

Crowdbasierte Biodiversitätserfassung – Entwicklung eines Prototypen mit Open-Source-Software

Crowd Based Biodiversity Survey – Development of a Prototype with Open Source Software

Ann-Cathrin Albrecht

Umwelt-Campus Birkenfeld, Hochschule Trier · s17cee@umwelt-campus.de

Zusammenfassung: Um einen Überblick über die am Umwelt-Campus der HS Trier vorkommenden Tier- und Pflanzenarten zu bekommen, wurde im Rahmen des Projektes „Biodivers“ ein Webdienst „Artenfinder“ entwickelt. Dieser ist mittels Open-Source-Software implementiert worden und bietet Studierenden und Beschäftigten die Möglichkeit, gesichtete Arten mit Fundort und Funddatum zu melden und so zum einen die Sensibilität für die heimische Flora und Fauna zu stärken und zum anderen mittelfristig Fragen zu Auswirkungen des Klimawandels auf die Biodiversität beantworten, oder aber Biotop-Managementpläne und Pflegemaßnahmen umweltfachlich fundiert entwickeln zu können. Überdies hat das Projekt zum Ziel, entsprechend den Visionen der nationalen Strategie zur Biologischen Vielfalt (NBS), das gesellschaftliche Bewusstsein für die Natur und den Artenreichtum zu fördern.

Schlüsselwörter: Arterfassung, Biodiversität, Webservice, Citizen Science, OpenLayers

Abstract: *A web service was developed in the project “Biodiversity” to gain an overview about regional animals and plants. It was implemented with open source software and offers the opportunity to report seen species with location and date. This is intended to raise awareness for native flora and fauna among the population and should give information about commonness and occurrence of the different species.*

Keywords: *Species detection, biodiversity, webservice, citizen science, OpenLayers*

1 Motivation

Getreu dem Zitat „Nur was man kennt, kann man auch schützen“ erfordern alle Aktivitäten auf dem Gebiet der Biodiversität, des Artenschutzes oder auch im Biomonitoring die Fähigkeit, Arten sicher bestimmen zu können. Die Dokumentation von Veränderungen – auch bezüglich des Auftretens invasiver Arten und Neubürger – erfordert dabei die Anlage von ökologischen Katastern, da Bestandsentwicklungen nur dann erkannt werden, wenn sie durch Langzeitbeobachtungen von jährlichen Fluktuationen unterscheidbar sind.

Dieser Beitrag beschreibt die Entwicklung eines Prototypen zur Arterfassung im Forschungsprojekt „Biodivers“, welches das Ziel hat, die am Umwelt-Campus der Hochschule Trier vorkommenden Tier- und Pflanzenarten mittels des Citizen-Science-Ansatzes zu dokumentieren. Die Ergebnisse des Projektes sollen zu einem späteren Zeitpunkt dabei helfen, unter anderem Fragen zu Auswirkungen des Klimawandels auf die Biodiversität zu beantworten, oder aber Biotop-Managementpläne und Pflegemaßnahmen am Umwelt-Campus umweltfachlich fundiert entwickeln zu können. Überdies hat das Projekt zum Ziel, entsprechend den Visionen der nationalen Strategie zur Biologischen Vielfalt (NBS), das gesellschaftliche Bewusstsein für die Natur und den Artenreichtum zu fördern (FISCHER-STABEL, 2020).

Um die Arterfassung technisch zu unterstützen werden mittels einer Smartphone App sowie eines Webservices zwei technisch unterschiedliche Wege bestritten, um die Ergebnisse der Freilandkartierungen in die Datenbank zu überführen.

Im Folgenden wird ein mit OpenLayers realisierter Webservice beschrieben, welcher die Möglichkeit bietet, durch Klick auf eine Karte und der Angabe spezifischer Informationen zum Sichtungsobjekt wie Artbezeichnung, Datum, Uhrzeit, beobachtete Tiere und Pflanzen zu melden. Die Besonderheit des Artenfinders liegt darin, dass dieser gänzlich mit frei verfügbaren Programmen entwickelt worden ist und speziell für das Untersuchungsgebiet angelegt wurde, um über eine eigene, heimische Lösung zu verfügen und somit eine Übersicht über die Biodiversität in der Region erlangen zu können.

2 Citizen Science

Für die Datensammlung soll der Citizen-Science-Ansatz genutzt werden. Die Bürgerwissenschaft wird häufig verwendet, wenn Daten in lebenspraktischen Gebieten gesammelt werden sollen. Die Teilnehmenden können ihr Wissen über Natur oder Technik erweitern und bekommen einen greifbaren Einblick in die Wissenschaft und Forschung (FINKE, 2014). Dies steigert die Anerkennung von Forschungsarbeiten, da ein Bewusstsein der Aufwendigkeit der Untersuchungen geschaffen wird. Zudem ist es möglich innerhalb kurzer Zeit viele Daten zu sammeln. Insbesondere im Umweltbereich dient der Citizen-Science-Ansatz einer Verbesserung der Datenbasis durch die Beteiligung von interessierten Laien und Ehrenamtlichen an der Datenerhebung.

Abbildung 1 zeigt den Workflow zur Datenerfassung. Der Kartierer oder die Kartiererin (Studierende oder Beschäftigte) sieht beim Spaziergang ein Reh. Bei Ankunft im Büro bzw. Labor setzt sich dieser an den Computer und trägt die Sichtung am Webinterface ein. Die Meldung wird anschließend in einem Datenbankmanagementsystem gespeichert. Die Daten werden schließlich von einem Forschungsteam gesichtet, hinsichtlich Korrektheit überprüft und ausgewertet.

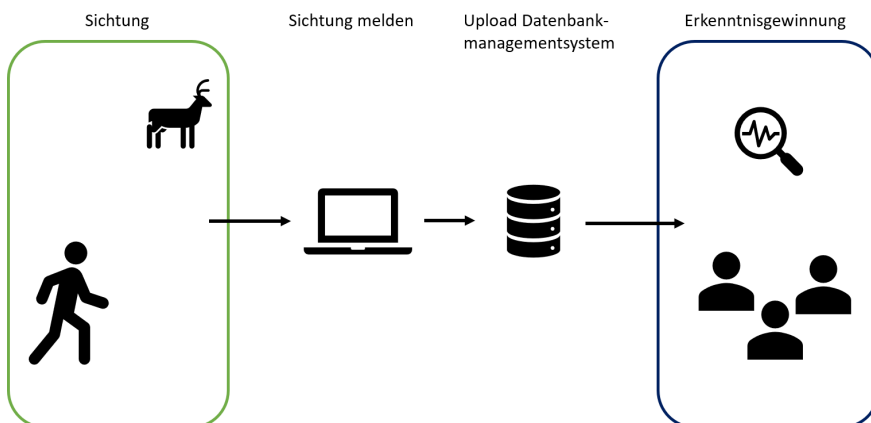


Abb. 1: Schematische Darstellung des Workflows zur Arterfassung

3 Methodik

Nachfolgend sollen die Anforderungen und Grundlagen für das Projekt genannt werden, ehe im folgenden Kapitel die Umsetzung erläutert wird.

3.1 Requirements

Um die Systementwicklung nicht an den Anwendenden vorbei zu betreiben, wurden im Rahmen von Diskussionen mit den Stakeholdern minimale Anforderungen an das System definiert, welche auch im Rahmen eines studentischen Projektes umsetzbar bleiben sollten. Zu den Stakeholdern gehörten die leitenden Professoren des Projektes „Biodivers“, sowie die zuständigen Hochschulangehörigen des Nachhaltigkeitsberichts.

3.2 Funktionale Anforderungen

Funktionale Anforderungen beschreiben, was ein System können soll und sind abhängig von der Zweckbestimmung des Produktes (Robertson, 2012). Für den Artenfinder wurden folgende Anforderungen festgelegt:

- Das Produkt soll eine Plattform zur Biodiversitätserfassung darstellen.
- Der Fundort soll auf der Karte gewählt werden, Angabe erfolgt in Dezimalgradkoordinaten und soll nicht änderbar sein.
- Es soll nur eine eindeutige Artbezeichnung möglich sein.
- Es soll ein Sichtungsdatum eingegeben werden.
- Eine Fotoeingabe soll optional sein.
- Es soll lediglich eine vollständige Meldung an den Server übermittelt werden.
- Im Zentrum der Weboberfläche soll die Karte stehen, welche Zoomen ermöglichen sowie über eine Refresh-Funktion verfügen soll.

3.3 Designanforderungen

Um ein ansprechendes Design zu erreichen (Bühler et al, 2017), sind zudem die folgenden Anforderungen an das Design gestellt worden:

- Die Bedienung soll intuitiv sein.
- Die Weboberfläche soll übersichtlich sein und auf unrelevante Verweise, sowie Animationen verzichten.
- Die Koordinaten sollen lediglich durch Klick auf die Karte gefüllt werden können, um eine Verfälschung durch den Nutzenden zu vermeiden.
- Die Artangabe soll mit deutschem Artnamen möglich sein oder mit wissenschaftlicher Bezeichnung. Diese sollen zur erleichterten Bedienung auswählbar sein. Hierbei muss darauf geachtet werden, dass ein Widerspruch in den beiden Eingaben vermieden wird.

3.4 Technische Anforderungen

3.4.1 Programme

Als Programmieroberfläche dient Visual Studio Code¹. Als Beschreibungssprachen wurde HTML und CSS und als Programmiersprache JavaScript verwendet. Zusätzlich musste

¹ <https://code.visualstudio.com/download>

OpenLayers eingebunden werden, welches kostenfrei Geoverarbeitungswerkzeuge zur Verfügung stellt. Für den Aufbau des Servers wurde der Client PuTTY² und zur Datenübertragung WinSCP³ genutzt.

3.4.2 OpenLayers

Als Grundlage für die verwendete Übersichtskarte des Untersuchungsgebietes dient OpenLayers, eine Werkzeugkiste, welche auf Grundlage von JavaScript Methoden zur Darstellung von Geodaten im Browser zur Verfügung stellt (HAZZARD, 2011). Es bietet die Möglichkeit, eine dynamische Karte in eine Webseite einzubinden und Vektordaten darzustellen. Zudem können durch verschiedene Funktionen z. B. ein Marker auf der Karte platziert werden, der die Lokalisation der nutzenden Person abrufen und anzeigt (OpenLayers, 2006).

3.4.3 Interface

Wie Stone (2005, p. 3) beschreibt, kommen Menschen täglich mit vielen Webseiten in Kontakt. Für die Interaktion ist es stets wichtig, dass die Interfaces leicht bedienbar und verständlich sind, damit ein User die Anwendung gerne nutzt. Daher wurde sich für eine einfache Weboberfläche entschieden, mit einer Karte des Untersuchungsgebietes im Zentrum. Der Nutzende kann den deutschen Artnamen oder die wissenschaftliche Bezeichnung in einer Auswahlliste wählen. Diese Darstellungsweise soll dem Anwendenden die wissenschaftliche Artbezeichnung näherbringen. Durch die Festlegung der auswählbaren Artnamen ist das System zudem vor Meldungen sicher, bei denen als Artnamen willkürlich aneinandergereihte Buchstabenketten angegeben sind. Da es sich bei dem Webservice aktuell um einen Prototypen handelt, stehen zum gegenwärtigen Zeitpunkt nur die Arten der Gruppen der Amphibien und Reptilien zur Auswahl bereit. Auf die zwingende Eingabe eines Fotos der gesichteten Art wurde verzichtet. Hierdurch eröffnet sich die Möglichkeit, im Nachgang gesichtete Arten zu melden, auch wenn keine Aufnahmen in der Natur gemacht wurden. Auch ist es schwierig Fotos von z. B. Vögeln zu machen, weshalb eine Meldung dieser durch die Fotopflicht sehr erschwert wäre. Dies birgt aber die Gefahr, dass fehlerhafte Meldungen erfolgen und keine Plausibilitätsprüfung möglich ist. Um bewusst falsche Meldungen auf ein Minimum zu reduzieren, wurde daher der Artenfinder nur für Hochschulangehörige, Studierende und Beschäftigte freigeschaltet, indem eine VPN-Verbindung zum Hochschulnetz aktiv sein muss, um die Webanwendung aufrufen zu können.

3.4.4 Serverseite

Die Datenablage soll in einer postgres-Datenbank erfolgen, welche auf einem virtuellen Server der Hochschule liegt. Hierfür wurden die freien Tools PuTTY und WinSCP genutzt. Der Client PuTTY nimmt über eine textbasierte Terminaloberfläche Befehle entgegen und führt diese auf dem zuvor angegebenen Server aus (PuTTY, 1998). WinSCP baut eine Verbindung zu einem Server auf und kann Daten in einem geschützten Transfertunnel übertragen (WinSCP, 2000).

² <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

³ <https://winscp.net/eng/download.php>

4 Umsetzung

Der Umsetzung liegen neben der gängigen HTML- und CSS-Implementierung Besonderheiten aufgrund der Karteneinbindung und Karteninteraktion durch den Nutzenden zugrunde, welche im Folgenden skizziert werden sollen.

4.1 Frontend

Zunächst muss für die Verwendung der Karten im HTML-Header die OpenLayers-Bibliothek angegeben werden. Anschließend folgt die Initialisierungsfunktion welche mit *new OpenLayers.Map* eine Karte aufbaut. Über die Funktionen *.setCenter* wird der Mittelpunkt der Karte festgelegt. Mit *.addControl(new OpenLayers.Control.MousePosition());* wird die Mercator-Koordinate des Pixels im unteren rechten Eck angezeigt, über welchem sich der Mauszeiger gerade befindet. Um die Karte des Untersuchungsgebietes beim Aufruf der Seite zu laden, ist der Aufruf dieser im Beginn des HTML-Body notwendig:

```
<body onload="init();" > .
```

Mit Klick auf die Karte werden die auf der Mausposition liegenden Koordinaten in das Formular eingetragen, wofür ein Click-Event in Kombination mit OpenLayers notwendig ist. Hierfür wird dem Mausklick zunächst eine Variable zugewiesen, diese dann der Methode *.addControl* übergeben und die zum Trigger gehörende Funktion ausgeführt. Die Implementierung lautet wie folgt:

```
var click = new OpenLayers.Control.Click();
map.addControl(click);
click.activate();
```

Da OpenLayers Mercator-Koordinaten verwendet, in diesem Projekt aber das Dezimalgrad-System genutzt wird, muss im Hintergrund eine Umrechnung der beiden Koordinatensysteme erfolgen. Hervorzuheben ist bei dieser Funktion die von OpenLayers bereitgestellte Möglichkeit zur Erkennung der Koordinate hinter dem Pixel:

```
var lonlat = map.getLonLatFromPixel(e.xy);.
```

Der Zugriff auf die Längen- bzw. Breitengradangabe erfolgt synchron zum Aufruf von Klassenattributen in der Objektorientierten Programmierung mit beispielsweise *lonlat.lon*. Die Koordinaten werden mit diesen Formeln umgerechnet:

$$LAT_{Dezimal} = \frac{2 * \left(\arctan \left(e^{\left(\frac{LAT_{Mercator}}{6.371.000} \right)} \right) - \frac{\pi}{4} \right)}{\pi} * 180$$

$$LON_{Dezimal} = \frac{LON_{Mercator} * 180}{6.371.000 * \pi}$$

Anschließend wählt der Nutzende den deutschen Artnamen oder die wissenschaftliche Bezeichnung in einer Auswahlliste aus. Das jeweilige Gegenstück wird ebenfalls selektiert, um fehlerhafte „Mischmeldungen“ zu vermeiden. Um dies zu implementieren muss eine JavaScript-Anweisung eingebunden werden. Diese sieht für das Auswahlfeld des deutschen Artnamens wie folgt aus:

```

let name = document.getElementById("name");
name.addEventListener("change", function () {
    document.getElementById("nameWissenschaftlich").value = name.value;
});

```

Die fertige Weboberfläche ist in Abbildung 2 dargestellt. Wie in der Methodik beschrieben, nimmt die Karte des Untersuchungsgebietes den größten Teil der Webseite ein. Rechts daneben befindet sich das Formularfeld, in welchem der Anwendende die Informationen zur gesichteten Art einträgt. Mit Klick auf den Senden-Button werden die Daten übermittelt.

The screenshot shows the 'Artenfinder Umwelt-Campus Birkenfeld' web application. The top navigation bar is green with links for 'Artenfinder', 'Projekt', 'Kartiertermine pdf', and 'AppStore'. The main area is split into two columns. The left column contains a map of the 'Umwelt-Campus Birkenfeld' with various buildings and green spaces labeled. The right column contains a form titled 'Ich habe eine Art entdeckt!'. The form has the following fields: 'Fundort' with a text input showing coordinates; 'Artnamen' with a dropdown menu showing 'Feuersalamander'; 'wissenschaftliche Bezeichnung' with a dropdown menu showing 'Salamandra salamandra'; 'Foto einfügen' with a button 'Datei auswählen' and a note 'Keine ausgewählt'; 'Sichtungszeit' with a date input showing '01.06.2021'; and 'Kommentar' with a text area. A 'Senden!' button is at the bottom of the form. Below the form is a small photo of two owls perched on a tree branch. The bottom right corner of the page has the text '© Umwelt-Campus Birkenfeld, 2020'.

Abb. 2: Grafische Oberfläche Webservice

4.2 Backend

Nach der PuTTY-Einrichtung müssen folgende Pakete installiert werden:

- apache2 für den Webserver. Dieser läuft standardmäßig auf Port 80.
- php7.0 als Skriptsprache bildet das Bindeglied zwischen Webserver und Datenbank, indem es den Code des Webserver interpretiert und ausführt (Pollakowski, 2003).
- mysql-server stellt die Grundlage für das spätere relationale Datenbanksystem.
- tomcat9, welches automatisch openjdk-11-jre installiert, wird benötigt, damit später OpenLayers genutzt werden kann, da dieses auf einer JavaScript Bibliothek basiert.
- postgresql und postgis, um Geodaten in einer Datenbank abzulegen (Strobl, 2008).

Um den Webservice online zu stellen, wird das HTML-File mithilfe von WinSCP in den Ordner `www > html` abgelegt. Der Artenfinder ist unter `localhost/artenfinder.html` aufrufbar.

Anschließend wird in der PostgreSQL-Oberfläche eine neue Datenbank angelegt, in welcher im weiteren Verlauf die gemeldeten Sichtungen abgelegt werden können, wie Abbildung 3 zeigt.

```

artenfinder=# SELECT * FROM meldung;
          artname | wissname | lon | lat | sichtszeit | kommentar
-----+-----+-----+-----+-----+-----
Blindschleiche | Anguisfragilis | 7.1765 | 49.6009 | 2021-01-20 | 0
(1 row)
artenfinder=#

```

Abb. 3:
Datenbank Art-
erfassung in
PuTTY

Dieser Ansatz konnte aber nicht weiterverfolgt werden, da das Rechenzentrum Sicherheitsbedenken äußerte.

Somit wurde eine andere Möglichkeit zur Datenablage benötigt. Da die Studierenden und Beschäftigten von rheinland-pfälzischen Hochschulen einen kostenfreien Zugriff auf Seafile haben und diese Plattform datenschutzkonform ist, wurde sich für diese vorübergehende Alternative entschieden. Durch das Cloudsystem können Daten mit anderen Personen geteilt werden und liegen auf den eingeloggtten Endgeräten synchronisiert vor (Seafile, 2021), was elementar für den Artenfinder und die Datenauswertung ist. Somit ergibt sich der in Abbildung 4 dargestellte Workflow:

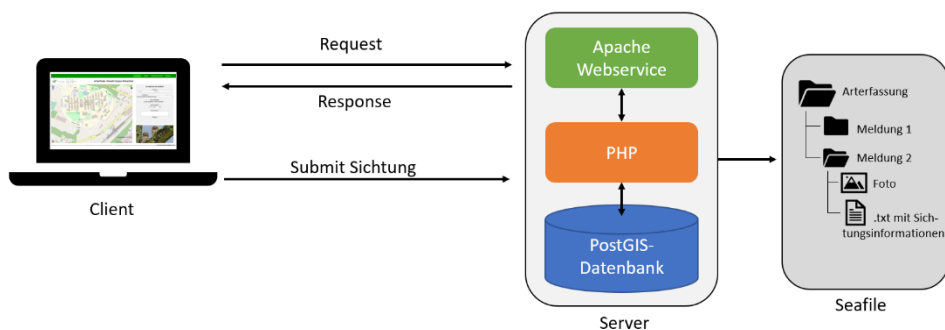


Abb. 4: Technischer Workflow

Der Client sendet eine REQUEST-Anfrage an den Webserver des Artenfinders, woraufhin dieser mit der entsprechenden RESPONSE reagiert und der Artenfinder auf dem Client geladen wird. Bei einem SUBMIT der Sichtung wird erneut eine Nachricht an den Server gesendet. Diese wird vom PHP-Interpreter verarbeitet und die entsprechenden Aktionen angestoßen. In diesem Fall werden die durch die meldende Person eingegebenen Daten zur gesichteten Art in eine einfache Textdatei geschrieben und gemeinsam mit dem optionalen Foto in einem neuen Unterordner im Arterfassung-Verzeichnis in Seafile abgelegt. Die Ordnerstruktur ist der Abbildung 5 zu entnehmen. Die Ablage der App-Anwendung erfolgt identisch, um eine konsistente Datenablage zu haben und eine spätere Auswertung zu erleichtern.

[Für mich freigegeben](#) / [Artenfinder](#) / [UCB](#) / 8

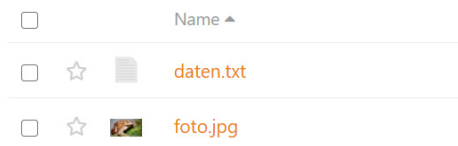


Abb. 5:
Datenablage in Seafile

5 Diskussion und Ausblick

Durch den Artenfinder wurde ein Tool geschaffen, welches die webbasierte Erfassung von Artsichtungen im Rahmen eines Crowdsourcing-Ansatzes erlaubt. Zusätzlich existiert eine App für Android, welche das Melden von gesichteten Arten direkt in der Natur erlaubt. Da diese von einer anderen Projektgruppe implementiert wurde, verzichtet dieser Beitrag auf eine tiefere Erläuterung.

Da es sich bei dem Artenfinder aktuell noch um einen Prototypen handelt, welcher bisher lediglich Amphibien und Reptilien abdeckt, sollen die Erfassungsmasken nach einer Testphase im Frühjahr 2021 um Aquatische Invertebraten, Schmetterlinge, Blütenpflanzen und Vögel ergänzt werden. Auch ist eine zusätzliche Funktion der Einblendung einer Heatmap geplant, damit der Nutzende visualisieren kann, an welcher Stelle z. B. die meisten Honigbienen gesichtet wurden und welche Bereiche bisher wenige Meldungen aufweisen. In jedem Fall soll die Datenablage wie beschrieben optimiert werden, um die Kartier-Ergebnisse mittelfristig in einer Datenbank bereitzustellen.

Literatur

- Apache (1995). <https://www.apache.org/>, Retrieved Feb 2, 2021.
- Bonn, A. et al. (2017). *Grünbuch Citizen Science Strategie 2020 für Deutschland*.
- Bühler, P., Schlaich, P., & Sinner, D. (2017). Interfacedesign. In *Webdesign* (pp. 11–44). Berlin/Heidelberg: Springer Vieweg.
- Finke, P. (2014). *Citizen science. Das unterschätzte Wissen der Laien*. München: oekom.
- Fischer-Stabel, P. (2020). *Dokumentations- und Managementsystem für Biodiversität*. Retrieved Jan 24, 2021. <https://www.umwelt-campus.de/iss/projekte/laufende-projekte/biodivers>.
- Freeman, R. E. (2010). *Strategic management: A stakeholder approach*. Cambridge Univ. Press.
- Hazzard, E. (2011). *Openlayers 2.10 beginner's guide*. Packt Publishing Ltd.
- OpenLayers (2006). <https://openlayers.org/>, Retrieved Jan 24, 2021.
- Robertson, S., & Robertson, J. (2012). *Mastering the requirements process: Getting requirements right*. Addison-Wesley.
- Pollakowski, M. (2003). Datenbankzugriff über ein Web-Interface mit PHP. In *Grundkurs MySQL und PHP* (pp. 87–102). Wiesbaden: Vieweg+ Teubner.
- PuTTY (1998). <https://www.putty.org/>, Retrieved Feb 2, 2021.
- Seafire (2021). <https://www.seafile.com/en/home/>, Retrieved Feb 2, 2021.
- Stone, D. et al. (2005). *User interface design and evaluation*. Elsevier.
- Strobl, C. (2008). PostGIS. https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-35973-1_1012. Retrieved Feb 1, 2021.
- WinSCP (2000). <https://winscp.net/eng/docs/lang:de>. Retrieved Feb 1, 2021.