Position Prediction Based on Movement Data

System Design Document

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

Contents

1 Änderungen

Version	Autor	Beschreibung	Datum
1.0	ALL	Erstellen von SRS und SDD	15.5.2018
	BM	Klarstellung Begriff IDs	21.5.2018
	BM	Neu: Abschnitte Erstinstallation, Erstausführung	22.5.2018
	BM	Klarstellung Abschnitt 2.2	22.5.2018
	BM	Erweiterung Abschnitt 4.5	23.5.2018

Legende:
SSJ = Sebastian Strumbelj
BM = Benjamin Moser
OM = Oliver Mänder
SS = Simon Suckut
TJ = Timo Jockers
ALL = SSJ, BM, OM, SS, TJ

2 Einleitung

2.1 Zweck

Das Erstellen einer App, um einem Ornithologen das Finden eines Vogels zu erleichtern, indem sie dessen zukünftige Position mit einem Vorhersagemodell abschätzt.

2.2 Design-Ziele

Diese Dokumentation spezifiert alle Eigenschaften, die unsere App benötigt, um deren Zweck zu erfüllen. Um eine Vorhersage für Vögel zu treffen und dem Ornithologen bei seiner Suche zu unterstützen, sollte die App folgende Funktionen beinhalteten:

- Aufbereiten und Speichern der vorhandenden und für die Berechnung einer Vorhersage benötigten Tracking-Daten von Vögeln

- Berechnung einer Positionsvorhersage
- Visualisierung des Ergebnisses der Positionsvorhersage
- User-Interface zur Interaktion

Der genaue Funktionsumfang ist im SRS definiert.

2.3 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.

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

Offline-Karten ...ist ein 2D-Karten-Provider der Kartendaten liefert, wenn keine Internetverbindung besteht.

StudyID, VogelID Eindeutige Identifikatoren einer Studie bzw. eines einzelnen Vogels in der *Movebank*-Datenbank.

SRS Software Requirements Document

2.4 Referenzen

- SRS
- Treffen mit den Betreuern
- Treffen mit einem Forscher
- Notizen zu Recherche über Nutzungsbedingungen und Lizenzen von Systemkomponenten.

2.5 Überblick

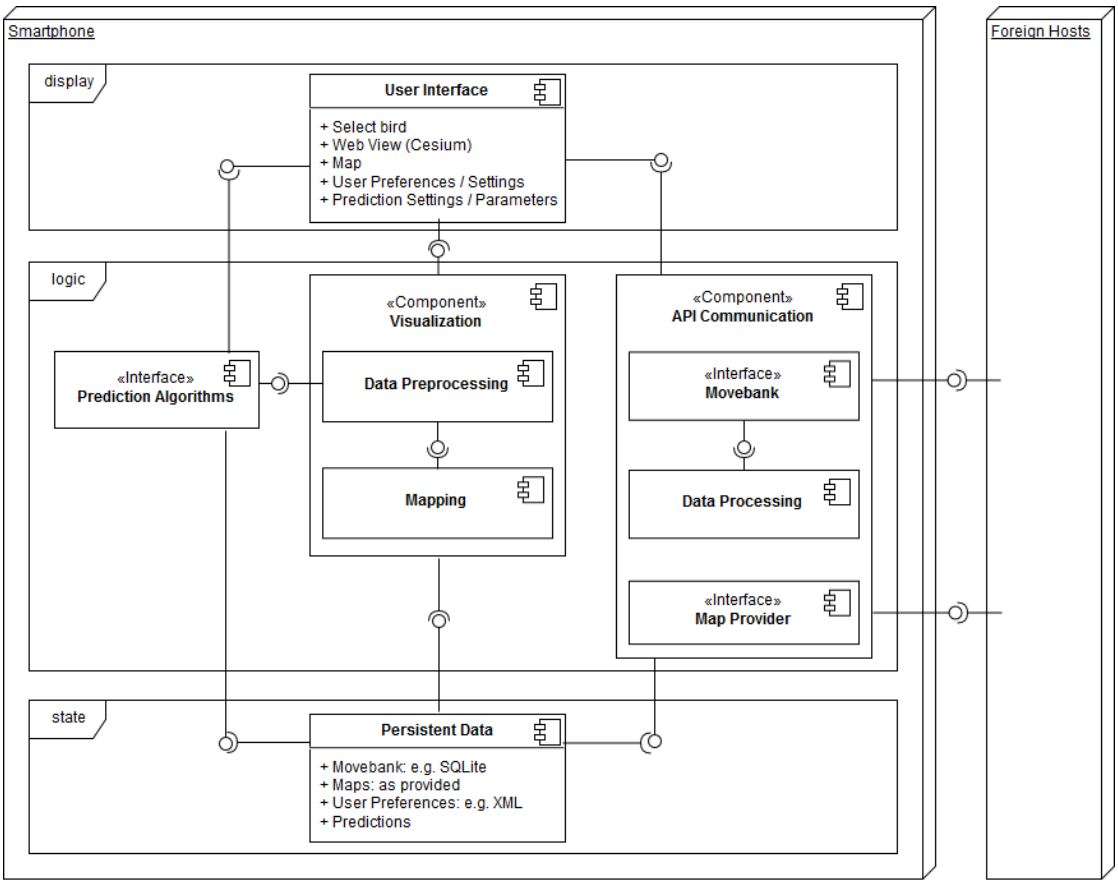
Der erste Teil legt grundlegende Ideen und Definitionen der App fest. Der zweite Teil beschäftigt sich mit bereits existierenden und ähnlichen Systemen. Der dritte Teil liefert eine Beschreibung für die für unsere App genutzte Architektur.

3 Aktuelle Architektur

"Animal Tracker" ist bereits eine App, welche Trackingdaten von Vögeln visualisiert. Allerdings berechnet diese keine Vorhersagen für die Zukunft und fokussiert sich dementsprechened nur auf die Darstellung vergangener Daten. Interne Architekturen dieser App sind allerdings nur gering bekannt.

4 Vorgeschlagene Architektur

4.1 Teilsysteme und Hardware-Software-Zuordnung



Das SRS verlangt ein modulares Softwaresystem. Die Subsysteme sind dazu in einer dreischichtigen Architektur (State, Logic, Display) eingebettet. Die Benutzeroberfläche stellt verschiedene *Activities* zur Verfügung, in der zum einen Nutzereingaben getätigt werden können (Nutzerdaten/-einstellugen, Auswahl des Vogels und Parametereinstellungen für den Vorhersagealgorithmus) und zum anderen die Ergebnisse der Berechnung visualisiert werden (als Kartenansicht und/oder Cesium-WebView).

Abhängig von dem gewählten Vogel bezieht das System Daten von der *Movebank*-API, verarbeitet sie so, dass sie für eine Vorhersage verwendet werden können und speichert sie in der lokalen Datenbank, falls sie dort noch nicht vorhanden sind. Ebenso werden benötigte Kartendaten von einem geeigneten Anbieter abgerufen

und lokal gespeichert. Die gespeicherten Daten werden für die Vorhersage verwendet, deren Ergebnisse wiederum lokal gespeichert werden. Diese werden für die Visualisierung aufbereitet und auf der Karte bzw. in Cesium dargestellt.

4.2 Management von persistenten Daten

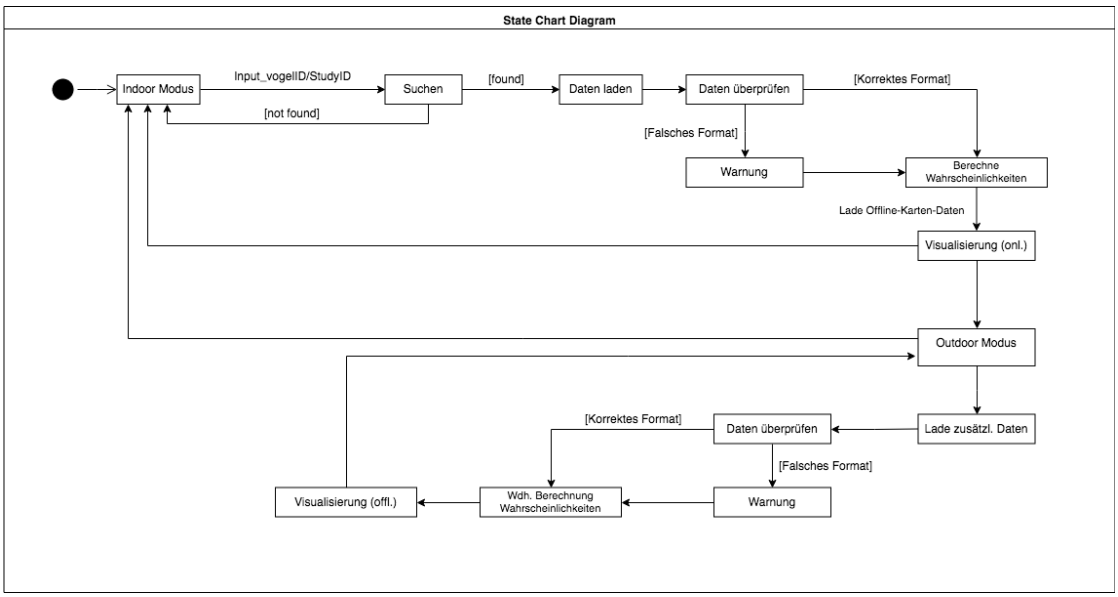
Um Daten nach dem Schließen der Applikation zu behalten, die Menge des Datenvolumens für Anfragen an die Movebank zu reduzieren und um Daten leichter verwalten zu können verfügt die Applikation über eine Schnittstelle zum Parsen und Speichern von XML-Dateien und eine SQLite-Datenbank. Außerdem werden Karten für den Outdoor-Modus lokal gespeichert.

- *SQLite-Datenbank*: In dieser Datenbank werden Daten von der Movebank lokal zwischen gespeichert. Dadurch wird die Menge an Anfragen an die Movebank so gering wie möglich gehalten, die Menge des verbrauchten Datenvolumens reduziert und die Daten werden leichter zu verwalten.
- *XML-Parser*: In der Applikation getätigte Einstellungen werden persistent in einer XML-Datei gespeichert.
- *Offline-Karten*: Karten-Daten werden in dem vom Kartenanbieter verwendeten Format lokal zwischengespeichert.

4.3 Sicherheit

Mit einem Benutzernamen und einem Passwort kann der Zugriff für die Movebank-Daten erweitert werden. Es gibt aber keinen erweiteren Zugriff (z.B. auf die Verwaltung) der App.

4.4 Global Software Control



Das Global-Software-Control-Diagram beschreibt das allgemeine Verhalten der App. Es besteht aus dem allgemeinen Ablauf der App im Indoor- (obere Hälfte) und Outdoormodus (untere Hälfte).

Zu Beginn befinden wir uns im Indoor-Modus und können nun mit der VogelID und StudyID nach einem Vogel suchen. Ist der Vogel nicht verfügbar, so haben wir keine Möglichkeit, weiter zu machen und beginnen von vorne. Ist er vorhanden, so laden wir dessen Daten. Anschließend werden diese geprüft bzw. geparst. Im Fall, dass die Daten formal falsch sind (siehe SRS für die Definition von formal falsch), werden diese trotzdem zur Berechnung benutzt. Allerdings wird eine Warnung über die Qualität der Daten erscheinen. Nach dem Berechnen der Wahrscheinlichkeiten, werden die Kartendaten und weitere Daten für die Visualisierung heruntergeladen, um diese auch ohne Internetverbindung visualisieren zu können. Solange wir uns aber im Indoor-Modus befinden, können diese mit Cesium visualisiert werden. Sollen andere Vögel untersucht werden, so kann die Suche von vorne beginnen.

Geht der Forscher nun in das Feld, schaltet er in den Outdoor-Modus. Der Ablauf ist nun ähnlich wie im Indoor-Modus. Es können zusätzliche oder neue Daten des Vogels über die mobile Datenverbindung geladen werden. Sie fließen in die erneute Wahrscheinlichkeitsberechnung ein und aktualisieren die Visual-

isierung, die nun allerdings mit einem Offline-Karten-Provider dargestellt wird.

4.4.1 Interfaces

- Tracking-Daten werden von der *Movebank*-Datenbank bezogen. Hierfür stellt diese eine Web-API bereit. Es sind verschiedene Endpunkt verfügbar:
  - `/movebank/service/direct-read`
  - `/movebank/service/public`
  - `/movebank/servie/json-auth`

  Es wird von der Applikation ausschließlich der Endpunkt `direct-read` verwendet.
- Daten werden mittels einem Cesium-Interface im Indoor-Modus visualisiert.
- Daten werden mittels einem Open-Street-Map-Interface im Outdoor-Modus visualisiert.

4.5 Boundary Conditions

In diesem Abschnitt werden Randfälle beschrieben, die in der Kommunikation mit Schnittstellen auftreten können sowie desweiteren wie diese behandelt werden.

4.5.1 Anfrage

In Diagramm 1 wird der Ablauf einer Datenanfrage von der *Movebank*-API sowie darauffolgend vom Provider der 2D-Karten dargestellt.

Erläuterungen zu Segment "Server Error" Hier können verschiedene Szenarien auftreten, die aber alle hinsichtlich des Informationsaustausches gleichförmig sind.

- HTTP-Status ist `500 Internal Server Error` – Der Nutzer wird informiert, dass der Server momentan nicht erreichbar ist.

- HTTP-Status ist `401 Unauthorized`: Keine Zugangsdaten in Anfrage vorhanden. In diesem Fall wird dem Nutzer ein Hinweis angezeigt mit der Aufforderung, seine Zugangsdaten in den Einstellungen der App einzutragen.
- HTTP-Status ist `200 OK`
  - Response Body ist `<p>No data are available for download.</p>`. – Informiere Nutzer, dass keine Daten vorhanden sind.
  - Response-Header ist `accept-license: true` – In diesem Fall hat der Nutzer noch nicht die Nutzungsvereinbarung akzeptiert. Dies wird dem Nutzer über einen Hinweis mitgeteilt. Wie im SRS spezifiziert wird hier davon ausgegangen, dass dies durch den Nutzer gewährleistet wird.
  - Response-Body beginnt mit
    `Specify one of the following entity types: deployment, event, ...`
    In diesem Fall konnten die Anfrageparameter nicht von der API interpretiert werden. Zeige Hinweis an den Nutzer, dass die Anfrage nicht möglich war.
- HTTP-Status ist irgendein anderer Fall der hier noch nicht aufgeführt wurde – Dem Nutzer wird ein Hinweis angezeigt, dass die Anfrage nicht erfolgreich durchgeführt werden konnte.

In jedem Fall wechselt die App nach Anzeigen des Hinweises in den Zustand vor der getätigten Anfrage.

4.5.2 Besondere Abläufe bei Einrichtung

Bei Erstinstallation : Bei Erstinstallation der Applikation müssen keine besonderen Prozeduren ausgeführt werden.

Bei Erstausführung : Bei Erstausführung müssen keine besonderen Prozeduren ausgeführt werden. Bei Erstausführung werden noch keine Movebank-Zugangsdaten in den App-Einstellungen eingetragen sein. Bei Anfrage von Daten, die Zugangsdaten erfordern verfährt das System wie im SRS im Abschnitt 7.1 (Anwendungsfälle) spezifiziert.

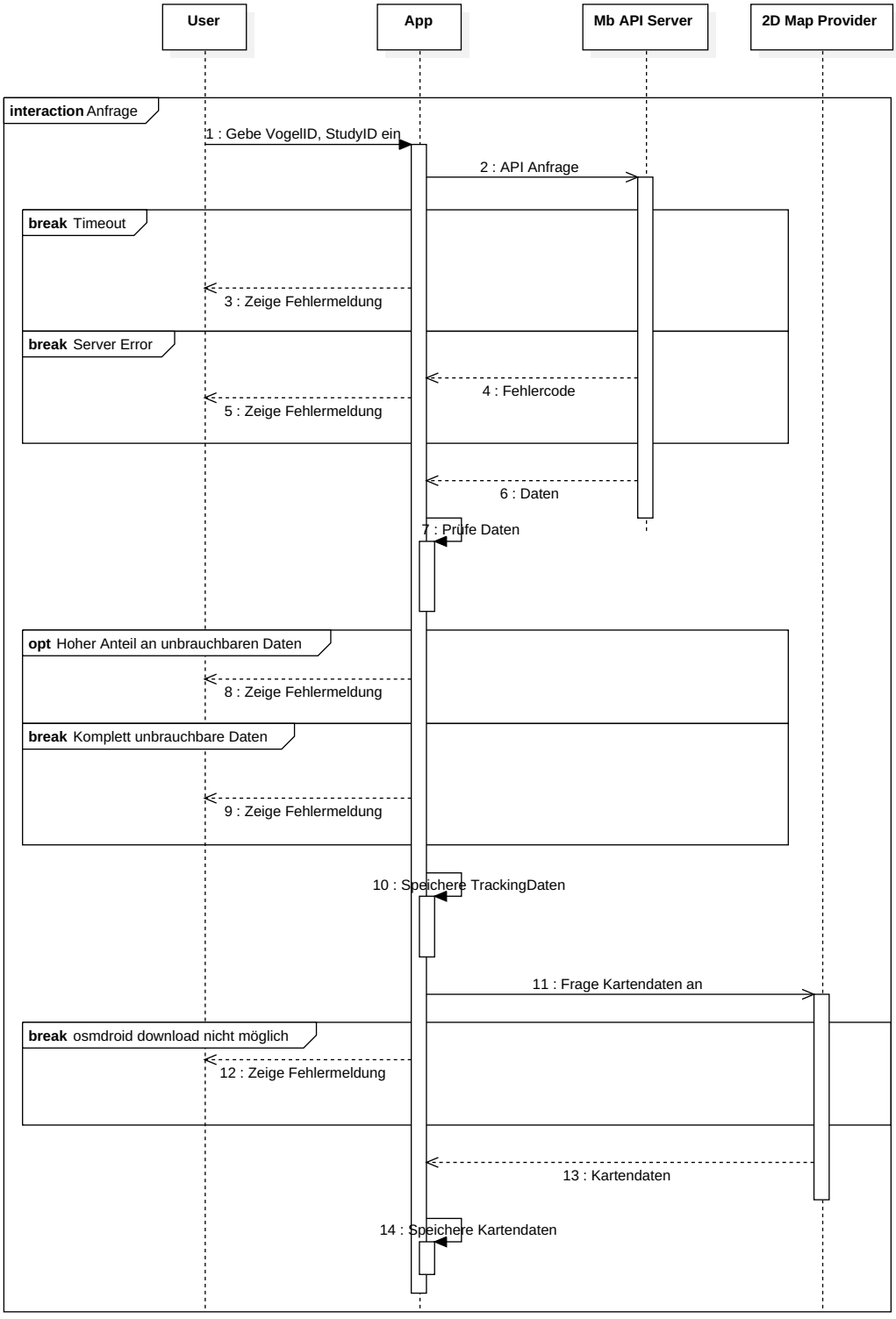


Figure 1: Anfrage