

# POSITION PREDICTION BASED ON MOVEMENT DATA

## SOFTWARE REQUIREMENTS SPECIFICATION

*Autoren*

Timo Jockers, Oliver Mänder, Benjamin Moser,  
Manuel Prinz, Sebastian Strumbelj, Simon Suckut

# Contents

<b>1</b>	<b>Einleitung</b>	<b>3</b>
1.1	Überblick . . . . .	3
1.2	Zweck dieses Dokumentes . . . . .	3
1.3	Produktbeschreibung . . . . .	3
1.4	Definitionen und Abkürzungen . . . . .	4
1.5	Referenzen . . . . .	4
<b>2</b>	<b>Produktkontext</b>	<b>5</b>
2.1	Externe Produkte . . . . .	5
2.2	Nutzercharakterisierung . . . . .	5
2.3	Szenario . . . . .	5
2.4	Annahmen und Abhängigkeiten . . . . .	6
2.5	Einschränkungen . . . . .	7
<b>3</b>	<b>Mindestanforderungen</b>	<b>8</b>
3.1	Funktionen . . . . .	8
3.1.1	Datenmanagement . . . . .	8
3.1.2	Vorhersageberechnung . . . . .	9
3.1.3	Visualisierung . . . . .	10
3.1.4	Nutzerinteraktion . . . . .	12
3.2	Nicht-funktionale Anforderungen . . . . .	14
3.2.1	Hardware . . . . .	14
3.2.2	Implementierung . . . . .	14
3.2.3	Robustheit . . . . .	15

3.2.4	Performanz . . . . .	15
<b>4</b>	<b>Optionale Anforderungen</b>	<b>16</b>
<b>5</b>	<b>Nicht-Anforderungen</b>	<b>18</b>
<b>6</b>	<b>Projektzeitplan</b>	<b>19</b>
<b>7</b>	<b>Diagramme</b>	<b>21</b>
7.1	Anwendungsfälle . . . . .	21
7.2	Abläufe . . . . .	22

# **1 Einleitung**

## **1.1 Überblick**

Dieses Dokument ist in fünf Abschnitte aufgeteilt. Im ersten Abschnitt (Kapitel 1) wird der Zweck dieses Dokuments erläutert und der Funktionsumfang der Applikation grob beschrieben. Desweiteren werden verwendete Definitionen und Abkürzungen sowie weitere referenzierte Dokumente aufgeführt. Im zweiten Abschnitt (Kapitel 2) wird der Kontext beschrieben, in dem das Produkt verwendet werden soll. Darunter fallen das Verhältnis zu anderen Produkten, die Nutzercharakteristika sowie mögliche Abhängigkeiten und Einschränkungen. Der dritte Abschnitt (Kapitel 3 bis 5) enthält eine detaillierte Auflistung der Anforderungen. Im vierten Abschnitt (Kapitel 6) wird der Projektzeitplan, insbesondere die Meilensteine, festgelegt. Der letzte Abschnitt (Kapitel 7) enthält die wesentlichen UML-Diagramme.

## **1.2 Zweck dieses Dokumentes**

Dieses Dokument dient zur Fixierung der erhobenen Anforderungen und Einschränkungen an die zu entwickelnde Applikation. Damit liefert es eine Grundlage und Referenz für den Entwicklungsprozess sowie für die Überprüfung des Systems auf Vollständigkeit nach Fertigstellung. Für den Kunden legt dieses Dokument fest, welcher Funktionsumfang realisiert werden wird.

## **1.3 Produktbeschreibung**

Die zu entwickelnde Android-Applikation trifft, basierend auf Informationen aus der Vergangenheit, eine Vorhersage über den Aufenthaltsort eines Vogels zu einem gegebenen Zeitpunkt in der Zukunft.

Dabei kann der Nutzer wählen, welche der verfügbaren Daten und Verfahren in die Vorhersage einfließen.

Das Ergebnis der Vorhersage wird in einer zwei- oder dreidimensionalen Karten-

landschaft visualisiert.

## 1.4 Definitionen und Abkürzungen

**App, Applikation** Soweit nicht anders angemerkt wird hiermit die zu entwickelnde Android-Applikation bezeichnet.

**Forscher** Personen, die wissenschaftlich das Verhalten von Vögeln untersuchen.

**Mobile Daten** Ein Smartphone kann sich entweder über ein WiFi-Netzwerk oder über ein mobiles Datennetzwerk (3G, LTE, ...) mit dem Internet verbinden.

**Vorhersage-Modell/Algorithmus** Die zur Berechnung einer Vorhersage verwendete Methodik.

**API** *Application Programming Interface*, hier insbesondere Schnittstellen mit Providern wie *Movebank* oder *Cesium*.

**Outdoor-Modus, Normal-Modus** Die Applikation verfügt über zwei verschiedene Funktionsmodi. Einer davon ist der sog. Outdoor-Modus. (Siehe Anforderung (R16).)

**Cesium** Ist eine Javascript-Bibliothek für die Arbeit mit dreidimensionalen Kartendaten.

## 1.5 Referenzen

- Begleitdokument zum Softwareprojekt SS 2018
- Cesium API-Dokumentation

## 2 Produktkontext

### 2.1 Externe Produkte

Die Applikation soll Forschern eine Vorhersage bieten können, wo sie einen Vogel mit einer gewissen Wahrscheinlichkeit in naher Zukunft antreffen werden.

In der Vergangenheit aufgezeichnete Positionsdaten werden dabei von *Movebank* bezogen. Das Ergebnis der Vorhersage wird mittels *Cesium* und *Offline-Karten* visualisiert.

### 2.2 Nutzercharakterisierung

Die Nutzer der Applikation sind Forscher, die das Verhalten von Vögeln untersuchen.

Der Nutzer verfügt über mindestens gute Erfahrung im Umgang mit Smartphones und deren Bedienung.

Der Nutzer hat im Allgemeinen eine Vorstellung vom Umfang der verwendeten Datenmengen, der Komplexität der Analyse und ist sich bewusst, dass entsprechende Ergebnisse - je nach Gerät und technischen Voraussetzungen - mehrere Sekunden dauern können.

Der Nutzer hat Erfahrung im Umgang mit geographischen Daten. Er kennt grundlegende Visualisierungstechniken und deren Interpretation.

### 2.3 Szenario

Der folgende Text beschreibt beispielhaft ein Szenario, wie die Applikation verwendet wird.

Herr Herr Storch ist ein Forscher am Max-Planck-Institut für Ornithologie. Herr Storch beschäftigt sich mit der Analyse von Tracking-Daten von Vögeln. Für das Tracking werden Vögel mit Sendern ausgestattet.

Herr Storch möchte nun einen bestimmten in der Vergangenheit über einen solchen Sender verfolgten Vogel nun wieder auffinden, etwa um den Sender auszutauschen oder um das Tier zu betrachten.

Herr Storch befindet sich in seinem Büro und hat Zugriff auf eine schnelle Datenverbindung. Herr Storch startet die Applikation auf seinem Smartphone im Normal-Modus und gibt die Vogel-ID ein. Er benötigt noch etwas Zeit zur Vorbereitung und für das Zurücklegen des Weges, also fragt er eine Vorhersage für eine dreiviertelstunde in der Zukunft an.

Die Vorhersage wird Herr Storch in der *Cesium*-Umgebung visualisiert. Es wird ein Bereich angezeigt, in dem der Vogel mit hoher Wahrscheinlichkeit anzutreffen ist sowie weitere Abstufungen die einer geringeren Wahrscheinlichkeit entsprechen. Mithilfe der dreidimensionalen Kartendarstellung macht sich Herr Storch außerdem mit der Beschaffenheit des Geländes vertraut. Währenddessen speichert die Applikation die Daten der Vorhersage und 2D-Kartendaten für spätere Verwendung.

Nun bricht Herr Storch auf. Da das Gebiet recht abgelegen ist verfügt sein Smartphone lediglich über eine langsame mobile Datenverbindung und an manchen Stellen über gar keine Internetverbindung. Herr Storch wechselt nun in den Outdoor-Modus. Die Applikation greift nun zur Visualisierung der Vorhersage auf die vorher heruntergeladenen 2D-Karten zurück.

Die aktuelle Position von Herr Storch wird mittels des GPS-Sensors seines Smartphones ebenfalls in der Visualisierung angezeigt. Damit nähert sich Herr Storch dem Zielgebiet an und wird bei der Vogelsuche unterstützt.

## 2.4 Annahmen und Abhängigkeiten

- Die verwendeten Schnittstellen (*Movebank*, *Cesium*, ...) sind zu jedem Zeitpunkt verfügbar.
- Eine für die Abfrage der Daten aus *Movebank* ausreichende Identifikation des gesuchten Vogels ist vorhanden.
- Werden für die Datenabfrage aus *Movebank* weitere Schritte oder Informationen benötigt (Zugangsdaten, vorherige Anfrage bei Provider der Daten, ...) so werden diese vom Nutzer erledigt bzw. bereitgestellt.

- Es wird bis zu Milestone 2 ein rudimentäres Vorhersagemodell auf Basis von Positionsdaten aus der Vergangenheit implementiert. Im Laufe der Entwicklung wird in Absprache mit Betreuern und Kunde ermittelt, welche weiteren Daten und welche weiteren Vorhersagemodelle sinnvoll einbezogen werden können.

## 2.5 Einschränkungen

- Die Applikation soll unter einer Open-Source-Lizenz entwickelt werden. Wird Code oder werden Bibliotheken von Dritten verwendet, so müssen die Lizenzen kompatibel sein.
- Die Abfrage von Daten von Schnittstellen kann limitiert sein.
- Die Internetverbindung des Gerätes kann während der Nutzung für kurze oder längere Zeit ausfallen.
- Bei Verwendung des mobilen Datennetzes kann nur eine kleine Übertragungsgeschwindigkeit verfügbar sein.
- Von *Movebank* bezogene Daten können unvollständig oder falsch formatiert sein.
- Die Software soll aus Modulen aufgebaut werden. Diese Module sollen möglichst unabhängig voneinander sein.
- Der Programmcode soll kommentiert und dokumentiert werden.



## 3 Mindestanforderungen

### 3.1 Funktionen

#### 3.1.1 Datenmanagement

##### (R1) DatenLadenAusMovebank

**Funktion:** Daten sollen aus der Movebank geladen werden.

**Beschreibung:** Das Laden der für die Berechnung benötigten Daten geschieht ausschließlich über die Movebank, die die gegebenen Daten zur Verfügung stellt.

**Quelle:** Treffen am 26.04.2018

**Abhängigkeiten:** -

##### (R2) SpeicherungMovebankNutzerdaten

**Funktion:** Nutzerdaten für die Movebank sollen in den Einstellungen eingetragen werden können.

**Beschreibung:** Die App soll in den Einstellungen eine Möglichkeit anbieten, Movebank-Nutzerdaten eintragen zu können, um auch Daten aus der Movebank zu laden, die nur mit einem dortigen Account abrufbar sind.

**Quelle:** Treffen am 26.04.2018

**Abhängigkeiten:** (R1)

##### (R3) OfflineDatenSpeicherung

**Funktion:** Die App soll für die aktuelle Vorhersage Offline-Daten speichern.

**Beschreibung:** Die App soll die aktuelle Vorhersage lokal speichern, damit man diese auch ohne aktiven Internetzugang betrachten kann.

**Quelle:** Treffen am 26.04.2018

**Abhängigkeiten:** (R5)

(R4) AktualisierungUeberMobileDaten

**Funktion:** Über mobile Daten soll die aktuelle Vorhersage aktualisiert werden können.

**Beschreibung:** Die App soll auch über mobile Daten aktuelle Daten aus der Movebank empfangen und für eine aktuelle Vorhersage verwenden können.

**Quelle:** Treffen am 30.04.2018

**Abhängigkeiten:** (R9), (R1), (R12), (R5), (R6), (R14)

### 3.1.2 Vorhersageberechnung

(R5) BerechnungDesAufenthaltsraumes

**Funktion:** Gegeben die StudyID und VogelID soll ein wahrscheinlicher Aufenthaltsraum berechnet werden.

**Beschreibung:** Für einen Vogel (und Study??) soll berechnet wie wahrscheinlich es ist, diesen in einem vom Algorithmus bestimmten Gebiet anzutreffen.

**Quelle:** Treffen am 26.04.2018

**Abhängigkeiten:** (R1)

(R6) VorhersageFuerBestimmtenZeitraum

**Funktion:** Die Positionsvorhersage soll für einen gegebenen Zeitraum berechnet werden.

**Beschreibung:** Die Positionsvorhersage soll für einen gegebenen Zeitraum ab dem aktuellen Moment berechnet werden.

**Quelle:** Treffen am 26.04.2018

**Abhängigkeiten:** (R1)

(R7) WaehlbarerVorhersagezeitraum

**Funktion:** Der Nutzer soll den Zeitraum für die Vorhersage wählen können.

**Beschreibung:** Der Nutzer soll die Möglichkeit haben, einzustellen für welchen Zeitraum ab dem aktuellen Zeitpunkt die Vorhersage berechnet wird.

**Quelle:** Treffen am 26.04.2018

**Abhängigkeiten:** -

(R8) WahlDerDatenFuerVorhersage

**Funktion:** Der Nutzer soll einstellen können auf welchen Daten die Vorhersage basiert.

**Beschreibung:** Der Nutzer soll über Parameter die Berechnung der Vorhersage justieren können.

**Quelle:** Treffen am 26.04.2018

**Abhängigkeiten:** (R1)

### 3.1.3 Visualisierung

(R9) VisuelleDarstellungDerVorhersage

**Funktion:** Die Vorhersagen sollen visuell dargestellt werden.

**Beschreibung:** Die Berechnungen zur Vorhersage der wahrscheinlichen Vogelposition sollen in der App angezeigt werden.

**Quelle:** Treffen am 26.04.2018

**Abhängigkeiten:** (R1)

(R10) Kompass

**Funktion:** Es soll ein Kompass angezeigt werden, der zum gesuchten Vogel zeigt.

**Beschreibung:** Eine Visualisierung soll anzeigen, in welcher Richtung ausgehend vom aktuellen Standpunkt der gesuchte Vogel am wahrscheinlichsten anzutreffen ist.

**Quelle:** Treffen am 02.05.2018

**Abhängigkeiten:** (R1)

(R11) AktuellePosition

**Funktion:** Es soll die aktuelle Position angezeigt werden.

**Beschreibung:** Eine Visualisierung soll auf der dargestellten Kartenlandschaft anzeigen, wo sich der Nutzer momentan befindet.

**Quelle:** Treffen am 02.04.2018

**Abhängigkeiten:** (R13), ??

(R12) VisualisierungDerVorhandenenDaten

**Funktion:** Die vorhandenen Daten für einen gegebenen Vogel und Zeitraum sollen visualisiert werden.

**Beschreibung:** Wird ein Vogel ausgewählt, so sollen dessen vorhandene Daten bis zu einer gewählten maximalen Zeitgrenze in der Vergangenheit angezeigt werden können.

**Quelle:** Treffen am 26.04.2018

**Abhängigkeiten:** (R1)

(R13) VisualisierungInCesiumBeiNormalModus

**Funktion:** Im Normal-Modus sollen die Daten in Cesium visualisiert werden.

**Beschreibung:** Befindet sich die Applikation im Normal-Modus, so sollen die getrackten Positionen und die Vorhersagen mit Cesium visualisiert werden.

**Quelle:** Treffen am 26.04.2018

**Abhängigkeiten:** (R1), (R12), (R5), (R6)

(R14) VisualisierungBeiOutdoorModus

**Funktion:** Im Outdoor-Modus sollen die Daten visualisiert werden.

**Beschreibung:** Befindet sich die Applikation im Outdoor-Modus, so sollen die getrackten Positionen und die Vorhersagen visualisiert werden.

**Quelle:** Treffen am 30.04.2018

**Abhängigkeiten:** (R1), (R12), (R5), (R6)

(R15) VisualisierungAufOfflineKarten

**Funktion:** Im Outdoor-Modus sollen die Daten auf offline-Karten visualisiert werden.

**Beschreibung:** Falls offline-Karten vorhanden sind, sollen diese für die Visualisierung der Vorhersage und der getrackten Positionen verwendet werden.

**Quelle:** Treffen am 30.04.2018

**Abhängigkeiten:** (R1), (R12), (R5), (R6)

### 3.1.4 Nutzerinteraktion

(R16) Modi

**Funktion:** Die Applikation soll über einen Normal- und einen Outdoor-Modus verfügen.

**Beschreibung:** Die Applikation soll über zwei Modi bzw. Zustände verfügen mit ähnlichem aber ungleichem User Interface und Funktionalität.

**Quelle:** Treffen am 02.05.

**Abhängigkeiten:** -

(R17) WechselDesModus

**Funktion:** Der Nutzer soll zwischen dem Normal- und dem Outdoor-Modus wechseln können.

**Beschreibung:** Der Nutzer soll durch Bedienung des User Interfaces zwischen dem Normal und dem Outdoor-Modus wechseln können.

**Quelle:** Treffen am 02.05

**Abhängigkeiten:** (R16)

(R18) TouchBasierteBenutzeroberflaeche

**Funktion:** Die App soll über eine touch-basierte Benutzeroberfläche verfügen.

**Beschreibung:** Die App soll über eine touch-basierte Benutzeroberfläche bedienbar sein.

**Quelle:** Treffen am 26.04.2018

**Abhängigkeiten:** -

(R19) SucheNachVogel

**Funktion:** In der Software soll nach einem Vogel gesucht werden können.

**Beschreibung:** Der Benutzer soll anhand einer ID nach einem bestimmten Vogel suchen können.

**Quelle:** Treffen am 26.04.2018

**Abhängigkeiten:** -

(R20) HinweisAufFormalFalscheDaten

**Funktion:** Der Nutzer soll auf einen hohen Anteil falsch formatierter Daten hingewiesen werden.

**Beschreibung:** Die App soll dem Nutzer einen Hinweis anzeigen, wenn in den der Berechnung der Vorhersage zugrunde liegenden Daten so viele Datensätze falsch formatiert sind, dass die Qualität der Vorhersage nicht vergleichbar mit der auf Basis eines vollständig korrekten Datensatzes ist.

**Quelle:** Treffen am 30.04.2018

**Abhängigkeiten:** (R1)

## 3.2 Nicht-funktionale Anforderungen

### 3.2.1 Hardware

(R21) KorrekteArchitekturDerSmartphones

**Funktion:** Die App sollte auf Smartphones mit Android 5.1 oder höher, Internet-Zugang und einer armeabi, armeabi-v7a oder arm64-v8a Architektur lauffähig sein.

**Beschreibung:** Die App sollte auf Smartphones mit Android 5.1 oder höher, Internet-Zugang und einer armeabi, armeabi-v7a oder arm64-v8a Architektur lauffähig sein.

**Quelle:** Treffen am 30.04.2018

**Abhängigkeiten:** -

### 3.2.2 Implementierung

(R22) EinheitlicheSchnittstelleFuerVorhersagealgorithmen

**Funktion:** Die Software soll über eine einheitliche Schnittstelle für Vorhersagealgorithmen verfügen.

**Beschreibung:** Zum einfachen Hinzufügen von weiteren Vorhersagealgorithmen soll die Software über ein einheitliches Interface für diese verfügen.

**Quelle:** Treffen am 26.04.2018

**Abhängigkeiten:** -

### 3.2.3 Robustheit

(R23) VerhaltenOhneInternetverbindung

**Funktion:** Die App sollte bei fehlender Internetverbindung mit eingeschränkter Funktionalität weiter nutzbar sein.

**Beschreibung:** Bricht die Verbindung zum Internet ab, sollte die App robust darauf reagieren. Einige Grundfunktionen sollten dabei weiter nutzbar sein.

**Quelle:** Treffen am 30.04.2018

**Abhängigkeiten:** -

(R24) RobustGegenueberFormalFalschenDaten

**Funktion:** Die App soll robust gegenüber formal falschen Daten aus der Movebank sein

**Beschreibung:** Die App soll formal falsche Daten, die sie aus der Movebank bekommt, so aufbereiten, dass das Berechnen und Visualisieren eines Ergebnisses möglich ist.

**Quelle:** Treffen am 26.04.2018

**Abhängigkeiten:** (R1)

### 3.2.4 Performanz



(R25) BerechnungsdauerVorhersage

**Funktion:** Der Berechnungsdauer der Vorhersage soll die vom Nutzer erwartete Dauer nicht deutlich überschreiten.

**Beschreibung:** Die App soll die Vorhersage in einem adäquaten Zeitraum berechnen und diese dem Nutzer möglichst schnell visualisieren.

**Quelle:** Treffen am 30.04.2018

**Abhängigkeiten:** (R9), (R1), (R12), (R5), (R6), (R8), (R24)

## 4 Optionale Anforderungen

Die folgenden Anforderungen sind optional und müssen nicht zwingend erfüllt werden.

(O1) VirtualReality

**Funktion:** Integration von *Virtual Reality*-Modul

**Beschreibung:** Es wird ein zur Verfügung gestelltes Modul in die Applikation integriert, dass die Visualisierungen in einem *Virtual Reality*-Raum visualisiert.

**Quelle:** Treffen am 26.04.2018

**Abhängigkeiten:** -

(O2) ZusaetzlicheParameter

**Funktion:** Es werden - neben den Positionsdaten aus der Vergangenheit - noch weitere Parameter für die Berechnung der Vorhersage verwendet.

**Beschreibung:** Weitere Parameter / Datenquellen fließen in die Berechnung der Vorhersage ein.

**Quelle:** Begleitdokument

**Abhängigkeiten:** -

(O3) GewichtungParameter

**Funktion:** Parameter, die in die Berechnung der Vorhersage einfließen sollen gewichtet werden können.

**Beschreibung:** Der Nutzer soll die Möglichkeit haben einzustellen, mit welchem Gewicht die jeweiligen Parameter in die Berechnung der Vorhersage einfließen.

**Quelle:** Begleitdokument

**Abhängigkeiten:** (R5)

(O4) VariableDarstellung

**Funktion:** Es sind verschiedene Arten der Visualisierung verfügbar.

**Beschreibung:** Der Nutzer hat die Möglichkeit, das Ergebnis der Vorhersage auf verschiedene Arten zu visualisieren.

**Quelle:** Begleitdokument

**Abhängigkeiten:** (R5), (R9)

(O5) Evaluation

**Funktion:** Eine Evaluation der Vorhersage ist möglich.

**Beschreibung:** Der Nutzer hat die Möglichkeit, das Ergebnis der Vorhersage zu bewerten.

**Quelle:** Begleitdokument

**Abhängigkeiten:** (R5)

(O6) MehrereVoegel

**Funktion:** Es kann eine Vorhersage für mehrere Vögel gemeinsam getroffen und visualisiert werden.

**Beschreibung:** Für eine gegebene Menge von Vögeln kann eine Vorhersage getroffen und eine Visualisierung erstellt werden.

**Quelle:** Begleitdokument

**Abhängigkeiten:** (R5), (R9)

(O7) Gattung

**Funktion:** Andere Tiere derselben Gattung werden in die Berechnung der Vorhersage einbezogen.

**Beschreibung:** Es werden die Daten von anderen Tieren derselben Gattung für die Berechnung der Vorhersage eines bestimmten Vogels verwendet.

**Quelle:** Begleitdokume

**Abhängigkeiten:** (O6), (O2)

(O8) AehnlichkeitenFinden

**Funktion:** Es werden Muster in den Daten aus der Vergangenheit für die Berechnung der Vorhersage verwendet.

**Beschreibung:** Die Analyse von Mustern in Daten aus der Vergangenheit fließt in die Berechnung der Vorhersage ein.

**Quelle:** Begleitdokument

**Abhängigkeiten:** (R5), (O2)

## 5 Nicht-Anforderungen

Die App soll lediglich ein einfaches Vorhersageverfahren beinhalten, und zusätzlich eine interne Schnittstelle zur Verfügung stellen, um die vom Kunden zur Verfügung gestellten Modelle/Algorithmen zu integrieren.

Es ist nicht Teil des Entwicklungsprozesses, ein komplexes Vorhersagemodell zu entwickeln.

Es werden keine Vorhersagen für lange Zeiträume (im Maßstab von mehreren Tagen) getroffen.

## 6 Projektzeitplan

### Milestone 1

Fertigstellung am 15.5.

- Grundliegende Software-Architektur festgelegt
- Grundliegendes Software-Design festgelegt
- Ein Grundgerüst der Applikation ist fertiggestellt: Es sind folgende Ansichten funktional umgesetzt:
  - Cesium (eine Kartenumgebung wird angezeigt, Navigation möglich)
  - Offline-Karten (eine Kartenumgebung wird angezeigt, Navigation möglich)
  - Diverse UI-Elemente (mindestens Eingabefeld, Button)
- Die Positionsdaten für die Vorhersage werden von der *Movebank*-API bezogen.

### Milestone 2

Fertigstellung am 5.6.

- Es wird eine Vorhersage mittels eines einfachen Vorhersagemodells berechnet.
- Es existiert eine funktionale Visualisierung des Vorhersage-Ergebnisses.
- Der Aufbau des User-Interfaces ist festgelegt (nicht-funktionale Mockups).

## Milestone 3

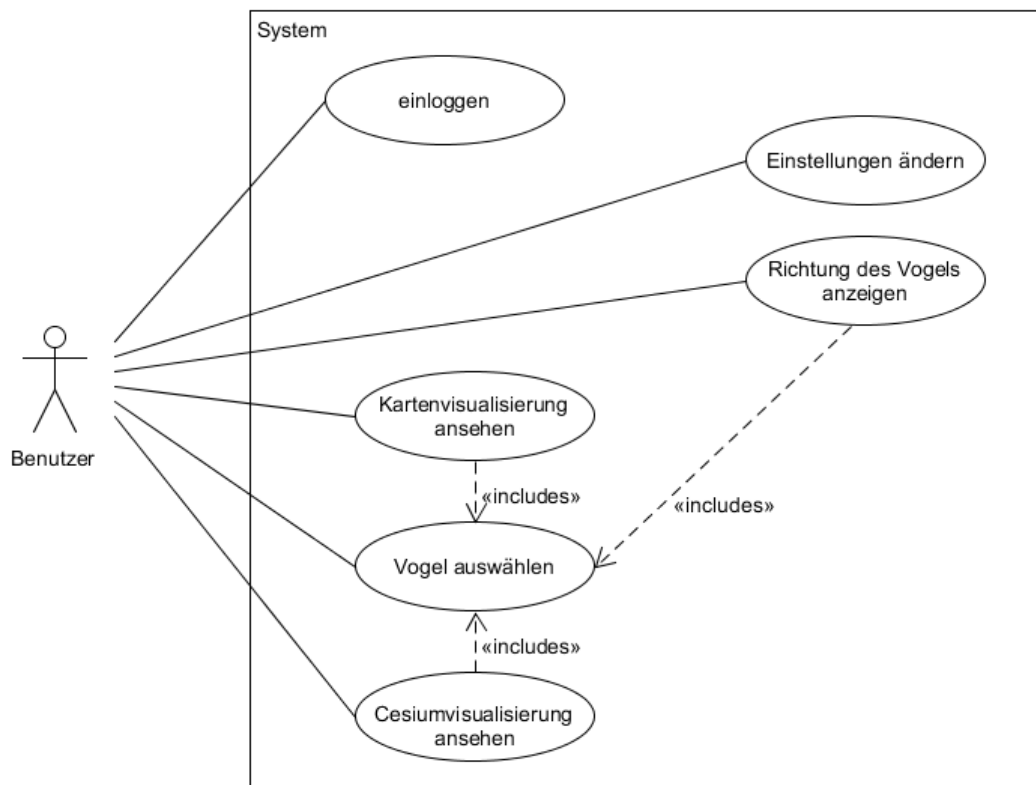
Fertigstellung am 10.7.

- Anforderungen, die mit eingeschränktem Internetzugriff in Verbindung stehen sind realisiert.
- Die momentan angezeigte Vorhersage kann aktualisiert werden (Prüfung, ob neue Daten verfügbar, Neuberechnung; auch unter Verwendung von lediglich mobilem Datennetz)
- Die Visualisierung ist vollständig und auf gute Interpretierbarkeit und Ästhetik optimiert.
- Das User Interface ist vollständig und auf gute Benutzbarkeit und Ästhetik optimiert.

## 7 Diagramme

### 7.1 Anwendungsfälle

In folgendem Diagramm sind die wesentlichen Anwendungsfälle der Applikation und ihre Beziehungen dargestellt. Die einzelnen Anwendungsfälle werden im Folgenden detailliert erläutert und eventuell alternative Abläufe beschreiben.



**Einloggen** Der Nutzer hat die Möglichkeit, sich bei der *Movebank*-API zu authentifizieren.

1. Eingabe der Zugangsdaten
2. Übermittlung der Zugangsdaten bei Anfrage an *Movebank*
  - a. Rückgabe der angeforderten Daten

- b. Benachrichtigung über fehlgeschlagene Authentifizierung

**Einstellungen ändern** Der Nutzer hat die Möglichkeit, Einstellungen der Applikation zu ändern.

**Richtung des Vogels anzeigen** Es wird visualisiert, in welcher Richtung - ausgehend vom Nutzer - der Vogel anzutreffen ist.

**Vogel auswählen** Der Nutzer kann einen Vogel anhand seiner ID auswählen.

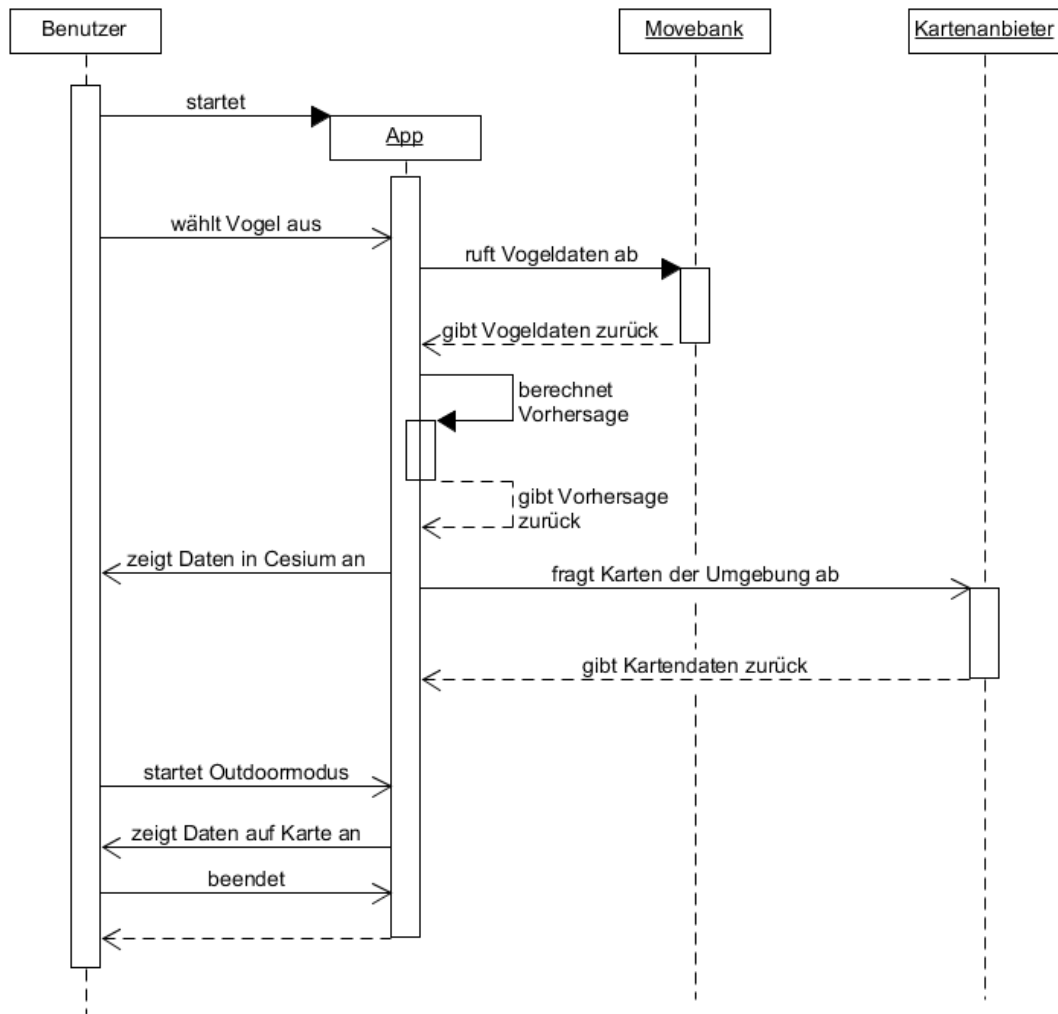
1. Eingabe der ID
2. Abfrage der ID von *Movebank*
  - a. Weiter zur Visualisierung
  - b. Benachrichtigung über fehlerhafte oder nicht gefundene ID

**Kartenvisualisierung ansehen, Cesiumvisualisierung ansehen** Der Nutzer betrachtet die Visualisierung des Ergebnisses der Vorhersage auf einer Kartenlandschaft.

- a. Falls im Normal-Modus: Darstellung in *Cesium*.
- b. Falls im Outdoor-Modus: Darstellung mit *Offlinekarten*.

## 7.2 Abläufe

Im folgenden Diagramm wird exemplarisch ein Informationsaustausch zwischen beteiligten Komponenten und dessen zeitlicher Ablauf in einer vollständigen Sitzung des Nutzers illustriert.



1. Der Nutzer startet die Applikation.
2. Der Nutzer wählt über das User Interface einen Vogel aus.
3. Die Applikation ruft für den spezifizierten Vogel Daten aus *Movebank* ab.
4. Die Applikation berechnet eine Vorhersage.
5. Die Vorhersage wird in *Cesium* visualisiert.
6. Es werden 2D-Kartendaten von einem Kartenanbieter abgefragt und zwischengespeichert.
7. Der Nutzer wechselt in den Outdoor-Modus.



8. Die Vorhersage wird auf einer 2D-Kartenlandschaft visualisiert.
9. Die Applikation wird beendet.