Apps entwickeln lassen. Durch die durch das Software Development Kit bereitgestellten Funktionen lassen sich die Features der DJI GO App replizieren und erweitern.

Das manuelle fliegen lässt sich einfach über die mitgelieferte Fernbedienung realisieren. Die zwei Steuerknüppel dienen hierbei zur Steuerung. Die Funktion der einzelnen Steuerknüppel lässt sich über die DJI eigene App konfigurieren. Hier kann der Nutzer seine Vorlieben einstellen.

Um die App bequem während des Fluges bedienen zu können ist an der Fernbedienung eine Halterung montiert in die man das Endgerät, auf der die App ausgeführt wird, befestigen kann.

Der Aufbau der Fernbedienung ist in folgender Abbildung zu sehen.



Abbildung 2.2: DJI Phantom 3 Standard - Fernbedienung

Die zweite Möglichkeit ist, die Drohne autonom fliegen zu lassen. Hierbei werden sogenannte Waypoints einer Mission hinzugefügt, welche gestartet wird. Die Waypoints beinhalten die Koordinaten mit Längengrad und Breitengrad, sowie der Flughöhe und weiteren Informationen, wie der Fluggeschwindigkeit oder dem Radius mit welchem der Punkt umflogen werden soll. Um den autonomen Flug zu starten wird die Mission gestartet. Während des Fluges erkennt die Drohne über die Kamera Hindernisse und vermeidet eine Kollision mit diesen.

Hierbei spielt das Global Positioning System ([GPS](#)) der Drohne eine wichtige Rolle. Das intelligente System merkt sich den Startpunkt der Drohne. Je nach Einstellung kehrt die Drohne automatisch zum Startpunkt zurück, sollte der Akkustand eine bestimmte



Grenze unterschreiten, die Drohne außer Reichweite der Fernbedienung sein oder die return-to-home-Funktion ausgeführt wird.

Die folgende Tabelle gibt eine Übersicht über die wichtigsten, für diese Arbeit relevanten technischen Daten.

Gewicht	1216g
Diagonale Größe	350mm
max. Steiggeschwindigkeit	$5 \frac{m}{s}$
max. Sinkgeschwindigkeit	$3 \frac{m}{s}$
max. Fluggeschwindigkeit	$16 \frac{m}{s}$
max. Flughöhe über NN	6000m
max. Flugzeit	ca. 25 min
Betriebstemperatur	0° bis 40°C
Positionsbestimmung	GPS
	FCC: 1000m
max. Sendereichweite	CE: 500m
	Flughöhe: 120m

Tabelle 2.2: DJI Phantom 3 Standard - technische Daten

### 2.2.2 iPad 3

Um die in dieser Arbeit zu erstellende App auszuführen und das Produkt zu testen ist ein mobiles Endgerät notwendig.

Bei dem mobilen Endgerät handelt es sich um ein iPad der 3. Generation von der Firma Apple.

Das iPad hat die Abmessungen:

- Höhe: 241,2 mm
- Breite: 185,7 mm
- Tiefe: 9,4 mm

- Gewicht: 652 g

Die Bedienfläche ist ein 9,7"großer Multi-Touch Display. Das iPad ist **WLAN!** fähig und kann eine Bluetooth Verbindung aufbauen.



Abbildung 2.3: iPad 3

## 2.3 iOS-Appentwicklung

Bei der Appentwicklung für iOS Geräte bietet sich die Apple eigene Programmiersprache Swift an, welche für die in dieser Arbeit erstellten App auch verwendet wurde.

Die Entscheidung für ein für das Projekt sinnvolles Design-Pattern fiel auf das Model View Controller (MVC) Pattern.

Für die Ansteuerung der DJI-Drone ist das DJI-SDK notwendig, sowie für die Einbindung externer Bibliotheken ist Wissen über Cocoa Pods notwendig.

Im folgendem werden die genannten Grundlagen in einzelnen Unterkapiteln kurz beschrieben.

### 2.3.1 Model View Controller (MVC)

Das MVC-Pattern besteht, wie der Name sagt, aus drei verschiedenen Teilen. Dem Model, dem Controller und der View. Das Model dient ausschließlich zur Speicherung von Daten. Zum Beispiel werden aktuelle Daten der Anwendung, wie zum Beispiel eine Flugroute in einem Model abgespeichert. Die View ist für die Darstellung der Inhalte und Daten

zuständig. Ebenso ist die View dafür zuständig die Eingaben eines Nutzers an den entsprechenden Controller weiterzuleiten. Die View beinhaltet auch die gesamte Graphical User Interface (GUI). Der Controller beinhaltet die Anwendungslogik und ist für die Steuerung der Anwendung verantwortlich.

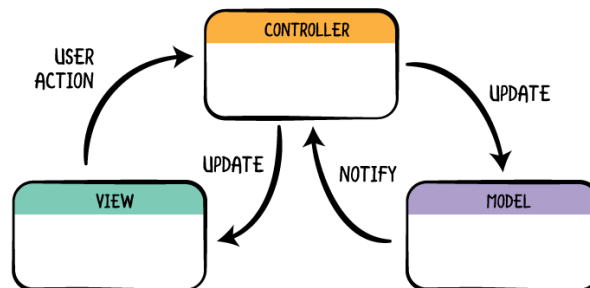


Abbildung 2.4: MVC Design-Pattern Diagramm **mvc1**

Das **MVC** Pattern kann verschieden streng implementiert werden. In dieser Arbeit wird das Pattern in einer leichten Form angewendet. Es wird sich nicht exakt an die Spezifikation aus der Literatur gehalten.

Eine mit Xcode und Swift geschriebenen App hat als Kern eine `UIApplication`-Klasse. Diese Klasse verarbeitet die Interaktion zwischen System und Objekten der App. Sie verwaltet die geöffneten Views und leitet Ereignisse durch Nutzereingaben oder vom System kommend an die entsprechenden Controller weiter.

Die `ViewController` Objekte stellen in einer iOS-Anwendung die Controller des **MVC** Patterns dar. Eine `ViewController`-Klasse verarbeitet Nutzereingaben und aktualisiert die View.

Die View ist in der iOS-Programmierung durch sogenannte Storyboards realisiert.

### 2.3.2 Cocoa

#### Application Programming Interface (**API**)

Cocoa ist eine Application Programming Interface (**API**) zur Programmierung unter den Betriebssystem (**OS**) MacOS. Für die Apple **OS** von mobilen Endgeräten, welche über Touch-Displays verfügen, wurde die Cocoa **API** zur CocoaTouch **API** erweitert. CocoaTouch beinhaltet Funktionen für die Nutzereingabe über Eingaben durch Gesten. Somit wird für die Entwicklung von iOS Apps, wie in dieser Arbeit, die CocoaTouch **API** verwendet.

Die Entwicklung für Apps mit der Cocoa **API** erfolgt mit den Apple eigenen Developer

Tools, Xcode, welches im nächsten Kapitel beschrieben wird, und dem Interface Builder. Die hauptsächlich für die [API](#) gedachten Programmiersprachen sind Objective-C und die Apple eigene Programmiersprache Swift. Die Programmierung in C und C++ ist generell auch möglich.

Der Aufbau von Cocoa ist im Allgemeinen einfach gehalten. Cocoa besteht aus drei verschiedenen Frameworks.

- *Foundation*: beinhaltet alle relevanten Basisklassen, wie Strings, Arrays, Iterators, et cetera.
- *UIKit*: stellt Klassen zur Entwicklung von Graphical User Interface ([GUI](#)) zur Verfügung. Zum Beispiel Buttons, Labels, Menüs, usw..
- *Core Data*: dient zur Erstellung von Objektgraphen.

Klassen des Cocoa-Frameworks sind im Quellcode durch die Buchstaben *NS* im Objektnamen zu erkennen.

## Pods

CocoaPods ist ein application level dependency manager für Objective-C, Swift und andere Programmiersprachen, welche in XCode laufen. Pods stellt ein Standardformat zum managen von externen Bibliotheken bereit.

Die Projektabhängigkeiten werden mittels Podfile-Dateien in einem Projekt beschrieben. Im Folgendem ist ein Beispiel zu sehen.

```
1  # platform: iOS, '9.0'
2  use_frameworks!
3  project 'AirQualityDrone.xcodeproj'
4  target 'AirQualityDrone' do
5  pod 'DJI-SDK-iOS' '~> 4.4'
6  pod 'DJI-UILibrary-iOS', '~> 4.4'
7  pod 'CocoaAsyncSocket'
8  pod 'DTMHeatmap'
9  end
```

Listing 2.1: Podfile Beispiel

Um die externen Bibliotheken zu installieren, wird der Befehl *pod install* aufgerufen. Dadurch werden die Quellen der Bibliotheken geladen und das Projekt in Xcode eingerichtet, sodass die Bibliotheken separat gebaut werden. In das Projekt werden die Dateien über eine statische Bibliothek (*libPods.a*).

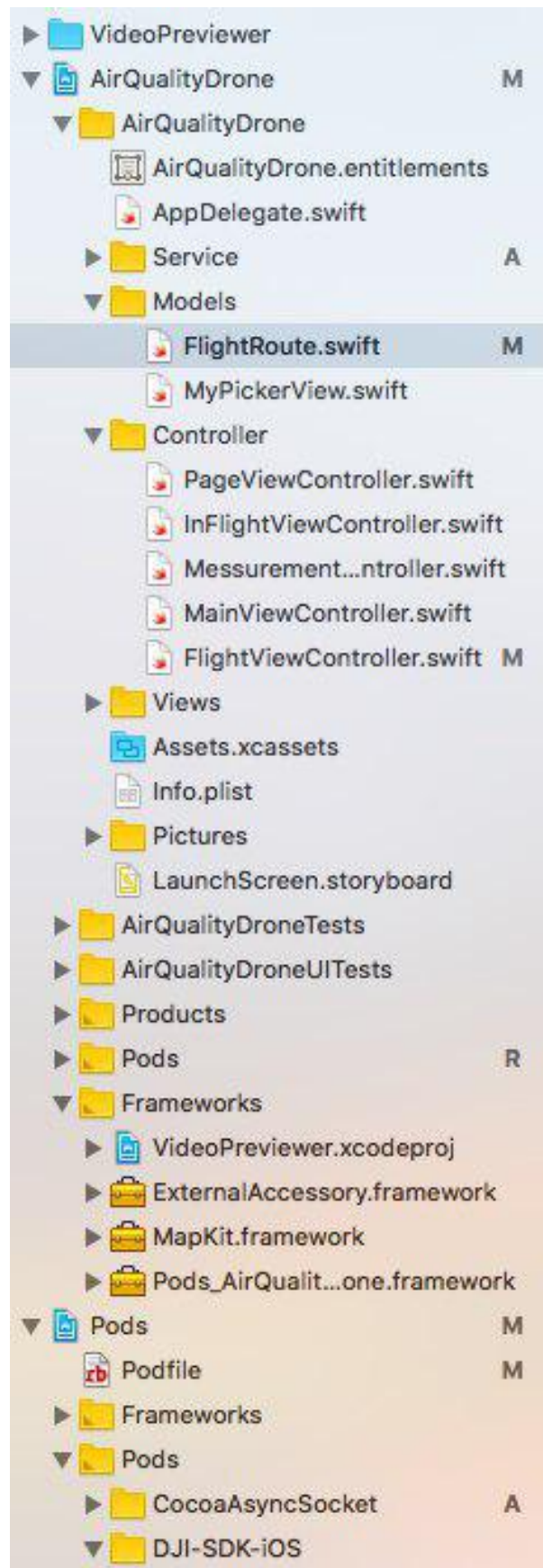


Abbildung 2.5: Externe Bibliotheken in Xcode

### 2.3.3 Xcode

Xcode ist eine Integrated Development Environment von Apple für das OS macOS. Xcode ist für die Entwicklung von Programmen und Apps für macOS, iOS, tvOS und watchOS gedacht. Die IDE ist Bestandteil der Xcode Tools.

Die Xcode Tools beinhalten:

- Xcode IDE
- Interface Builder
- Instruments
- Xcode Core
- Dashcode
- Quanz Composer
- iPhone Simulator

Die iOS App, welche Bestandteil dieser Arbeit ist, wird ausschließlich mit Xcode entwickelt. Die folgende Abbildung zeigt eine Übersicht der Xcode-Oberfläche.

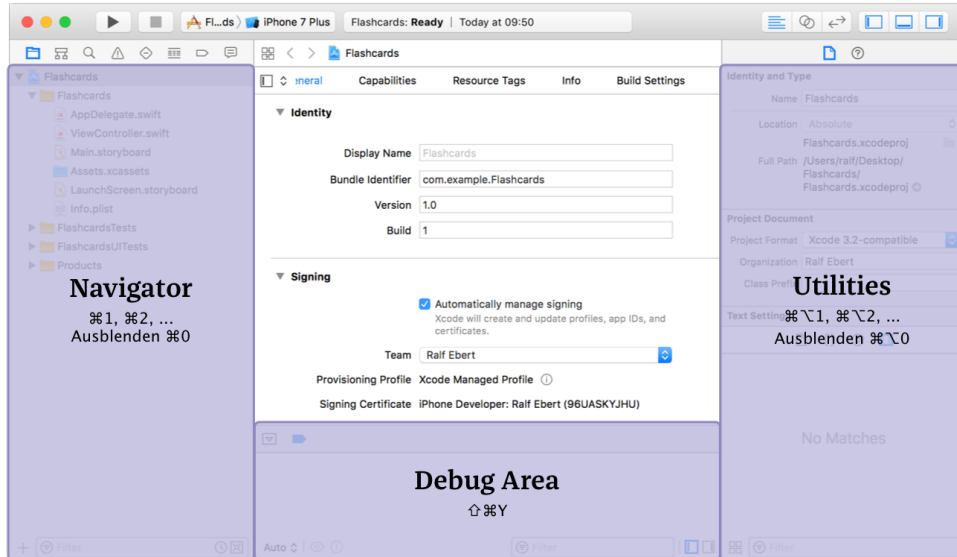


Abbildung 2.6: Xcode-Oberfläche

Zentral ist in Xcode der Editor zu finden. Auf der linken Seite befindet sich der Navigator. Unterhalb des Editors befindet sich die Debug Area und auf der rechten Seite die Utilities. Navigator, Debug Area und Utilities lassen sich über die Buttons in der oberen rechten

Ecke ausblenden und ermöglichen es so das Editor Fenster größer zu machen. Die Projektstruktur ist in Abbildung 2.5 zu sehen.

### 2.3.4 SWIFT

Swift ist eine Programmiersprache des Technik Konzerns Apple. Sie wurde für die Entwicklung von Apps für iOS, Mac, Apple TV und Apple Watch kreiert. Swift ist kostenlos und Open Source. Die Sprache steht unter der Apache 2.0 Open Source Lizenz. Somit kann eine große Community direkt zum Swift Quellcode beitragen.

Swift vereint unterschiedliche Konzepte verschiedener Programmiersprachen, wie zum Beispiel Objective-C und Python. Dies führt dazu, dass Sie sich durch Paradigmen wie objektorientiert und imperativ beschreiben lässt. Swift greift Mechanismen, wie Klassen, Vererbung, Closures, Typinferenz, generische Typen, etc., auf, welche von anderen Programmiersprachen bereits bekannt sind.

Für die Appentwicklung wurde die Programmiersprache Swift unter der Version 4.1 verwendet.

### 2.3.5 DJI-Software Development Kit (SDK)

DJI bietet neben dem Verkauf von Drohnen auch noch andere Leistungen an. Hierzu gehören diverse [SDK](#), die die Entwicklung von Apps und Programmen für die Drohnen von DJI erleichtern. [SDK](#) die angeboten werden sind:

- Mobile SDK
- Onboard SDK
- Guidance SDK
- Payload SDK

In dieser Arbeit ist nur das Mobile SDK relevant.

Das DJI Mobile SDK unterstützt die Plattformen iOS 9.0 oder höher, sowie Android 5.0.0 oder höher. Da das mobile Endgerät mit einem iOS [OS](#) läuft, wird in dieser Arbeit die Ausführung des DJI Mobile SDK für iOS verwendet. Hierbei ist die Programmierung in den Sprachen Swift und Objective-C möglich. Wie im Kapitel 2.3.4 erwähnt wird in dieser Arbeit die Programmiersprache Swift 4.1 verwendet.

Das SDK bietet verschiedene Kernfunktionalitäten an. Zu diesen wichtigen und nützlichen Features gehört die Obstacle avoidance, High and low level flight control, Aircraft state through telemetry and sensor data, Live video feed, Pre defined missions, wie Waypoint,

HotPoint oder FollowMe und State information und control of Battery und Remote Controller.

Für die Entwicklung von [GUI](#) stellt DJI eine UXLibrary zur Verfügung, welche graphische Elemente für die wichtigsten Funktionen des mobile [SDK](#) bereitstellt.



## 3 Die iOS App

Dieses Kapitel beschreibt die Struktur und den Aufbau der für die Ansteuerung der Drohne benötigten iOS App. Es wird ein Überblick über die verwendete Architektur und Benutzeroberfläche, sowie über die verwendeten Klassen gegeben.

### 3.1 Aufbau

Die iOS-App ist mit Hilfe des [MVC](#) Pattern entwickelt. Dadurch gibt sich ein Aufbau aus verschiedenen Controllern und Views. In diesem Projekt gibt es für jede einzelne View einen eigenen Controller. Die Daten, wie die Flugroute sind als Models implementiert. Die Views sind alle in einem Storyboard zu finden. Die Views werden im Kapitel [GUI](#) genauer beschrieben.

zum Anwender dar. Darunter sitzt das DJI Mobile SDK, welches Funktionen der Drohne implementiert. Das [SDK](#) stellt die äußere Schnittstelle zur Drohne dar.

Die Kommunikation zwischen mobilem Endgerät und Drohne wird über die **WLAN!**-Verbindung der Fernbedienung ermöglicht.

### 3.2 Graphical User Interface ([GUI](#))

Die [GUI](#) besteht aus verschiedenen Views. Die folgende Abbildung zeigt den Aufbau der [GUI](#) in einem Storyboard.

#### 3.2.1 MainView

Es gibt eine MainView, welche an der oberen Kante des Displays eine Statusleiste abbildet, die Informationen über den Standort und Zustand der Drohne visualisiert. Neben der Statusleiste beinhaltet die MainView noch eine ContainerView.

## Statusleiste

Die Statusleiste bildet genauer folgende Informationen ab.

- Batteriestatus der Drohne
- Stärke des WiFi-Signals
- Qualität des GPS-Signals
- Die aktuelle Flughöhe der Drohne
- Flugstatus

Realisiert ist die Statusleiste über Elemente der DJI UXLibrary. Die einzelnen View-Elemente erben von den UXLibrary-Klassen. Die UXLibrary Klassen sind an dem Präfix *DUL* zu erkennen. So sind die Obengenannten Widgets durch die UXLibrary Klassen,

- DULBatteryWidget
- DULWifiSignalWidget
- DULGPSSignalWidget
- DULAltitudeWidget
- DULPreFlightStatusWidget

wie folgt realisierbar:

```
1 @IBOutlet var batteryWidget: DULBatteryWidget!  
2 @IBOutlet var wifiWidget: DULWifiSignalWidget!  
3 @IBOutlet var GPSWidget: DULGPSSignalWidget!  
4 @IBOutlet var altitudeWidget: DULAltitudeWidget!  
5 @IBOutlet var flightStatusWidget: DULPreFlightStatusWidget!
```

Listing 3.1: Statusleiste

Die folgende Abbildung zeigt die Darstellung der Statusleiste in der App.

## ContainerView

Die containerView nimmt, außer der Statusleiste am oberen Rand, den restlichen Platz des Displays ein. In die containerView werden die anderen Views geladen. Somit bekommt der Anwender in jeder Ansicht, über die Statusleiste, eine Übersicht über den Zustand der Drohne.

# Literatur

# Anhang

## A. Anforderungsanalyse

## **Anforderungsanalyse**



# Air Quality Drone

## Pflichtenheft Studienarbeit

an der Dualen Hochschule Baden-Württemberg Stuttgart

von

**Julian Riegger, Sebastian Breit**

17.11.2017

**Bearbeitungszeitraum**  
**Matrikelnummer, Kurs**  
**Ausbildungsfirma**  
**Betreuer**

04.11.2017 - 17.11.2017  
1577610, 8870320, STG-TINF15ITA  
Robert Bosch GmbH, Stuttgart  
Rene Lasse, Thilo Ackermann

# Inhaltsverzeichnis

<b>1</b>	<b>Zielbestimmung</b>	<b>1</b>
1.1	Musskriterien . . . . .	1
1.2	Sollkriterien . . . . .	2
1.3	Kannkriterien . . . . .	2
1.4	Abgrenzungskriterien . . . . .	3
<b>2</b>	<b>Produkteinsatz</b>	<b>4</b>
2.1	Anwendungsbereiche . . . . .	4
2.2	Zielgruppen . . . . .	4
2.3	Betriebsbedingungen . . . . .	4
<b>3</b>	<b>Produktumgebung</b>	<b>5</b>
3.1	Software . . . . .	5
3.2	Hardware . . . . .	5
<b>4</b>	<b>Produktfunktionen</b>	<b>6</b>
<b>5</b>	<b>Produktdaten</b>	<b>10</b>
5.1	Non-persistente Daten . . . . .	10
5.2	Persistente Daten . . . . .	10
<b>6</b>	<b>Produktleistungen</b>	<b>12</b>
<b>7</b>	<b>Qualitätsanforderungen</b>	<b>15</b>
<b>8</b>	<b>Testszenarien und Testfälle</b>	<b>16</b>
<b>9</b>	<b>Benutzeroberfläche</b>	<b>23</b>
<b>10</b>	<b>Entwicklungsumgebung</b>	<b>31</b>
<b>11</b>	<b>Abschlussbewertung</b>	<b>32</b>

# 1 Zielbestimmung

## 1.1 Musskriterien

- i. Der Drohnenflug muss mittels der App gestartet werden können
- ii. Die Flugroute muss mittels der App festgelegt werden können
- iii. Der Drohnenflug muss mittels der App abgebrochen werden können
- iv. Die Messwerte müssen über die App einsehbar sein
- v. Die Messwerte müssen über die App exportierbar sein
- vi. Die App muss dem Benutzer ermöglichen, einen Flugbereich auf einer Karte zu markieren
- vii. Die App muss dem Benutzer ermöglichen, die Flughöhe für die Messung auszuwählen
- viii. Die auf der Karte ausgewählte Flugroute muss gestartet werden können
- ix. Die App muss dem Benutzer ermöglichen, aus verschiedenen Messhäufigkeiten auszuwählen
- x. Die Drohne muss Feinstaub messen können (2.5 & 10  $\mu m$ )
- xi. Die Drohne muss Druck messen können
- xii. Die Drohne muss Feuchtigkeit messen können
- xiii. Die Drohne muss Temperatur messen können
- xiv. Die Drohne muss Stickoxide ( $NO_x$ ) messen können



## 1.2 Sollkriterien

- i. Die Messwerte sollen in der App visualisiert werden
- ii. Die App soll die Messdaten in Abhängigkeit der Höhe zweidimensional auf der Karte anzeigen können
- iii. Basierend auf dem auswählbaren Flugbereichs auf einer Karte, soll die App eine Flugroute automatisch berechnen können.
- iv. Die App soll dem Benutzer ermöglichen, Bereiche explizit aus dem Flugbereich auszuschließen
- v. Die App soll die Drohnenposition während des Fluges auf einer Karte anzeigen
- vi. Die App soll eine Fortschrittsanzeige zur laufenden Messung bieten
- vii. Die App soll einen permanenten Video-Stream während des Drohnenflugs anzeigen
- viii. Die Drohne soll Kohlenstoffdioxid ( $\text{CO}_2$ ) messen können
- ix. Die Drohne soll Ozon ( $\text{O}_3$ ) messen können

## 1.3 Kannkriterien

- i. Der Benutzer kann in der App Flugrouten erstellen, speichern und abrufen
- ii. Der Benutzer kann in der App Messprofile erstellen, speichern und abrufen
- iii. Die App kann die Benutzerprofile exportieren können
- iv. Die App kann durch Verrechnung von verbleibendem Akkustand und der vorgegebenen Flugroute einen Warnhinweis an den Benutzer geben, dass die Messung potenziell nicht ausgeführt werden kann
- v. Die App kann die Messdaten dreidimensional darstellen
- vi. Die Drohne kann Kohlenstoffmonoxid ( $\text{CO}$ ) messen können
- vii. Die Drohne kann Schwefeldioxid ( $\text{SO}_2$ ) messen können
- viii. Die Drohne kann flüchtige organische Verbindungen (VOC) messen können
- ix. Die Drohne kann Methan ( $\text{CH}_4$ ) messen können
- x. Der Anwender kann die Messdaten mittels der App einem Server übermitteln können

## 1.4 Abgrenzungskriterien

- i. Die App muss keinen Mehrbenutzerbetrieb mittels Login Daten ermöglichen
- ii. Die App muss den Anwender nicht benachrichtigen, wenn der Nutzer in verbotenen Bereichen einen Drohnenflug durchführt.
- iii. Die Drohne verfügt über keinerlei Kollisionserkennung, somit muss der Benutzer die sichere Benutzung selbst sicherstellen

## **2 Produkteinsatz**

### **2.1 Anwendungsbereiche**

Das Produkt ist für den Einsatz im Freien gedacht. Es darf nur in Gegenden genutzt werden, in denen es grundsätzlich erlaubt ist, mit Drohnen zu fliegen. Hauptanwendungsbereiche stellen Gebiete dar, in denen eine erhöhte Luftverschmutzung vermutet werden kann, um die dortige Luftqualität und mögliche Gesundheitsrisiken zuverlässig bestimmen zu können.

### **2.2 Zielgruppen**

Das Produkt ist für Personen und Organisationen gedacht, welche einen Beitrag zu einem transparenteren Umgang mit dem Thema Luftqualität leisten wollen. Mögliche Interessengruppen wären hierbei beispielsweise staatliche Organisationen, die die Luftqualität in Städten überwachen wollen, Firmen, die die Luftverschmutzung in der Nähe ihrer Produktionsstätten messen wollen, sowie alle gemeinnützigen Organisationen und Privatpersonen, denen das Thema Luftqualität am Herzen liegt. Das Produkt ist nicht für Personen bestimmt, welche den Umgang mit Drohnen und mobilen Endgeräten nicht beherrschen oder der ihnen verboten ist.

### **2.3 Betriebsbedingungen**

Das Produkt ist nur für den Einsatz mit ausreichendem Akkustand gedacht. Das Produkt ist nicht für den Einsatz unter extremen Wetterbedingungen gedacht. Das Produkt ist nicht für den Einsatz innerhalb von Gebäuden und geschlossenen Räumen gedacht.

## **3 Produktumgebung**

### **3.1 Software**

- iOS

### **3.2 Hardware**

- Drohne (Phantom 3)
- iPhone/iPad (Gerät mit iOS)

## 4 Produktfunktionen

<b>ID</b>	«F-010»
<b>Funktion</b>	App starten
<b>Akteur</b>	Anwender
<b>Beschreibung</b>	Der Anwender muss die App auf seinem Endgerät (iPhone/iPad) starten können.

<b>ID</b>	«F-100»
<b>Funktion</b>	Flugroute anlegen
<b>Akteur</b>	Anwender
<b>Beschreibung</b>	Der Anwender muss in der Lage sein eine Flugroute in der App anzulegen. Durch setzen von Punkten auf einer Karte, kann die Flugroute gesetzt werden. Eine weitere Möglichkeit ist die Flugroute manuell abzufliegen, sodass diese gespeichert wird.

<b>ID</b>	«F-110»
<b>Funktion</b>	Flugroute bearbeiten
<b>Akteur</b>	Anwender
<b>Beschreibung</b>	Der Anwender muss in der Lage sein eine bestehende Flugroute zu bearbeiten.

<b>ID</b>	«F-120»
<b>Funktion</b>	Flugroute löschen
<b>Akteur</b>	Anwender
<b>Beschreibung</b>	Der Anwender muss in der Lage sein eine bestehende Flugroute zu löschen.

---

#### 4 Produktfunktionen

---

<b>ID</b>	«F-130»
<b>Funktion</b>	Flugroute auswählen
<b>Akteur</b>	Anwender
<b>Beschreibung</b>	Der Anwender muss in der App eine Flugroute auswählen können. Es kann eine Flugroute ausgewählt werden. Existiert noch keine Flugroute kann eine neue Flugroute erstellt werden (F-100).

<b>ID</b>	«F-200»
<b>Funktion</b>	Messprofil erstellen
<b>Akteur</b>	Anwender
<b>Beschreibung</b>	Der Anwender muss ein neues Messprofil erstellen können. Beim Erstellen muss die Messhäufigkeit und die zu messenden Daten ausgewählt werden.

<b>ID</b>	«F-210»
<b>Funktion</b>	Messprofil ändern
<b>Akteur</b>	Anwender
<b>Beschreibung</b>	Der Anwender muss ein von ihm erstelltes Messprofil ändern können.

<b>ID</b>	«F-220»
<b>Funktion</b>	Messprofil löschen
<b>Akteur</b>	Anwender
<b>Beschreibung</b>	Der Anwender muss ein von ihm erstelltes Messprofil löschen können.

<b>ID</b>	«F-230»
<b>Funktion</b>	Messprofil auswählen
<b>Akteur</b>	Anwender
<b>Beschreibung</b>	Der Anwender muss vor dem Starten einer Route ein Messprofil auswählen können.

---

#### 4 Produktfunktionen

---

<b>ID</b>	«F-300»
<b>Funktion</b>	Drohnenflug starten
<b>Akteur</b>	Anwender
<b>Beschreibung</b>	Der Anwender muss den ausgewählten Drohnenflug (F-130) starten können. Vor dem Start muss ein Messprofil ausgewählt werden (F-230).

<b>ID</b>	«F-310»
<b>Funktion</b>	Drohnenflug abbrechen
<b>Akteur</b>	Anwender
<b>Beschreibung</b>	Der Anwender muss einen Drohnenflug, den er gestartet hat (F-300) abbrechen können.

<b>ID</b>	«F-400»
<b>Funktion</b>	Messdaten als Tabelle anzeigen
<b>Akteur</b>	Anwender
<b>Beschreibung</b>	Der Anwender muss sich die Messdaten in einer Tabelle anzeigen lassen können.

<b>ID</b>	«F-410»
<b>Funktion</b>	Messdaten in Karte anzeigen
<b>Akteur</b>	Anwender
<b>Beschreibung</b>	Der Anwender muss sich die Messdaten in einer Karte anzeigen lassen können.

<b>ID</b>	«F-420»
<b>Funktion</b>	Messdaten exportieren
<b>Akteur</b>	Anwender
<b>Beschreibung</b>	Der Anwender muss die gemessenen Daten als csv-Datei exportieren können.

---

#### 4 Produktfunktionen

---

<b>ID</b>	«F-430»
<b>Funktion</b>	Messdaten an den Server übermitteln
<b>Akteur</b>	Anwender
<b>Beschreibung</b>	Der Anwender muss die gemessenen Daten als csv-Datei mittels der App an den Server übermitteln können.



## 5 Produktdaten

### 5.1 Non-persistente Daten

ID	«D-001»
Inhalt	Video-Stream
Bestandteile	<ul style="list-style-type: none"><li>• Video-Stream der Kamera an der Drohne</li></ul>

### 5.2 Persistente Daten

ID	«D-010»
Inhalt	Flugrouten-Koordinaten
Bestandteile	<ul style="list-style-type: none"><li>• Beschreibung der Flugroute</li><li>• GPS-Koordinaten der abzufliegenden Punkte</li></ul>

<b>ID</b>	«D-020»
<b>Inhalt</b>	Messprofil
<b>Bestandteile</b>	<ul style="list-style-type: none"> <li>• Beschreibung des Messprofils</li> <li>• Genauigkeit der Messung (in Messungen/Zeiteinheit)</li> <li>• Zu messende Werte (Feinstaub, NO<sub>x</sub>, CO<sub>2</sub>, ...)</li> </ul>

<b>ID</b>	«D-030»
<b>Inhalt</b>	Messdaten
<b>Bestandteile</b>	<ul style="list-style-type: none"> <li>• Flugkoordinaten</li> <li>• Zeitstempel</li> <li>• Temperatur</li> <li>• Feuchtigkeit</li> <li>• Druck</li> <li>• Feinstaub-Partikel-Konzentration</li> <li>• Stickoxid-Konzentration</li> <li>• Kohlenstoffdioxid-Konzentration</li> <li>• Ozon-Konzentration</li> <li>• Methan-Konzentration</li> </ul> <p>Jeweils ein Datensatz pro vorgegebener Zeiteinheit</p>

## 6 Produktleistungen

<b>ID</b>	«L-010»
<b>Leistung</b>	Systemanforderungen
<b>Beschreibung</b>	Die App muss auf den folgenden Betriebssystemen lauffähig sein: <ul style="list-style-type: none"><li>• iOS 9 und neuer</li></ul>

<b>ID</b>	«L-020»
<b>Leistung</b>	GPS Standort ermitteln
<b>Beschreibung</b>	Die App muss über die Drohne den GPS Standort der Drohne ermitteln können.

<b>ID</b>	«L-030»
<b>Leistung</b>	NO <sub>x</sub> Werte messen
<b>Beschreibung</b>	Das Produkt muss NO <sub>x</sub> Werte in seiner Umgebung messen können.

<b>ID</b>	«L-040»
<b>Leistung</b>	Feinstaub Werte messen (2,5 $\mu m$ & 10 $\mu m$ )
<b>Beschreibung</b>	Das Produkt muss Feinstaub Werte in seiner Umgebung messen können.

<b>ID</b>	«L-050»
<b>Leistung</b>	O <sub>3</sub> Werte messen
<b>Beschreibung</b>	Das Produkt muss O <sub>3</sub> Werte in seiner Umgebung messen können.

---

6 Produktleistungen

---

<b>ID</b>	«L-060»
<b>Leistung</b>	CO Werte messen
<b>Beschreibung</b>	Das Produkt muss CO Werte in seiner Umgebung messen können.

<b>ID</b>	«L-070»
<b>Leistung</b>	SO <sub>2</sub> Werte messen
<b>Beschreibung</b>	Das Produkt muss SO <sub>2</sub> Werte in seiner Umgebung messen können.

<b>ID</b>	«L-080»
<b>Leistung</b>	CO <sub>2</sub> Werte messen
<b>Beschreibung</b>	Das Produkt muss CO <sub>2</sub> Werte in seiner Umgebung messen können.

<b>ID</b>	«L-090»
<b>Leistung</b>	CH <sub>4</sub> Werte messen
<b>Beschreibung</b>	Das Produkt muss CH <sub>4</sub> Werte in seiner Umgebung messen können.

<b>ID</b>	«L-100»
<b>Leistung</b>	VOCs Werte messen
<b>Beschreibung</b>	Das Produkt muss VOCs Werte in seiner Umgebung messen können.

<b>ID</b>	«L-110»
<b>Leistung</b>	Luftfeuchtigkeit messen
<b>Beschreibung</b>	Das Produkt muss die Luftfeuchtigkeit in seiner Umgebung messen können.

<b>ID</b>	«L-120»
<b>Leistung</b>	Temperatur messen
<b>Beschreibung</b>	Das Produkt muss die Temperatur in seiner Umgebung messen können.

---

6 Produktleistungen

---

<b>ID</b>	«L-130»
<b>Leistung</b>	Luftdruck messen
<b>Beschreibung</b>	Das Produkt muss den Luftdruck in seiner Umgebung messen können.

<b>ID</b>	«L-140»
<b>Leistung</b>	Zeit messen
<b>Beschreibung</b>	Das Produkt muss die Zeiten der Messungen messen können.

## 7 Qualitätsanforderungen

	Wichtig	Mittel	Niedrig	Nicht relevant
Robustheit	x			
Verfügbarkeit			x	
Kompatibilität				x
Benutzerfreundlichkeit		x		
Zeitverhalten			x	
Änderbarkeit				x
Portierbarkeit			x	

## 8 Testszenarien und Testfälle

<b>ID</b>	«TC-010»
<b>Beschreibung</b>	App starten
<b>Vorbedingung</b>	App ist auf dem Endgerät installiert
<b>Testschritte</b>	1. Der Anwender drückt auf das App Icon auf dem Endgerät
<b>Zu erwartendes Ergebnis</b>	App wird geöffnet

<b>ID</b>	«TC-100»
<b>Beschreibung</b>	Flugroute anlegen
<b>Vorbedingung</b>	-
<b>Testschritte</b>	1. Der Anwender wählt verschiedene aufeinanderfolgende Punkte auf einer Karte aus 2. Der Anwender speichert die Flugroute
<b>Zu erwartendes Ergebnis</b>	Die Flugroute wird angelegt und gespeichert

<b>ID</b>	«TC-110»
<b>Beschreibung</b>	Flugroute bearbeiten
<b>Vorbedingung</b>	Mindestens eine Flugroute existiert
<b>Testschritte</b>	<ol style="list-style-type: none"> <li>1. Der Anwender wählt eine existierende Flugroute (TC-130)</li> <li>2. Der Anwender ändert Punkte der Flugroute</li> <li>3. Der Anwender speichert die Flugroute</li> </ol>
<b>Zu erwartendes Ergebnis</b>	Die Flugroute wird geändert und gespeichert

<b>ID</b>	«TC-120»
<b>Beschreibung</b>	Flugroute löschen
<b>Vorbedingung</b>	Mindestens eine Flugroute existiert
<b>Testschritte</b>	<ol style="list-style-type: none"> <li>1. Der Anwender wählt eine existierende Flugroute (TC-130)</li> <li>2. Der Anwender löscht die Flugroute</li> </ol>
<b>Zu erwartendes Ergebnis</b>	Die Flugroute wird gelöscht

<b>ID</b>	«TC-130»
<b>Beschreibung</b>	Flugroute auswählen
<b>Vorbedingung</b>	Mindestens eine Flugroute existiert
<b>Testschritte</b>	<ol style="list-style-type: none"> <li>1. Der Anwender wählt eine existierende Flugroute</li> </ol>
<b>Zu erwartendes Ergebnis</b>	Die Flugroute ist ausgewählt



<b>ID</b>	«TC-200»
<b>Beschreibung</b>	Messprofil erstellen
<b>Vorbedingung</b>	-
<b>Testschritte</b>	<ol style="list-style-type: none"> <li>1. Der Anwender wählt eine Messhäufigkeit aus</li> <li>2. Der Anwender wählt die zu messenden Daten aus</li> <li>3. Der Anwender speichert das Messprofil</li> </ol>
<b>Zu erwartendes Ergebnis</b>	Das Messprofil wird erstellt und gespeichert

<b>ID</b>	«TC-210»
<b>Beschreibung</b>	Messprofil bearbeiten
<b>Vorbedingung</b>	Mindestens ein Messprofil existiert
<b>Testschritte</b>	<ol style="list-style-type: none"> <li>1. Der Anwender wählt ein existierendes Messprofil (TC-230)</li> <li>2. Der Anwender ändert die Messhäufigkeit oder die zu messenden Daten</li> <li>3. Der Anwender speichert das Messprofil</li> </ol>
<b>Zu erwartendes Ergebnis</b>	Das Messprofil wird geändert und gespeichert

<b>ID</b>	«TC-220»
<b>Beschreibung</b>	Messprofil löschen
<b>Vorbedingung</b>	Mindestens ein Messprofil existiert
<b>Testschritte</b>	<ol style="list-style-type: none"> <li>1. Der Anwender wählt ein existierendes Messprofil (TC-230)</li> <li>2. Der Anwender löscht das Messprofil</li> </ol>
<b>Zu erwartendes Ergebnis</b>	Das Messprofil wird gelöscht

<b>ID</b>	«TC-230»
<b>Beschreibung</b>	Messprofil auswählen
<b>Vorbedingung</b>	Mindestens ein Messprofil existiert
<b>Testschritte</b>	<ol style="list-style-type: none"> <li>1. Der Anwender wählt ein existierendes Messprofil</li> </ol>
<b>Zu erwartendes Ergebnis</b>	Das Messprofil ist ausgewählt

<b>ID</b>	«TC-300»
<b>Beschreibung</b>	Drohnenflug starten
<b>Vorbedingung</b>	-
<b>Testschritte</b>	<ol style="list-style-type: none"> <li>1. Der Anwender wählt den Menüpunkt "Flug starten" aus</li> <li>2. Der Anwender wählt eine existierende Flugroute der Combobox "Flugrouten" aus</li> <li>3. Der Anwender wählt ein existierendes Messprofil aus der Combobox "Messprofile" aus</li> <li>4. Der Anwender startet den Flug durch drücken des Buttons „Flug starten“</li> </ol>
<b>Zu erwartendes Ergebnis</b>	Die Drohne startet und fliegt die ausgewählte Flugroute ab. Messungen werden nach dem ausgewählten Messprofil durchgeführt.

<b>ID</b>	«TC-310»
<b>Beschreibung</b>	Drohnenflug abbrechen
<b>Vorbedingung</b>	Drohnenflug ist im Gange & Drohne in Reichweite der Steuerung
<b>Testschritte</b>	<ol style="list-style-type: none"> <li>1. Der Anwender drückt auf den Button "Messung abbrechen"</li> <li>2. Der Anwender bestätigt die auftretende Warnmeldung</li> </ol>
<b>Zu erwartendes Ergebnis</b>	Der Drohnenflug wird abgebrochen und die Drohne kehrt zum Startpunkt zurück

<b>ID</b>	«TC-400»
<b>Beschreibung</b>	Messdaten als Tabelle anzeigen
<b>Vorbedingung</b>	-
<b>Testschritte</b>	<ol style="list-style-type: none"> <li>1. Der Anwender wählt den Menüpunkt "Messdaten anzeigen"</li> <li>2. Der Anwender wählt den Unterpunkt "Tabelle"</li> <li>3. Der Anwender wählt einen Zeitraum aus, für den er alle Messwerte angezeigt bekommen möchte</li> </ol>
<b>Zu erwartendes Ergebnis</b>	Alle Messdaten werden in einer Tabelle angezeigt

<b>ID</b>	«TC-410»
<b>Beschreibung</b>	Messdaten in Karte anzeigen
<b>Vorbedingung</b>	Es gibt existierende Messdaten
<b>Testschritte</b>	<ol style="list-style-type: none"> <li>1. Der Anwender wählt den Menüpunkt "Messdaten anzeigen"</li> <li>2. Der Anwender wählt den Unterpunkt "Karte"</li> <li>3. Der Anwender wählt einen Zeitraum aus, für den er alle Messwerte angezeigt bekommen möchte</li> </ol>
<b>Zu erwartendes Ergebnis</b>	Alle Messdaten werden in einer Karte angezeigt

<b>ID</b>	«TC-420»
<b>Beschreibung</b>	Messdaten exportieren
<b>Vorbedingung</b>	-
<b>Testschritte</b>	<ol style="list-style-type: none"> <li>1. Der Anwender wählt den Menüpunkt "Messdaten exportieren" aus</li> <li>2. Der Anwender wählt den Speicherort aus</li> </ol>
<b>Zu erwartendes Ergebnis</b>	Alle Messdaten werden als csv-Datei exportiert

<b>ID</b>	«TC-430»
<b>Beschreibung</b>	Messdaten an den Server übermitteln
<b>Vorbedingung</b>	-
<b>Testschritte</b>	<ol style="list-style-type: none"> <li>1. Der Anwender wählt den Menüpunkt "Messdaten an Server übermitteln" aus</li> </ol>
<b>Zu erwartendes Ergebnis</b>	Alle Messdaten werden als csv-Datei an den Server übermittelt

## 9 Benutzeroberfläche

Im folgenden sind Mockups zu sehen, die die zu entwickelnde App darstellen. Folgende Bildschirme werden hier dargestellt:

- Menü
- Menü Pop-Up
- Flug starten
- Flugrouten
- Messprofile
- Messwerte anzeigen (Karte)
- Messwerte anzeigen (Tabelle)



Abbildung 9.1: Menü

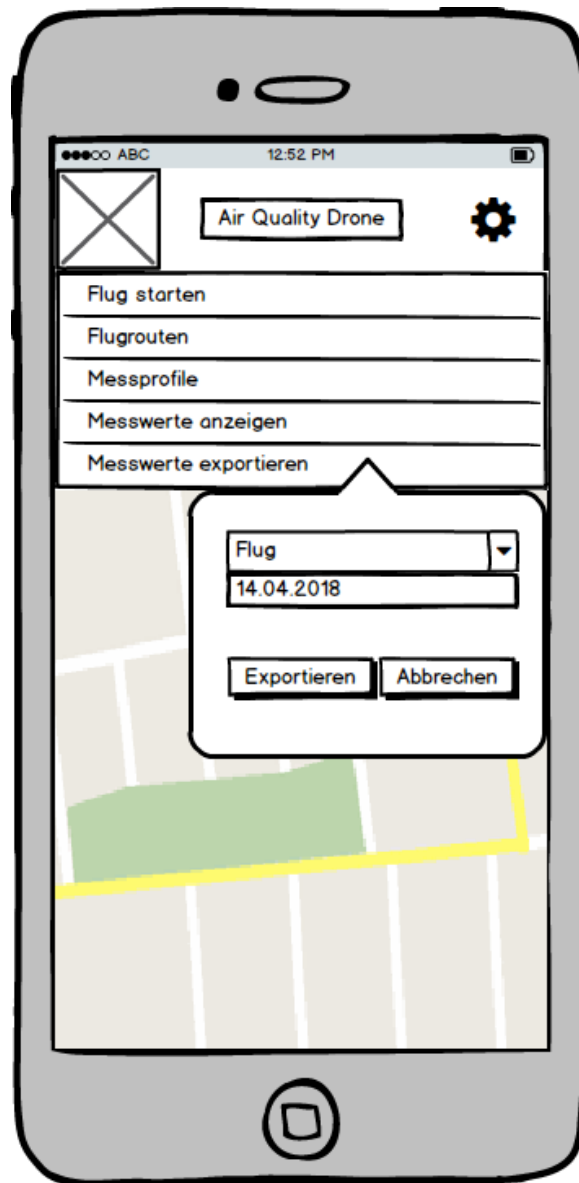


Abbildung 9.2: Menü Pop-Up



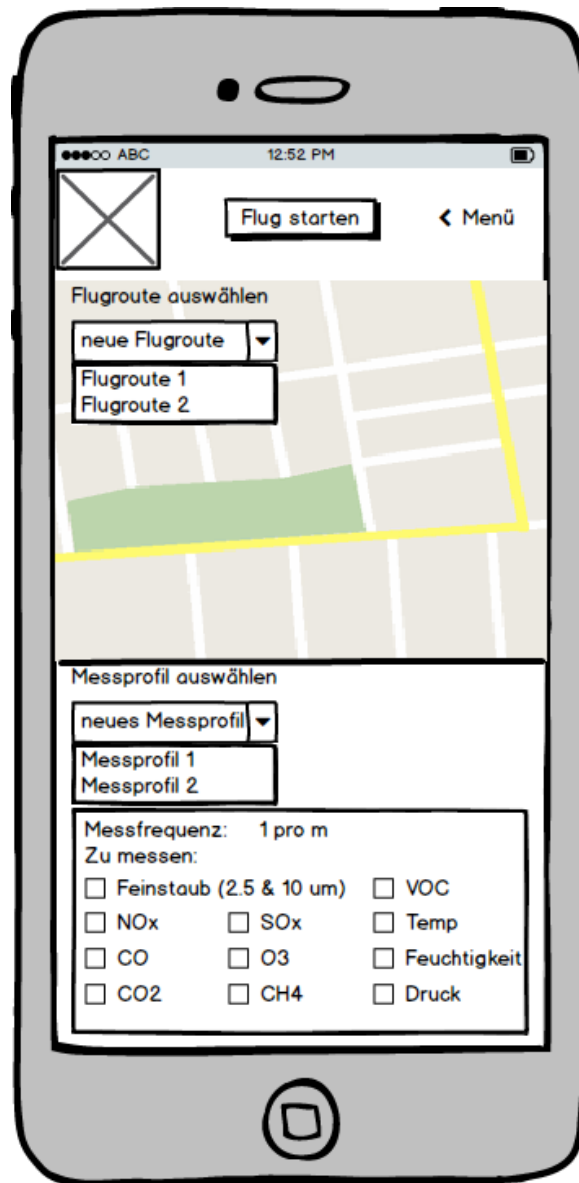


Abbildung 9.3: Flug starten

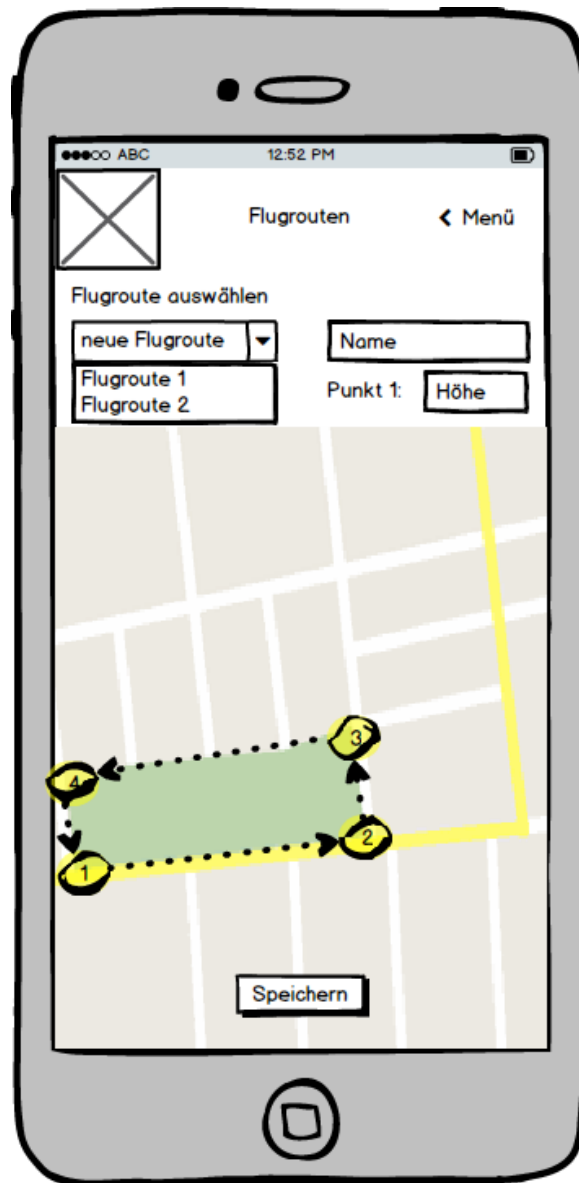


Abbildung 9.4: Flugrouten

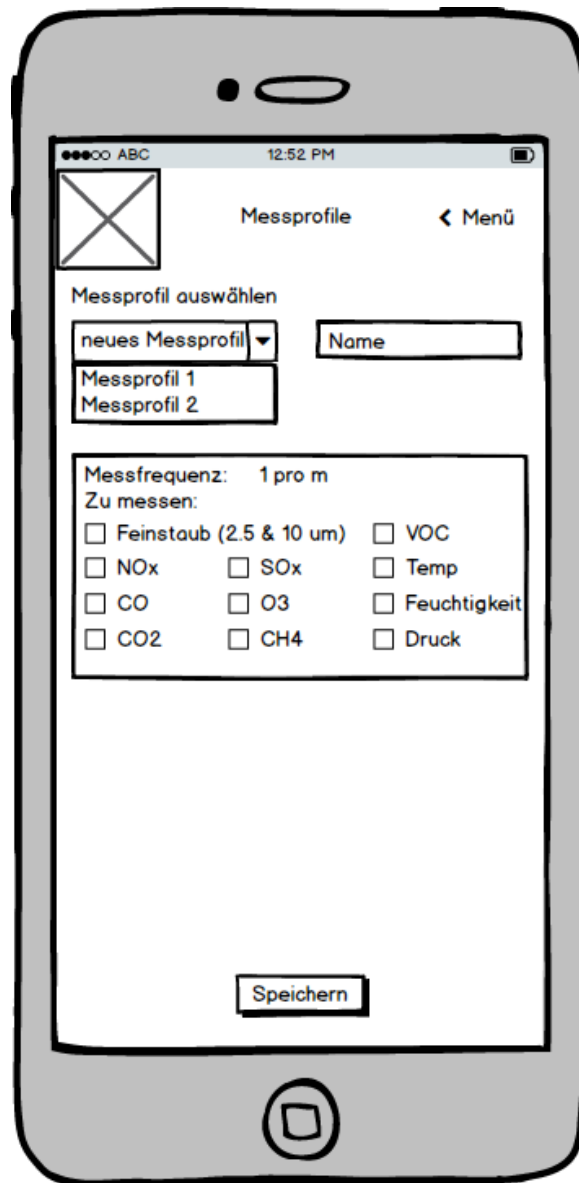


Abbildung 9.5: Messprofile

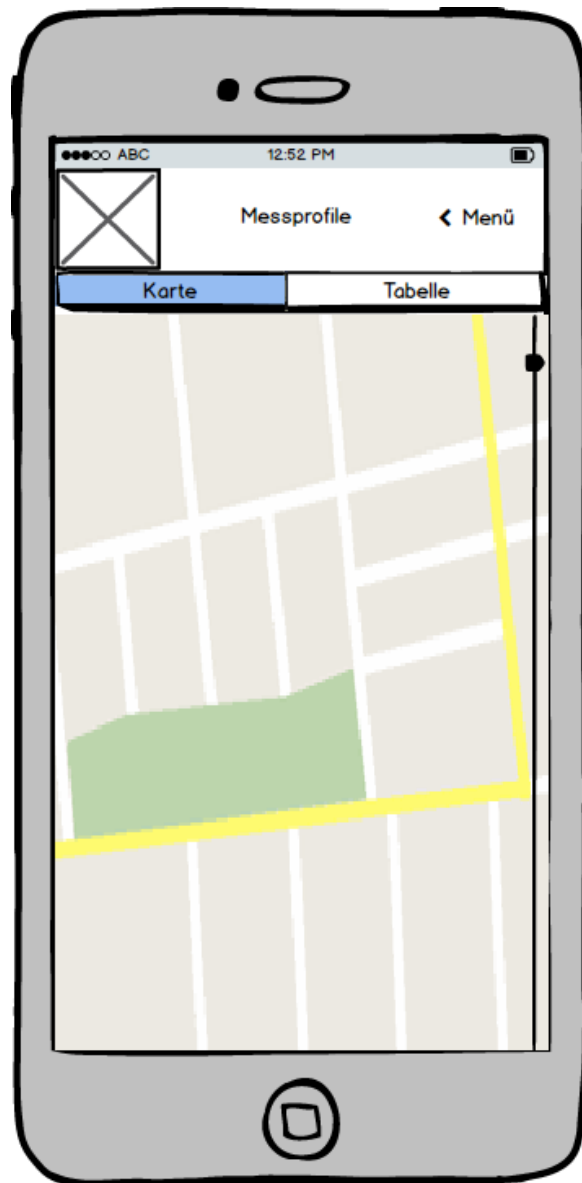


Abbildung 9.6: Messwerte anzeigen (Karte)

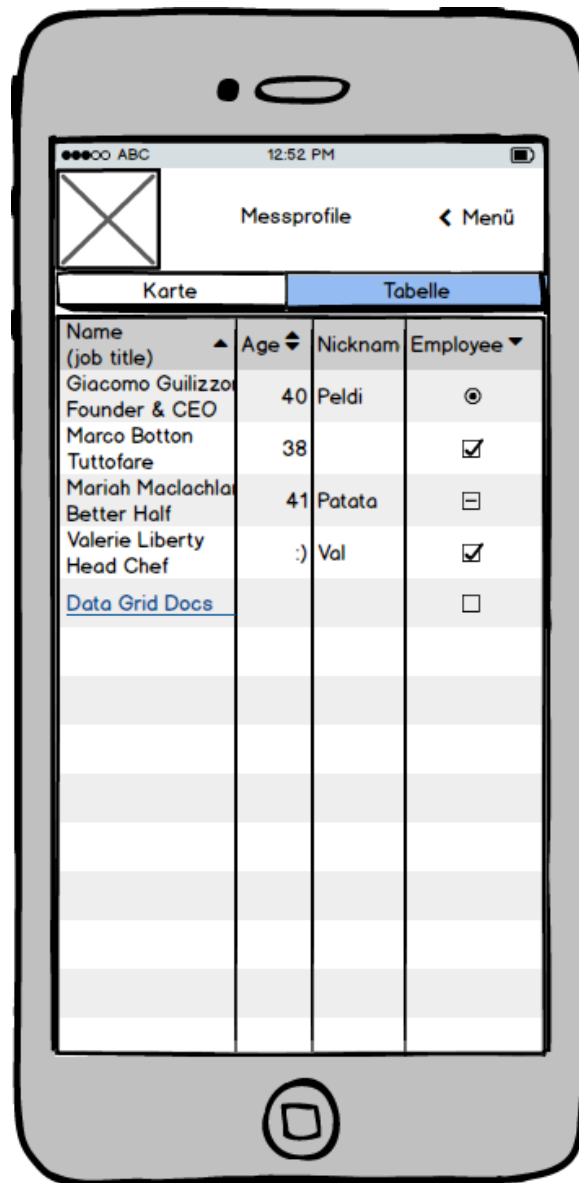


Abbildung 9.7: Messwerte anzeigen (Tabelle)

## 10 Entwicklungsumgebung

- Die App wird mit der Entwicklungsumgebung XCode entwickelt werden.
- Die Drohnenaspekte werden mit der Entwicklungsumgebung DJI Mobile SDK
- Die Anbindung der Sensoren erfolgt mithilfe des Bosch XDKs und der zugehörigen Entwicklungsumgebung

## **11 Abschlussbewertung**