System Design Document
"Drone Tour Visualizations"

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

May 17, 2018

Version: 2.3

Contents

1 Changelog

Version	Autor	Beschreibung	Datum
1.0	HCY	Erstellung des ersten Entwurfs (Englisch)	02.05.2018
1.1	HD	Übersetzung (Deutsch)	03.05.2018
1.2	MO	Changelog hinzugefügt	06.05.2018
1.3	HCY	"3. Aktuelle Architektur" hinzugefügt Glossartabelle hinzugefügt	07.05.2018
1.4	HD, JG, AK, MO, NATG, HCY	Sequence Diagram hinzugefügt und Texte in Rahmenbedingungen hinzugefügt	09.05.2018
1.5	HD, JG, AK, MO, NATG, HCY	Component-, Deployment- und Statechart-Diagramm hinzugefügt	14.05.2018
1.6	JG, NATG	Deployment-Diagramm überarbeitet	14.05.2018
1.7	HD	Statechart Beschreibung hinzugefügt	14.05.2018
1.8	AK	Klassendiagram und Beschreibung hinzugefügt	14.05.2018
1.9	MO	Rechtschreib- und Grammatikfehler behoben Beschreibung Component-Diagramm hinzugefügt	15.05.2018
2.0	HCY	Diagramme überarbeitet/ Anpassungen vorgenommen	15.05.2018
2.1	JG	Deployment Diagram und Beschreibung überarbeitet	16.05.2018
2.2	NATG	Component Diagram und Beschreibung überarbeitet	16.05.2018
2.3	HD	Statechart Diagram überarbeitet	16.05.2018

Legende:
HD = Heiko Dreyer
JG = Johannes Gleichauf
AK = Artuk Kakhorov
MO = Martin Olles
NATG = Nicolas-Andreas Tamm-Garetto
HCY = Hilmi-Can Yumak

2 Einführung

In diesem Abschnitt des SDD geben wir eine grobe Beschreibung des Inhalts. Wir beschreiben den Zweck, unsere Ziele bezüglich des Designs, unsere Referenzen, einen Überblick über die Gliederung des Dokuments und im Abschnitt 2.3 Definitionen, Akronyme und Abkürzungen, welche wichtige Termini beschreibt, die im ganzen Dokument genutzt werden.

2.1 Zweck des Systems

Erstellung einer mobilen Applikation zur Kartographie. Die Applikation ist in erster Linie dazu gedacht den Workflow des MPI für Ornithologie in Radolfzell zu automatisieren und zu vereinfachen.
Wie im Pflichtenheft erwähnt, ist jedoch der Einsatz in unterschiedlichen Gebieten durchaus denkbar.

2.2 Gestaltungsziele

In diesem Dokument geben wir eine ausführliche Beschreibung aller Attribute und Funktionen, die für die Implementierung der Anwendung erforderlich sind, an. Wie man dem Pflichtenheft entnehmen kann, soll das System im Grunde genommen folgende Funktionen erfüllen:

- Bietet die Möglichkeit ein Polygon zu zeichnen, um die zu kartographierende Umgebung festzulegen
- Generierung der optimalen Route, mit der das Gebiet abgeflogen wird
- Festlegung verschiedenster Kameraeinstellungen
- Ausgabe von Warnungen, falls Einstellungen widerrechtlich sind
- Ausgabe einer CSV-Datei

Folgende Funktionen werden nicht durch das System bereitgestellt:

- Keine Überprüfung der Akkulaufzeit
- Software ist nicht für die Steuerung der Drohne zuständig
- Software soll keine Eingrenzungen, wie Straßen oder Privatgrundstücke erkennen
- Kein Schutz vor widerrechtlichen Einstellungen

Für eine detaillierte Übersicht empfehlen wir, die Anforderungen im Pflichtenheft zu betrachten.

2.3 Definitionen, Akronyme und Abkürzungen

In der folgenden Tabelle geben wir eine Übersicht über alle im Dokument verwendeten Definitionen, Akronyme und Abkürzungen mitsamt einer kurzen Erklärung.

Begriff	Erläuterung
MPI	Max-Planck-Institut
Ornithologie	Vogelforschung
SDD	System Design Document
Applikation	Anwendungsprogramm
CSV	Dateiformat
IEEE	Standardisierung von Techniken, Hardware und Software
QGIS	Freies Geoinformationssystem
R	Programmiersprache
FOV	Field of View (Sichtfeld)
Litchi	Anwendung für nahezu autonome Drohnenflüge
OpenDroneMap	Anwendung, die kleine Einzelaufnahmen zu einer großen Karte zusammenfügt
Android	Betriebssystem für mobile Geräte
App	Applikation
Loadingscreen	Ladebildschirm mit Logo, während die Applikation geladen wird
User	Benutzer
Android Activity	Bildschirmseiten, aus denen die App besteht
API	Programmierschnittstelle
AR Pro3	App für nahezu autonome Drohnenflüge
Map Provider	Anbieter eines Diensts oder einer Ressource (Karten)
Interface	Schnittstelle

Table 1: Glossartabelle

2.4 Quellen

- Pflichtenheft
- IEEE 1016-2009

2.5 Übersicht

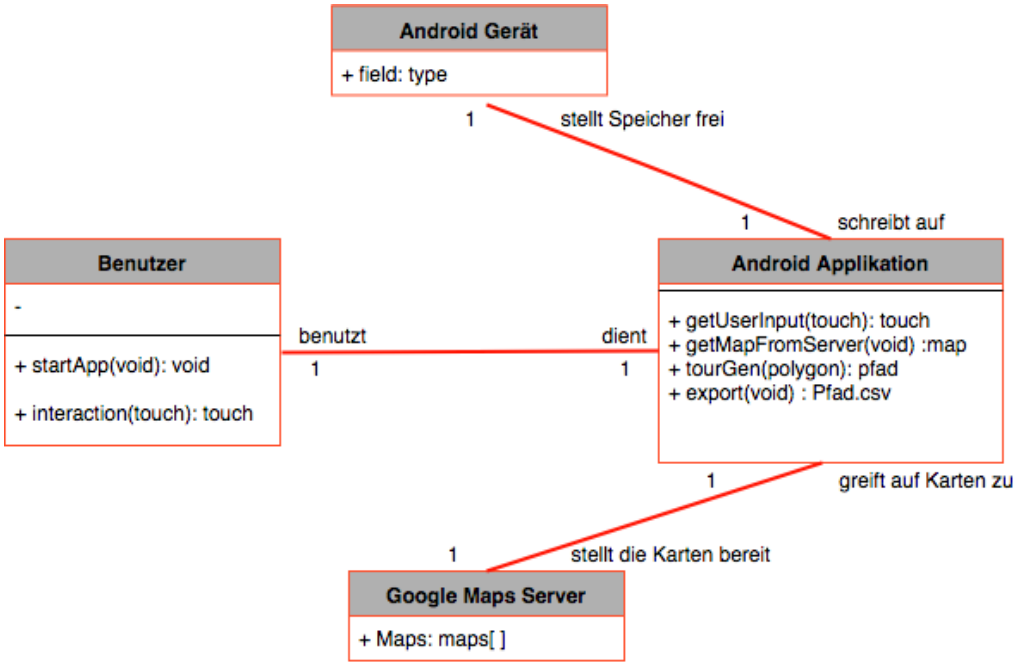
Das SDD ist in Übereinstimmung mit dem IEEE-Standard in drei Kapitel unterteilt, im folgenden geben wir eine kurze Beschreibung jedes einzelnen Kapitels:

3 Aktuelle Architektur

Aktuell ist noch kein bestehendes System verfügbar, aber es besteht ein einfacher jedoch ineffizienter Workflow von Frau Dr. Scharf, die viele unterschiedliche Systeme für die Tourgenerierung verwendet. Mit QGIS wird das zu kartographierende Gebiet ausgesucht, worauf dann ein Polygon gezeichnet wird. Diese Polygon-Daten werden dann von einem R-Skript für die optimale Tourgenerierung verwendet, dabei werden Tourpunkte für die Luftaufnahmen festgelegt. Die Berechnung der Tourpunkte erfolgt anhand der Flughöhe und des FOV, damit bei den Luftaufnahmen ausreichend Überlappung der Bilder vorhanden ist. Nachdem diese Tourpunkte generiert wurden, wird anhand des Traveling Salesman Algorithmus die optimal kürzeste Route berechnet, welche dann als CSV-Datei mit den benötigten Kamera- und Flugeinstellungen exportiert wird. Diese CSV-Datei wird dann in Litchi Online importiert und über die Litchi App ausgeführt. Als letzter Schritt werden die Luftaufnahmen in OpenDroneMap zu einer Map zusammengefügt.

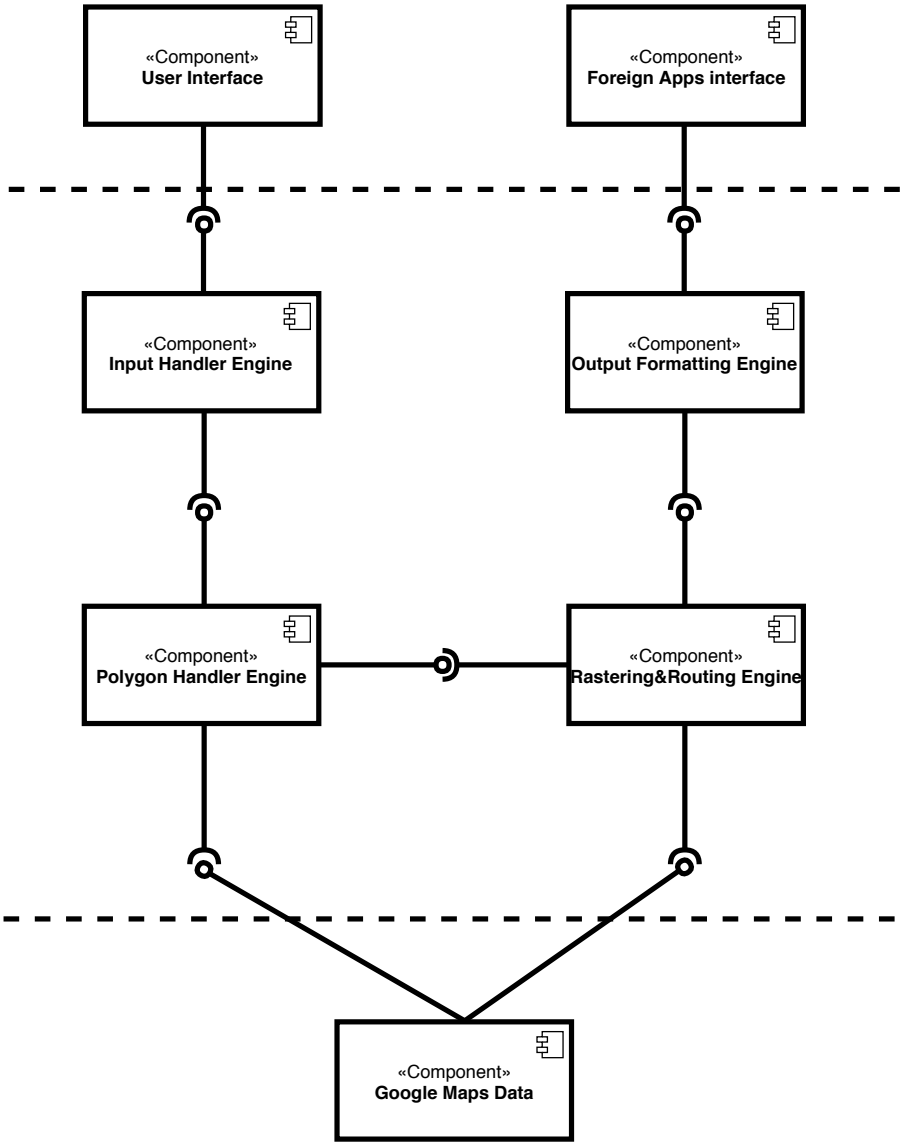
4 Geplante Architektur

4.1 Übersicht



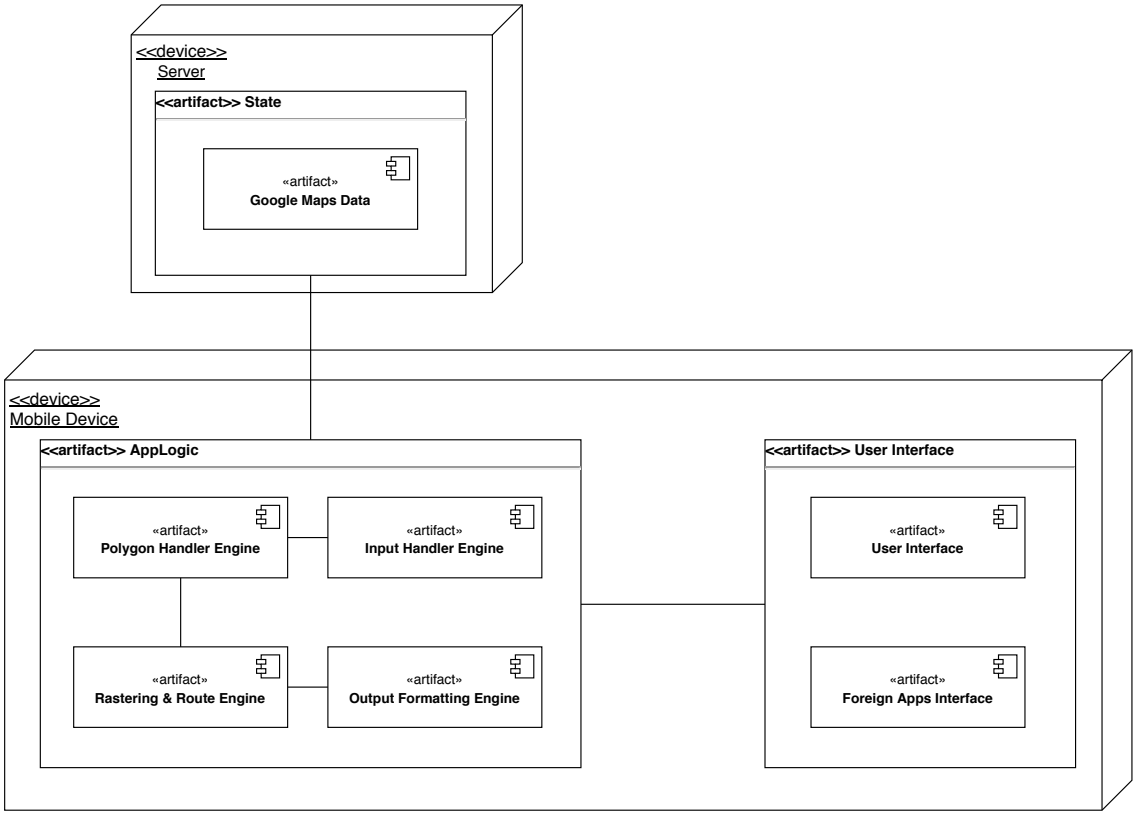
Das Klassendiagramm dient zur Veranschaulichung der unterschiedlichen Teile des Systems. Der Benutzer soll die App sowohl starten, als auch bedienen können. Im Laufe der Interaktion greift die App auf die von dem Google Maps Server bereitgestellten Karten zu. Um den optimalsten Pfad zu speichern, soll ein Zugriff auf den internen Speicher gewährleistet werden.

4.2 Einzelbetrachtung der Subsysteme



Das *Component Diagram* beschreibt, wie das System unterteilt wird. Wir folgen dem *User-BusinessLogic-State* architectural pattern, welches wie folgt angewendet wird: Unten steht die State-Komponente, in der Mitte die interne Logik und oben die Interfaces mit den Usern und anderen Apps.
Der Datenfluss sieht folgendermaßen aus: Von der Google Maps API erhalten wir die benötigten Daten, die danach durch unsere interne Logik und Parametrisierung mit den Einstellungen des Users ein Resultat für den User liefern. Die gestrichelte Linien bezeichnen die Architekturgrenzen.

4.3 Hard- und Software-Zuordnung



Dieses Diagramm beschreibt, wie die verschiedenen Kompnenten aus dem *Component Diagram* auf der Hardware-Ebene umgesetzt werden.
Wir teilen die Komponenten auf in externe Komponenten und Komponenten, die auf dem Smartphone laufen.
Extern auf einem, von uns unabhängigen Server, liegen die Daten von der Google Maps API.
Die Google Maps API versorgt uns mit kartographischen Daten, welche in den internen Komponenten verwendet werden, um durch einen User-Input ein Polygon zu erstellen. Google Maps liefert hierfür die Koordinaten der Eckpunkte. Aus dem Polygon werden die Routenpunkte für die Drohne berechnet. Für die Routenpunkte wird eine annähernd optimale Route berechnet. Diese Route wird dann dem Nutzer im App Interface angezeigt.

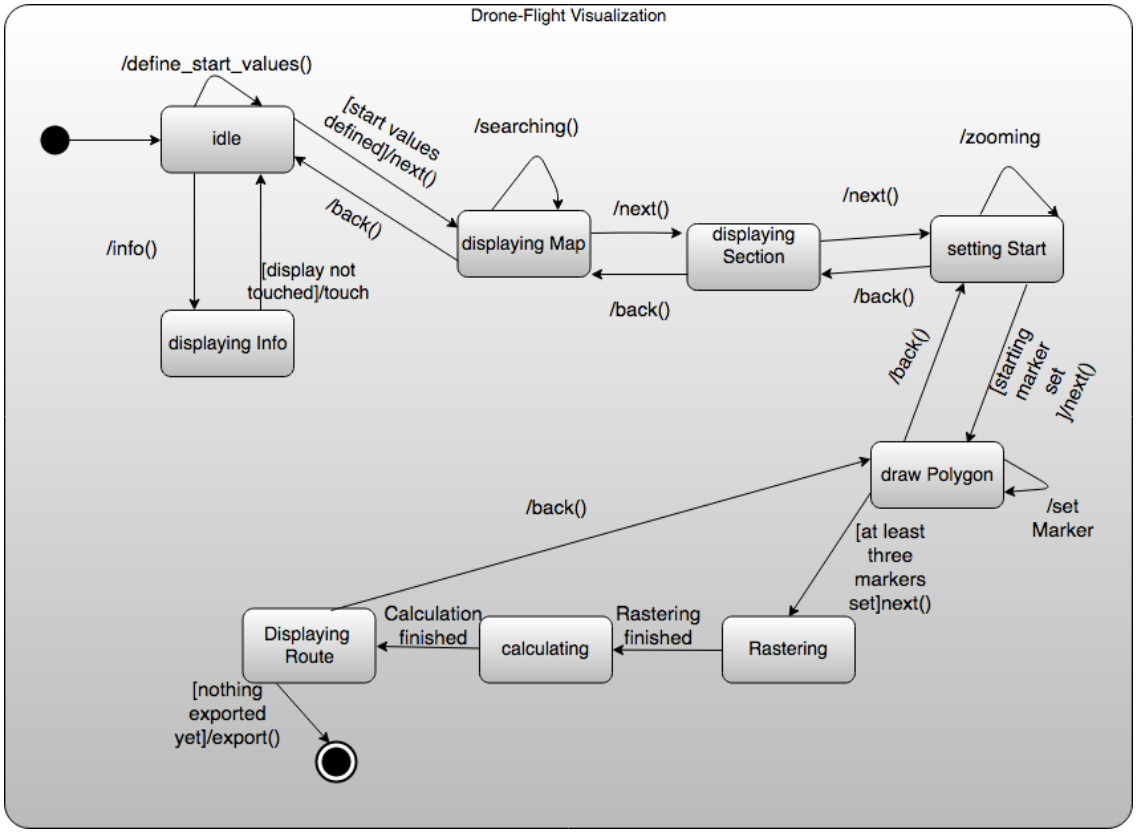
4.4 Persistente Datenverwaltung

Es wird keine Datenbank verwendet, da keine Daten von den Usern gespeichert werden. Berechnungen und Übergabe von Parametern finden intern statt und müssen nicht längerfristig gespeichert werden. Die Anwendung generiert und speichert lediglich die CSV-Output-Datei in dem von der Anwendung vorhergesehenen Pfad.

4.5 Zugangskontrolle und Sicherheit

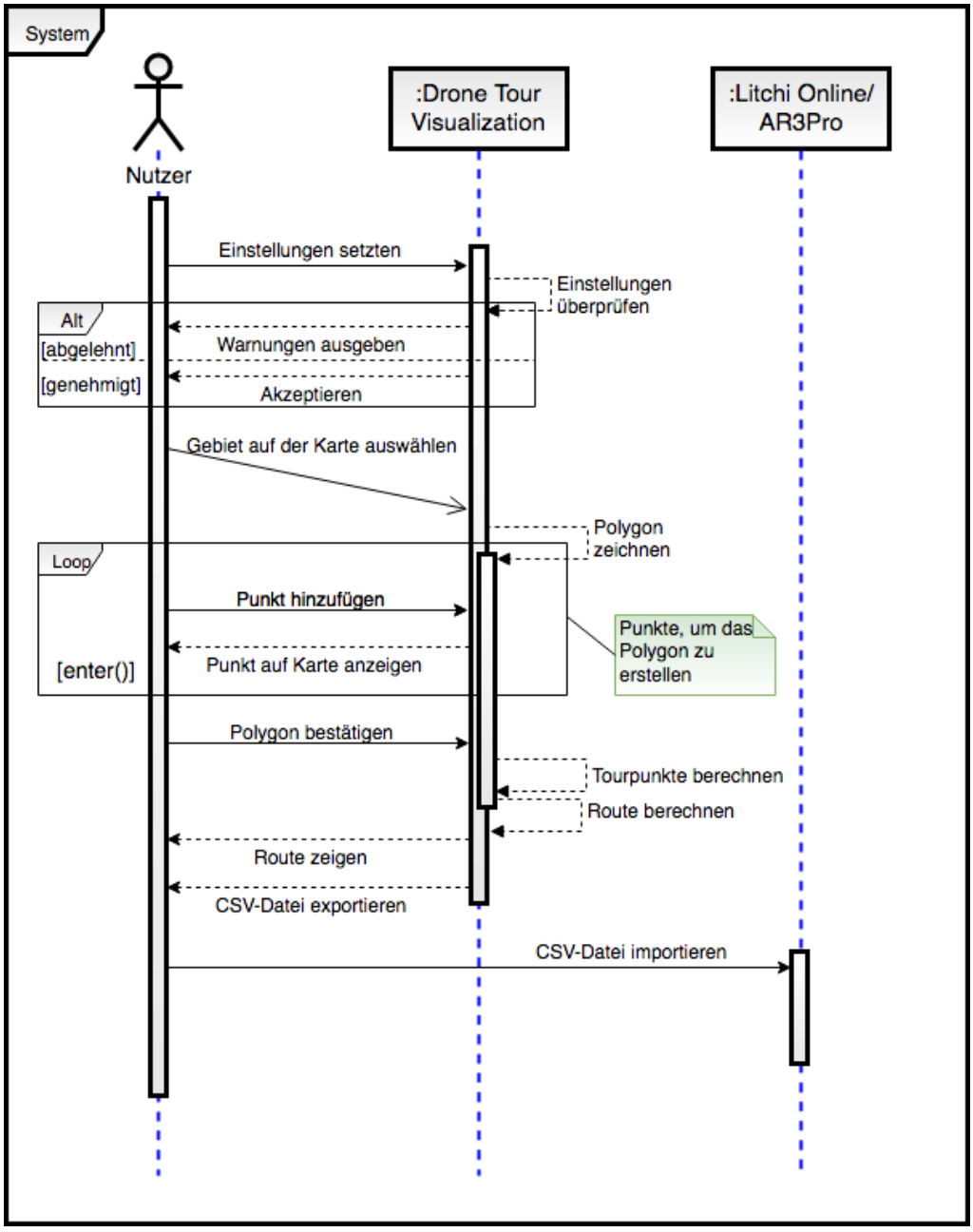
Für die Anwendung der Applikation ist keine Authentifizierung, Verschlüsselung und Schlüsselverwaltung notwendig, da für die Benutzung keine Zugangsdaten der jeweiligen Benutzer gespeichert werden. Jeder, der die Applikation verwendet, hat die gleichen Zugriffsrechte. Damit steht sie für jeden zur freien Verfügung.

4.6 Globale Softwarekontrolle



Das Statechart Diagramm beschreibt die globale Verwaltung der Software. Es werden keine Ereignisse parallel ausgeführt, sondern die Nutzung der Applikation verläuft rein sequenziell. Wenn der Nutzer die Applikation startet, wartet diese auf eine Aktion des Nutzers. Der Nutzer kann nun entweder Informationen abrufen oder sich durch das Betätigen des Next-Buttons die Karte des Map Providers ansehen, wodurch sich ebenfalls der Zustand des Systems ändert.
Im nächsten Zustand sieht der Nutzer nur noch einen Ausschnitt der Karte mit einem gewissen Toleranzrahmen. Nun ist es entweder möglich zurückzukehren, um einen anderen Ort zu wählen oder den Startpunkt zu setzen. Durch Betätigen des Next-Buttons kann der Nutzer nun beginnen das Polygon zu zeichnen, der Zustand bleibt so lange erhalten, wie der Nutzer am Zeichnen ist, betätigt der Nutzer wieder den Next-Button beginnt das System mit der Berechnung der Route und gibt diese dann auf dem Bildschirm aus.
Durch einen Klick auf den Back-Button kommt der Nutzer zu dem Zustand zurück, indem er das Polygon zeichnen kann. Der Nutzer hat nun die Möglichkeit auf Export zu drücken, was dazu führt, dass die Applikation die CSV Datei generiert.

4.7 Rahmenbedingungen



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.

4.7.1 Startverhalten

Man startet die Android-App und gelangt zum Loadingscreen, wo der User mit einem Logo begrüßt wird. Danach gelangt der User zu unserer ersten Activity.

4.7.2 Verhalten beim Herunterfahren

Beim Herunterfahren wird die App nach Androidstandards geschlossen.

4.7.3 Verhalten bei Fehlern

Der User bekommt Warnungen zu gewissen Rahmenbedingungen. Bei einem Fehler in der Benutzung der App wird ein Fehler geworfen und die Activity, welche nicht korrekt ausgeführt wurde, wird neu geladen.

4.7.4 Schnittstellen

1. Google Maps API
2. Litchi Online
3. AR Pro3
4. Litchi App