

Pflichtenheft
"Drone Tour Visualizations"

Authors:

Heiko Dreyer, Johannes Gleichauf, Artuk Kakhovov, Martin
Olles, Nicolas-Andreas Tamm-Garetto, Hilmi-Can Yumak

May 2, 2018

Version: 3.0

Contents

1 Zielbestimmung	3
1.1 Musskriterien	3
1.2 Wunschkriterien	3
1.3 Abgrenzungskriterien	3
2 Produkteinsatz	4
2.1 Anwendungsbereiche	4
2.2 Zielgruppen	4
2.3 Betriebsbedingungen	4
3 Produktübersicht	4
4 Produktfunktionen	7
4.1 Funktionale Anforderungen	7
5 Produktdaten	9
5.1 Dateidaten	9
5.2 Bilddaten	9
5.3 Kartendaten	9
6 Produktleistungen	9
7 Qualitätsanforderungen	9
8 Benutzeroberfläche	9
9 Nichtfunktionale Anforderungen	10
10 Technische Produktumgebung	10
10.1 Hardware	10
10.2 Software	10
10.3 Orgware	10
10.4 Produktschnittstellen	10
11 Spezielle Anforderungen an die Entwicklungsumgebung	11
12 Ergänzungen	11
13 Glossar	11
14 Referenzierte Dokumente	11
14.1 Software Projekt	11
14.2 Android Dokumentation	12
14.3 JAVA Dokumentation	12
14.4 R Dokumentation	12
14.5 Git Dokumentation	12
14.6 QGIS Dokumentation	12
14.7 Cesium Dokumentation	12
14.8 Litchi Dokumentation	12
14.9 AR Pro3 Dokumentation	12
14.10 Betriebsanleitung Drohnen	12
14.10.1 Mavic Pro	12
14.10.2 Parrot Bebop	12
14.11 OpenSource Lizenzen	12

1 Zielbestimmung

Neben der Beobachtung von Vögeln und der Analyse ihres Flugverhaltens ist es wichtig, Änderungen an ihrer Umgebung festzustellen bzw. ihre Umgebung mittels Kartographierung festzuhalten. Die auf diese Weise gewonnenen Erkenntnisse können genutzt werden, um Korrelationen zwischen dem Flugverhalten, den Vögeln und ihrer Umgebung zu ermitteln.

Das Max-Planck-Institut für Ornithologie benötigt deshalb für Ihre Forschungen eine App, die ihren komplexen Workflow vereinfacht und generalisiert. Das entsprechende Gebiet soll mit einer Drohne abgeflogen und parallel dazu Luftaufnahmen gemacht werden. Zusätzlich wird eine Berechnung der korrekten Flugpunkt-Abstände und der approximativ optimalen Flugroute benötigt.

Ziel des Projektes ist eine höchstmögliche Automatisierung des Workflows durch das Projektteam. Gleichzeitig sollen Möglichkeiten zur Verbesserung der Kompatibilität, bspw. zu anderen Drohnen, Benutzerfreundlichkeit, sowie benutzerdefinierte Einstellungen im Programm integriert werden.

1.1 Musskriterien

- 1.1.1 Zeichnen eines Polygons um die zu kartographierende Umgebung auf einer Map festzulegen. (Eingrenzung des zu fotografierenden Areals)
- 1.1.2 Generierung von Tourpunkten
- 1.1.3 Berechnung der (approximativ) optimalen Tour für das Polygon (Travelling Salesman Problem, kürzester Weg)
- 1.1.4 Luftaufnahmen des Gebiets (80-85% Länge und 70-75% Breite Überlappung bei Bildern)
- 1.1.5 CSV Export der generierten Tour (und evtl. Transfer zu den entsprechenden Apps (z.B: Litchi Online))
- 1.1.6 Standpunkt des Piloten muss wählbar sein
- 1.1.7 Endpunkt der Route soll nahe am Startpunkt liegen
- 1.1.8 Drohne darf nicht höher als 100m in Relation zum Boden fliegen

1.2 Wunschkriterien

- 1.2.1 Integration von Cesium in den Workflow (evtl. Server/Client, Kommunikation mit App)
- 1.2.2 Support für mehr als 99 Tourpunkte
- 1.2.3 Aufteilung des Gebiets in gleich große Teilgebiete mit jeweils 99 Tourpunkten und Travelling Sales Problem für jedes der Teilgebiete
- 1.2.4 Die Anwendung soll eine Warnung ausgeben, falls die Drohne eine Entfernung von 300m überschreitet
- 1.2.5 Optimierung der Anzahl der aufgenommenen Bildern

1.3 Abgrenzungskriterien

- 1.3.1 Software überprüft keine Akkulaufzeit
- 1.3.2 Software ist nicht für die Steuerung der Drohne zuständig
- 1.3.3 Software soll keine Eingrenzungen, wie Straßen oder Privatgrundstücke erkennen

2 Produkteinsatz

Das Produkt dient zur Vereinfachung des Workflows für die Routenberechnung zum Kartographieren von Landschaften. Mit der Android Anwendung wird dem Benutzer ermöglicht, ein Polygon in eine topographische Karte einzuzichnen. Das Ziel der Anwendung ist es, für den Benutzer die optimalste Route für das Gebiet zu berechnen, damit diese Fläche effizient mit einer Drohne überflogen werden kann. Dabei sollen Bilder an bestimmten Routenpunkten aufgenommen werden, aus denen später eine hochauflösende Karte erstellt werden kann.

2.1 Anwendungsbereiche

2.1.1 Die Android Anwendung richtet sich speziell auf die Kartographierung im Bereich der Ornithologie. Die Ornithologie befasst sich z.B. mit der Untersuchung des Vogelzuges und welche Lokalisationen Vögel bevorzugen könnten. Das Ziel dieser Anwendung ist es, die Umgebung von Vögeln besser analysieren zu können.

2.2 Zielgruppen

2.2.1 Max Planck Institut für Ornithologie

2.2.2 Organisationen, Landwirte, Architekten, Landschaftsvermesser und Privatpersonen, die an optimalen Flugrouten für ihre Drohne interessiert sind und effizient ihre Flächen kartographieren möchten. Dabei ist die fehlerfreie Bedienung eines Smartphones oder Tablets mit Android Voraussetzung.

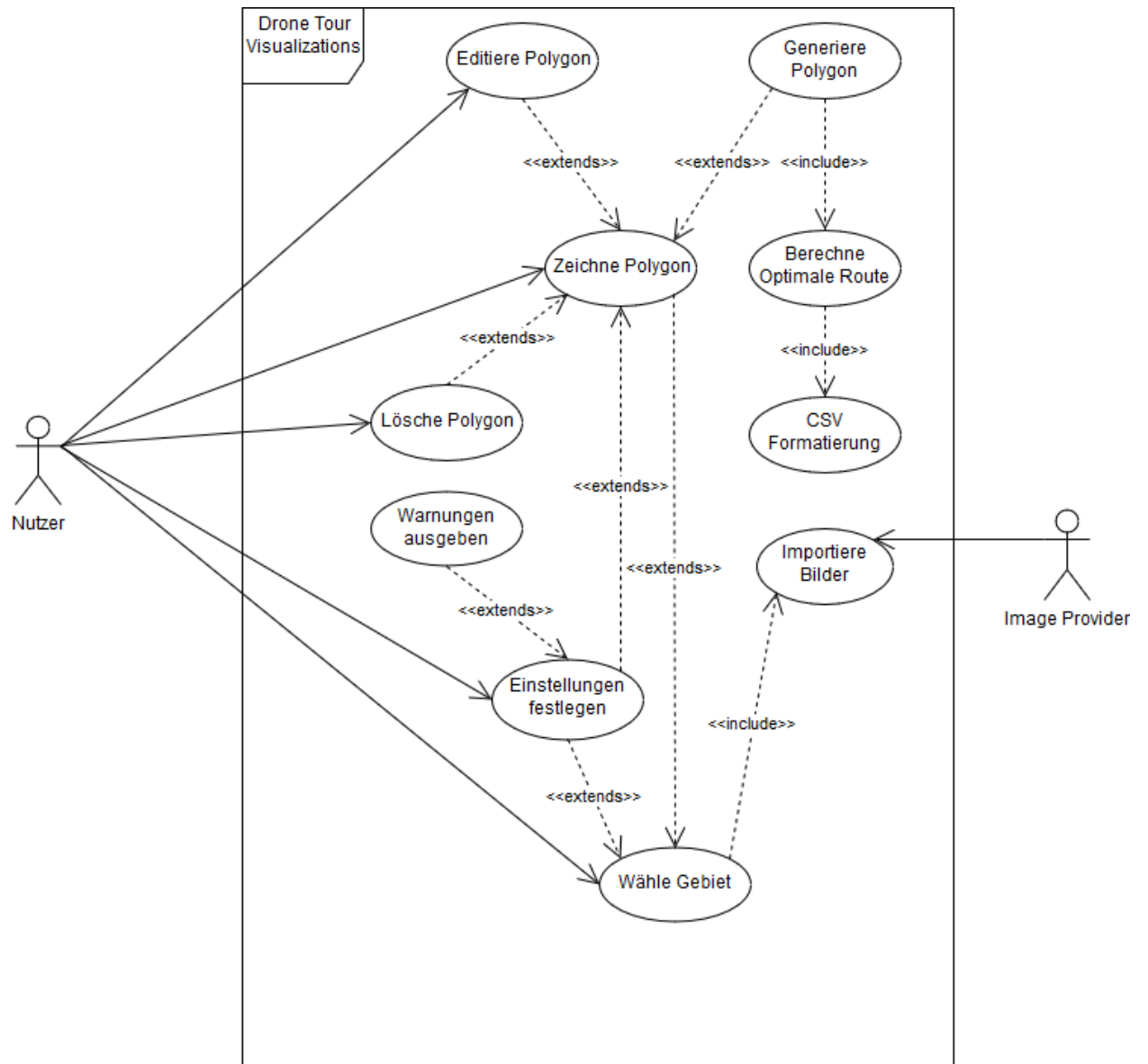
2.3 Betriebsbedingungen

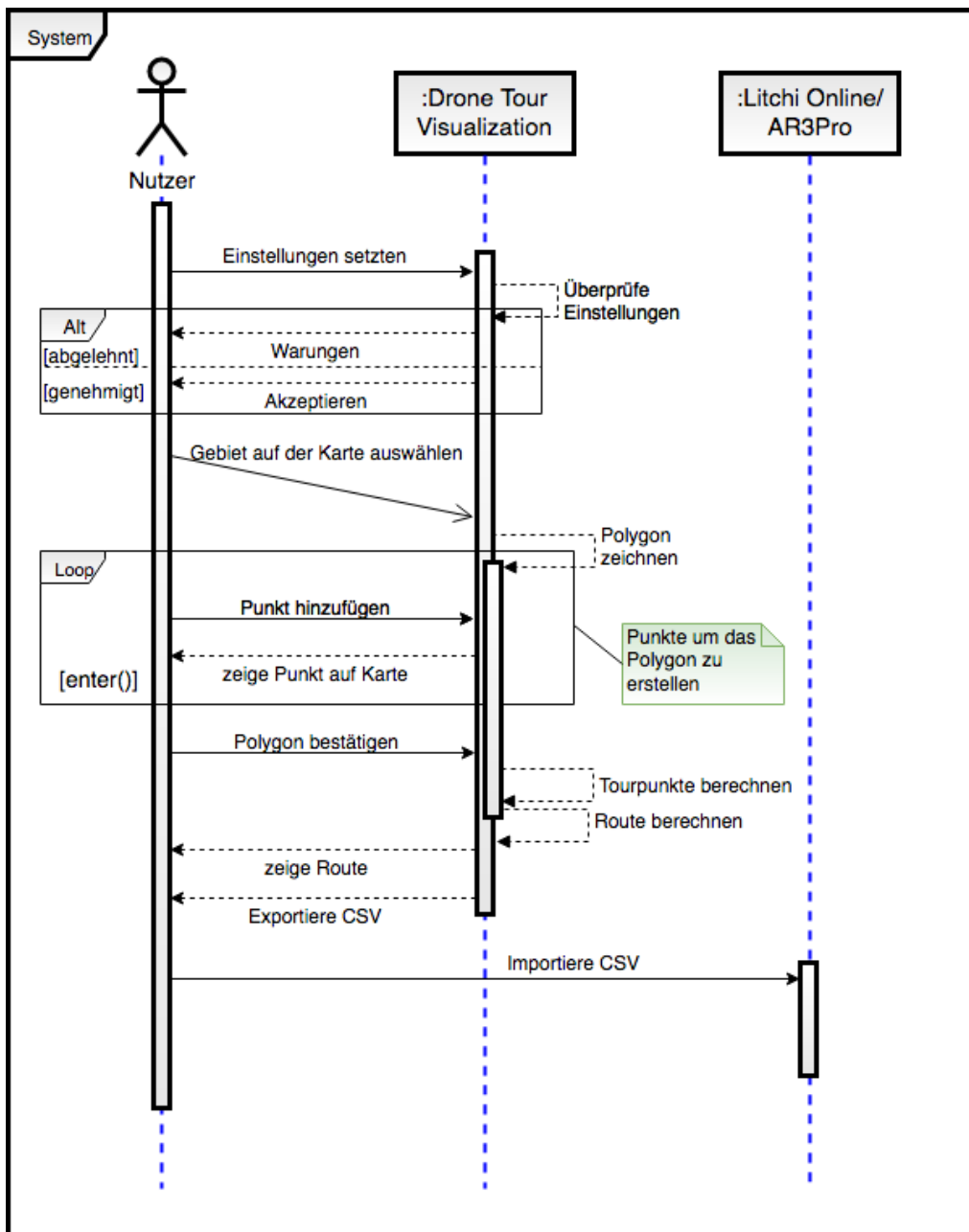
2.3.1 Die Android Anwendung wird unter einer OpenSource Lizenz entwickelt werden. Der Quelltext ist öffentlich und von Dritten einsehbar. Sie kann verändert und auch von jedem kostenlos genutzt werden.

2.3.2 Das Produkt bietet jedem die Möglichkeit, diese anzuwenden zu können und richtet sich somit an eine unbegrenzte Anzahl an Benutzern.

3 Produktübersicht

Mit der Anwendung können Touren für den Drohnenflug gezeichnet werden. Der Benutzer zeichnet auf einem androidfähigen Endgerät ein Polygon auf einer Karte ein und bekommt anschließend eine möglichst effiziente Flugtour berechnet. Der Anwender kann Touren erstellen, ändern, löschen und speichern. Zusätzlich gibt der Benutzer Informationen wie Flughöhe und Kameraeinstellungen an. Am Ende des Vorgangs kann der Anwender eine CSV-Datei exportieren, die dann für Litchi Online verwendet werden kann um eine MissionHub zu generieren. Das folgende Use-Case Diagramm beschreibt die Nutzung des Systems durch den Nutzer.





Das obige Sequenzdiagramm zeigt die zeitliche Abfolge von der Polygonerstellung bis hin zum erfolgreichen Export der CSV-Datei. Der Benutzer startet die App auf seinem Smartphone und gibt seine Einstellungen, wie FOV und Flughöhe ein. Die Anwendung reagiert mit einer Warnung, falls Richtlinien verletzt werden oder akzeptiert diese. Nach diesem Schritt kann der Benutzer das Gebiet auf der Karte aussuchen, auf der er das Polygon zeichnen möchte. Er kann Punkte hinzufügen, ändern oder löschen. Wenn der Benutzer mit der Erstellung des Polygons fertig ist, kann er dieses bestätigen und die Anwendung berechnet die optimale Route der Tour und generiert die CSV-Datei, die der Benutzer dann zur Weiterverwendung benutzen kann.

4 Produktfunktionen

4.1 Funktionale Anforderungen

4.1.1 PolygonEingabe

Funktion: Das System soll den Input vom Benutzer als ein Polygon interpretieren.

Beschreibung: Die Fläche, die wir fotografieren wollen, wird vom Benutzer als ein Polygon eingegeben. Hierbei wird der Benutzer die Knoten eingeben und das resultierende Polygon wird danach berechnet.

Quelle: Besprechung mit Dr. Sommer 2018

Abhängigkeiten: -

4.1.2 PolygonLöschen

Funktion: Das erstellte Polygon kann einfach gelöscht werden, um dieses neu einzeichnen zu können.

Beschreibung: Wenn die zu kartographierende Fläche zu groß ist oder man eine ganz andere Fläche eingrenzen möchte, bietet diese Funktion die Möglichkeit das Polygon zu löschen um von vorne beginnen zu können.

Quelle: Besprechung mit Dr. Sommer 2018

Abhängigkeiten: -

4.1.3 PolygonEditieren

Funktion: Das System bietet die Funktion das Polygon zu bearbeiten und zu verändern.

Beschreibung: Wenn Änderungen am Polygon vorgenommen werden sollen, kann man diese dann ganz leicht ausführen ohne das Polygon gleich löschen zu müssen.

Quelle: Besprechung mit Dr. Sommer 2018

Abhängigkeiten: -

4.1.4 FOVEinstellung

Funktion: Im System soll man Field of View festlegen können.

Beschreibung: Mit Hilfe von FOV berechnet das System die andere Variable und informiert den Benutzer darüber. Zum Beispiel: Wenn der Benutzer sich für eine Pixelgröße von 3 cm mit ein FOV von 70 Grad entscheidet, muss die resultierende Höhe ausgegeben werden.

Quelle: Anne Scharf Vortrag 24/04/18

Abhängigkeiten: -

4.1.5 HöhenEinstellung

Funktion: Im System soll man Höhe festlegen können.

Beschreibung: Mit Hilfe von der Höhe berechnet das System die andere Variable und informiert den Benutzer darüber. Zum Beispiel: Wenn der Benutzer sich für eine Höhe von 100m mit ein FOV von 70 Grad entscheidet, muss die resultierende Pixelgröße ausgegeben werden.

Quelle: Anne Scharf Vortrag 24/04/18

Abhängigkeiten: -

4.1.6 Bildgröße

Funktion: Das System soll die Bildgröße berechnen und ausgeben.

Beschreibung: Anhand der Daten, die bereits berechnet und eingegeben wurden, muss die Bildgröße berechnet werden.

Quelle: Besprechung mit Dr. Sommer 2018

Abhängigkeiten: 4.1.1

4.1.7 TourpunkteBerechnung

Funktion: Das System soll sicherstellen, dass die Bilder mit ausreichend vertikaler und horizontaler Überlappung aufgenommen werden.

Beschreibung: Für das Sticking benötigt man Aufnahmen, die sich ausreichend überlappen. Wir haben uns für die von der Literatur empfohlenen Überlappungen von 80-85% in der Länge und 70-75% in der Breite entschieden.

Quelle: Anne Scharf Vortrag 24/04/18

Abhängigkeiten: 4.1.1 - 4.1.4

4.1.8 Standpunkt des Piloten, Start- und Endpunkt

Funktion: Das System soll nach der Position des Benutzers fragen und diesen in der Karte anzeigen.

Beschreibung: Es ist wichtig für weitere Anforderungen zu wissen, wo der Pilot während des Flugs stehen wird, da dieser Start- und Endpunkt der Route sein wird.

Quelle: Anne Scharf Vortrag 24/04/18

Abhängigkeiten: 4.1.1 - 4.1.3

4.1.9 Warnungen

Funktion: Das System soll den Benutzer warnen, wenn die Flugeinstellungen widerrechtlich sein könnten.

Beschreibung: Das deutsche Gesetz setzt für das Drohnenfliegen viele Regulationen fest. Diese beschränken die Größe des Polygons und die Flughöhe. Wenn die Einstellungen gegen diese Regulationen verstoßen, muss der Benutzer informiert werden.

Quelle: Anne Scharf Vortrag 24/04/18

Abhängigkeiten: 4.1.1 - 4.1.5

4.1.10 Karten

Funktion: Das System soll Karten verfügbar machen.

Beschreibung: Für die Haupt-Activity des Systems sind Karten erforderlich. Diese müssen nicht offline zur Verfügung gestellt werden, aber man könnte versuchen, diese offline zur Verfügung zu stellen.

Quelle: Besprechung mit Dr. Sommer 2018

Abhängigkeiten: -

4.1.11 TravelingSalesman

Funktion: Das System soll die effektivste Route berechnen.

Beschreibung: Mit dem gegebenen Polygon soll das System jetzt die effektivste Route planen, wobei alle schon erwähnten Beschränkungen beachtet werden müssen.

Quelle: Besprechung mit Dr. Sommer 2018

Abhängigkeiten: 4.1.1 - 4.1.8

4.1.12 CSV-Export

Funktion: Das System soll diese Route (4.1.11) in CSV formatieren.

Beschreibung: Für Litchi-Online und AR Pro3 ist dieses das erwünschte Format.

Quelle: Anne Scharf Vortrag 24/04/18

Abhängigkeiten: 4.1.1 - 4.1.8, 4.1.10, 4.1.11

5 Produktdaten

Die Anwendung benötigt langfristig Speicherplatz für die zu speichernden Daten der Benutzer. Zu speichernde Daten sind Dateidaten bei der Erstellung der Touren. Dazu gehören Eingabedaten der Benutzer, das Zeichnen des Polygons, die Koordinaten der zu kartographierenden Fläche und die daraus folgende CSV-Datei mit der optimalen Tour. Zusätzlich speichert die Anwendung importierte Karten, damit diese jederzeit zur Verfügung stehen. Die gezeichneten Polygone auf der Karte werden als Bilddatei auf dem Smartphone abgespeichert, damit diese später noch einsehbar sind.

5.1 Dateidaten

Daten ID, Datei Name, Datei Pfad, Anzeige Name, Datei Beschreibung

5.2 Bilddaten

Bild ID, Datei Name, Datei Name Thumbnail, Datei Pfad, Anzeige Name, Datei Beschreibung, Bildkommentar

5.3 Kartendaten

Karten ID, Datei Name, Datei Pfad, Anzeige Name, Datei Beschreibung

6 Produktleistungen

6.1 Anwendung sollte speichereffizient gestaltet werden

6.2 Die Generierung der Routen sollte zeiteffizient sein

7 Qualitätsanforderungen

Die Applikation soll verschiedene qualitative Anforderungen für den optimalen und vielseitigen Einsatz erfüllen. Wir geben in dieser Sektion eine Übersicht über bestimmte Qualitätsmerkmale und deren Bewertung für unsere Applikation.

Qualität	sehr gut	gut	normal	nicht relevant
Funktionalität	×			
Zuverlässigkeit		×		
Benutzbarkeit	×			
Effizienz			×	
Änderbarkeit		×		
Übertragbarkeit	×			

Table 1: Qualitätsanforderungen

8 Benutzeroberfläche

Die Benutzeroberfläche ist auf die Touchbedienung auszulegen.

Das System soll aus einer Folge von Android Activities bestehen. Geplant sind drei grundlegende Activities: Ein Startmenü, eine Eingabekarte (um die Anforderungen 4.1.1-4.1.8 zu erfüllen) und eine Activity, welche einen Overview über das erstellte Polygon bietet und die Möglichkeit zum Export des Polygons hat.

Im Startmenü gibt es die Möglichkeit ein Polygon zu zeichnen, was einen zu der zweiten Activity führt: In der zweiten Activity kann das Polygon gezeichnet und der Standpunkt des Piloten festgelegt werden. Anschließend wird die optimale Route, anhand der berechneten Tourpunkte, berechnet und man gelangt dadurch zu der dritten Activity. Dort sieht man einen Overview über das Polygon mitsamt der berechneten Route und hat die Möglichkeit entweder nochmal in die zweite Activity zurückzugehen, um das Polygon abzuändern oder man kann das Polygon als CSV-Datei exportieren.

9 Nichtfunktionale Anforderungen

9.1 Die Anwendung wird unter einer OpenSource Lizenz herausgebracht

9.2 Lizenzen bei der Erstellung der Software müssen kompatibel sein

10 Technische Produktumgebung

10.1 Hardware

10.1.1 Smartphone/Tablet mit Android zum Testen und Ausführen der Software

10.1.2 PC zum Entwickeln der Software

10.2 Software

10.2.1 Android Studio

10.2.2 GitLab

10.2.3 QGIS

10.2.4 Cesium

10.2.5 Litchi-Online/ Litchi App

10.2.6 AR Pro3

10.2.7 draw.io (UML)

10.2.8 R-Studio

10.2.9 Eclipse

10.3 Orgware

10.3.1 Dji Marvic Pro Drohne

10.3.2 Parrot Bebop 1 Drohne

10.4 Produktschnittstellen

In dieser Sektion führen wir die Anwendungen auf, mit denen unsere Applikation zur Laufzeit kommunizieren muss. Darunter fällt:

10.4.1 Map Provider

10.4.2 AR Pro3

11 Spezielle Anforderungen an die Entwicklungsumgebung

- 11.1 Die Anwendung sollte kompatibel mit der Androidversion 7.0 sein, dazu wird Android Studio 2.3.1 sowie das Java SE Development Kit verwendet.
- 11.2 GitLab dient als Versionskontrolle und zum Austausch von benötigten Informationen bei der Entwicklung der Anwendung.

12 Ergänzungen

Die Software wird in erster Linie in Zusammenarbeit mit dem Max-Planck-Institut für Ornithologie entwickelt. Weitere Einsatzmöglichkeiten in verschiedenen Anwendungsgebieten sind denkbar. Man beachte aber, dass die zu entwickelnde Anwendung nur für die erwähnten Drohnen in 10.3 gelten, aber für andere Drohnen ausgebaut werden könnten.

13 Glossar

In dieser Sektion geben wir eine Übersicht über alle verwendeten Fachbegriffe, Abkürzungen und Definitionen. Dabei werden diese auch verständlich erklärt.

Begriff	Erläuterung
MPI	Max-Planck-Institut
CSV	Dateiformat
Cesium	JavaScript Bibliothek für 3D Karten
QGIS	Freies Geoinformationssystem
App	Applikation
OpenSource	Software mit öffentlichem Quelltext
Smartphone	Mobiltelefon mit umfangreichen Funktionen
Tablet	Tragbarer, kleiner Computer
Android	Betriebssystem für mobile Geräte
Android Studio	Entwicklungsumgebung für Android
Android Activities	Bildschirmseiten, aus denen die App besteht
ID	Identifikationsnummer
Litchi App	App für nahezu autonome Drohnenflüge
GitLab	Anwendung für Versionsverwaltung
UML	Unified Modeling Language
Eclipse	OpenSource Entwicklungsumgebung
R-Studio	Entwicklungsumgebung für die Programmiersprache R
JAVA	Programmiersprache
TSP	Traveling salesman problem
FOV	Field of view (Sichtfeld)
Thumbnail	Vorschaubild
Ornithologie	Die Wissenschaft, die sich mit Vögeln auseinandersetzt
Stiching	Erstellen einer großen Fotografie aus verschiedenen kleineren Einzelaufnahmen

Table 2: Glossartabelle

14 Referenzierte Dokumente

14.1 Software Projekt

<https://drive.google.com/drive/folders/1W0iksTtZYItUAbhXTdifFKhTn5r1clsN>
<https://drive.google.com/drive/folders/11J2QuRuT3WhTfWYQ1S8fe1XV8ZAhP-W>

14.2 Android Dokumentation

<https://developer.android.com/guide/>
<https://developer.android.com/studio/intro/>

14.3 JAVA Dokumentation

<https://docs.oracle.com/javase/8/docs/api/>

14.4 R Dokumentation

<https://support.rstudio.com/hc/en-us/categories/200035113-Dokumentation>
<https://cran.r-project.org/web/packages/rgdal/rgdal.pdf>
<https://cran.r-project.org/web/packages/TSP/TSP.pdf>

14.5 Git Dokumentation

<https://git-scm.com/doc>

14.6 QGIS Dokumentation

https://docs.qgis.org/testing/en/docs/user_manual/index.html

14.7 Cesium Dokumentation

<https://cesiumjs.org/Cesium/Build/Documentation/>

14.8 Litchi Dokumentation

<https://flylitchi.com/help#about>

14.9 AR Pro3 Dokumentation

http://ee.shellware.com/ARPro3/ARPro3_Manual.pdf

14.10 Betriebsanleitung Drohnen

14.10.1 Mavic Pro

http://www.produktinfo.conrad.com/datenblaetter/1500000-1599999/001508989-an-01-de-DJI_MAVIC_PRO.pdf

14.10.2 Parrot Bebop

https://www.generationrobots.com/media/DroneBebopdeParrot/Bebop-drone_User-guide_DE.pdf

14.11 OpenSource Lizenzen

<https://www.wbs-law.de/it-recht/open-source-lizenzen-ein-uberblick-35752/>
<https://www.it-economics.de/blog/2016-3/pt9oi6kzodgcl1tmsfc1yn1vzq4usb>

15 Zeitplan

- 01.05.2018: Pflichtenheft inkl. Projektzeitplan + jeweilige Abgaben der Milestones
- 15.05.2018: Milestone 1 / Abgabe der System Design Spezifikation
 1. Inputhandling
 2. Horizontaler Prototyp
 3. Konzeptionelle Fertigstellung der App
- 05.06.2018: Milestone 2 / Zwischenbewertung
 1. Alle Musskriterien vorhanden
 2. Etwaige Wunschanforderungen
 3. Testing
- 10.07.2018: Milestone 3
 1. Projekt fertig
 2. Präsentation vorbereiten
- **17.07.2018:** Endpräsentation