

**Duale Hochschule Baden-Württemberg**

**Stuttgart Campus Horb**



## **Geogram - Dokumentation**

***T3INF4310 - Entwicklung mobiler Applikationen***

---

eingereicht von:	Benita Dietrich, Paul Finkbeiner, Josua Stricker, Jonas Schwarz, Sven Stoll und Moris Kotsch
Modul:	T3INF4310 - Entwicklung mobiler Applikationen
Dozent:	B.Sc. Torsten Hopf
Kurs:	TINF2018
Studiengang:	Informatik
Hochschule:	DHBW Stuttgart Campus Horb
Bearbeitungszeitraum:	08.03.2021 - 06.06.2021

Horb am Neckar, 23. Mai 2021

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
1.1	Aufgabenstellung . . . . .	1
1.2	Projektidee . . . . .	1
1.3	Projektteilnehmer . . . . .	2
<b>2</b>	<b>Konzeption</b>	<b>3</b>
2.1	Mockup's . . . . .	3
2.2	MVP-Kriterien . . . . .	5
2.3	Soll- und Kann-Kriterien . . . . .	6
<b>3</b>	<b>Architektur</b>	<b>7</b>
3.1	Frontend . . . . .	7
3.2	Backend . . . . .	9
3.2.1	Credentialmanager . . . . .	9
3.2.2	Cloud Firestore . . . . .	9
3.2.3	Foto-Server . . . . .	12
<b>4</b>	<b>User-Guide</b>	<b>13</b>
4.1	Login-/Logout-/Nutzerverwaltung . . . . .	13
4.2	Profil - Ansicht . . . . .	15
4.3	Suchen - Ansicht . . . . .	16
4.4	Explore - Ansicht . . . . .	17
<b>5</b>	<b>Veröffentlichung im Play Store</b>	<b>18</b>

# Abbildungsverzeichnis

2.1	Login-Ansicht in weiß . . . . .	3
2.2	Login-Ansicht in schwarz . . . . .	3
2.3	Feed Overview . . . . .	4
2.4	Feed Detail . . . . .	4
2.5	Eigene Profil-Übersicht . . . . .	4
2.6	Eigenes Profil bearbeiten . . . . .	4
2.7	PopUp-Fenster für neuen Feed . . . . .	5
2.8	Neuen Feed bearbeiten . . . . .	5
3.1	Technologieübersicht . . . . .	7
3.2	Ionic Tags . . . . .	8
3.3	Speicherstruktur Firestore . . . . .	10
3.4	Felder eines „users“-Dokumentes . . . . .	10
3.5	Felder eines „images“-Dokumentes . . . . .	11
4.1	Login - Ansicht . . . . .	13
4.2	Registrierungs - Ansicht . . . . .	14
4.3	E-Mail-Vergessen - Ansicht . . . . .	14
4.4	Profil - Ansicht . . . . .	15
4.5	Post-PopUp . . . . .	15
4.6	Post Informationen . . . . .	15
4.7	Edit Profil - Ansicht . . . . .	16
4.8	Suchen - Ansicht . . . . .	16
4.9	Explore - Ansicht . . . . .	17
5.1	Übersicht der Logos . . . . .	18
5.2	Die App im Play Store . . . . .	19

# Programm-Listings

3.1	Implementierung der GPS-Abfrage . . . . .	8
3.2	Codeausschnitt aus dem Foto-Server . . . . .	12

# 1 **Kapitel 1**

---

## **Einführung**

Nachfolgend wird eine Einleitung in die hier vorliegende Projektdokumentation gegeben. Neben einer vorgegebenen Aufgabenstellung und selbstdefinierter Kernidee, wird eine Auflistung aller Projektteilnehmer aufgezeigt.

### **1.1 Aufgabenstellung**

Ziele dieses Projektes ist die Konzeption und Implementierung einer mobilen App. Jenes Projekt findet im Rahmen der Vorlesung „Entwicklung mobiler Applikationen“ statt. Als Rahmenbedingungen wurden lediglich folgende Punkte vorausgesetzt:

- Der Use Case sollte kein Spiel sein, sondern eher eine klassische App abbilden.
- Der Use Case soll mindestens von zwei der zur Verfügung stehenden Funktionen eines Mobilgeräts nutzen. Beispielsweise
  - GPS-Sensor
  - Neigungssensor
  - Kamera
  - Fingerabdruck
  - NFC
- Eine FSK Einstufung der Applikation sollte nicht nötig sein.

### **1.2 Projektidee**

Als Gruppe wurde eine Kernidee für die Erfüllung der Projektanforderungen entworfen. Die mobile Anwendung orientiert sich an der bestehenden mobilen Anwendung „Instagram“.

Ziel ist eine Plattform für das Teilen von Bildern und Entdecken von Content, welche sich in der Nähe des Benutzers befinden. Vergleichbar zu Instagram sollen relevante Beiträge in Feeds dargestellt werden. Der große Unterschied zu Instagram ist der Fokus auf die geografischen Hintergrundinformationen der Beiträge. So werden den Benutzern

lediglich die Beiträge von anderen Benutzern angezeigt, welche sich in einer festgelegten geografischen Entfernung befinden. Durch die Einbindung und Verwendung von geografischen Informationen, soll die Vermarktung von lokalen Angeboten, Aktivitäten und Sehenswürdigkeiten erleichtert werden.

Durch die große Ähnlichkeit mit Instagram, und der Erweiterung um die Nutzung von GPS-Informationen, für die Darstellung von ortsnahe Feed-Beiträgen, wurde sich einheitlich für den Projektnamen „Geogram“ entschieden.

Wie in der Aufgabenstellung (Abschnitt 1.1) gefordert, beinhaltet die mobile Anwendung „Geogram“ zwei Funktionen von Mobilgeräten. Verwendet wird unter anderem die **Kamera** und die **GPS-Sensorik** von heutigen Mobilgeräten.

## 1.3 Projektteilnehmer

Das Projekt wird von sechs Student\*innen bearbeitet. Zusätzlich wird das Projekt während der kompletten Laufzeit von einem Stakeholder (Dozent des Moduls) betreut.

Rolle	Name	Kontakt
Mitglied	Benita Dietrich	i18008@hb.dhbw-stuttgart.de
Mitglied	Paul Finkbeiner	i18011@hb.dhbw-stuttgart.de
Mitglied	Josua Stricker	i18039@hb.dhbw-stuttgart.de
Mitglied	Jonas Schwarz	i18037@hb.dhbw-stuttgart.de
Mitglied	Sven Stoll	i18038@hb.dhbw-stuttgart.de
Mitglied	Moris Kotsch	i18021@hb.dhbw-stuttgart.de
Stakeholder	Torsten Hopf	torsten.hopf@mhp.com

# Kapitel 2

## Konzeption

Während der Konzeptionsphase ging es darum, einen ersten Entwurf für die spätere Implementierung zu entwerfen. Hierfür wurden verschiedene Mockups erstellt, welche einen ersten Einblick in die Designvorstellungen für die mobile Anwendung „Geogram“ geben sollen.

Mit dem Definieren der MVP-Kriterien, wurde der Funktionsumfang, für die erste minimal funktionsfähige Iteration der mobile Anwendung „Geogram“ aufgelistet. Zusätzlich wurden noch weitere Soll- und Kann-Kriterien festgelegt.

### 2.1 Mockup's

In Abbildung 2.1 und Abbildung 2.2 ist die Login-Ansicht abgebildet. Bereits als Mockup realisiert, ist der Wechsel zwischen einem White- und Dark-Mode der mobilen Anwendung.



Abb. 2.1: Login-Ansicht in weiß

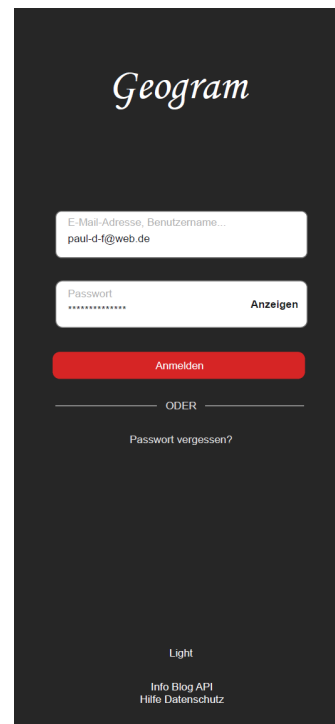


Abb. 2.2: Login-Ansicht in schwarz

In Abbildung 2.3 und Abbildung 2.4 sind die Feed-Ansichten abgebildet. Die Feed-Übersicht ähnelt der Feed-Übersicht von Instagram, jedoch mit dem Unterschied, dass mithilfe einer Standortangabe die Filterung der Feeds vollzogen wird.

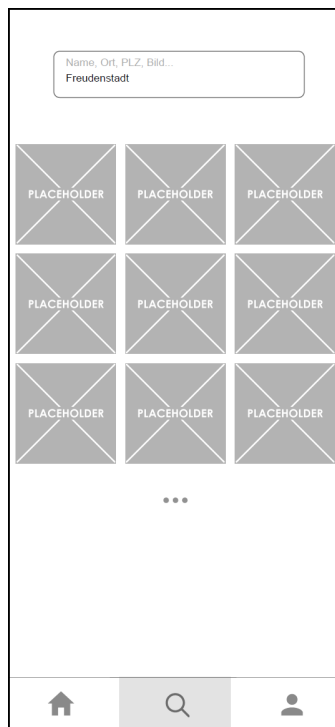


Abb. 2.3: Feed Overview

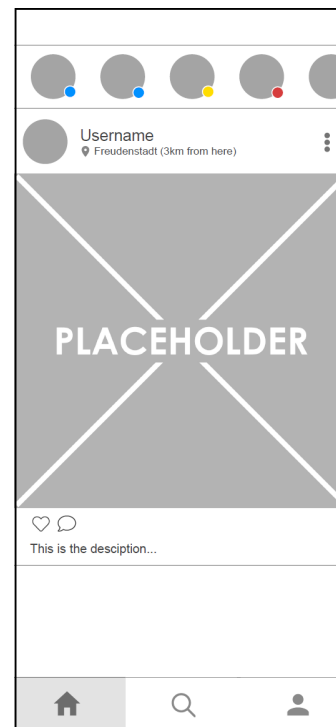


Abb. 2.4: Feed Detail

Die folgenden zwei Abbildungen zeigen die Profilansicht mit Änderungsmöglichkeit.

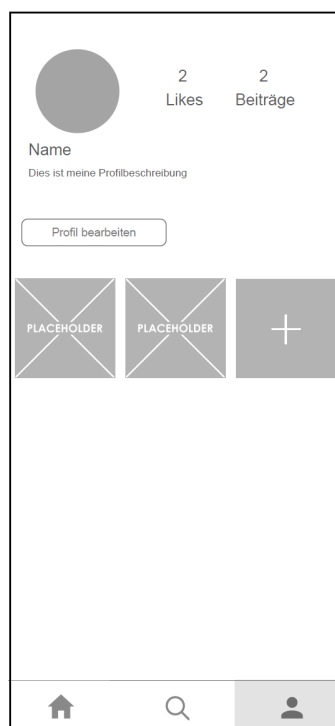


Abb. 2.5: Eigene Profil-Übersicht



Abb. 2.6: Eigenes Profil bearbeiten



Der Vorgang beim Hochladen eines neuen Feeds wird in Abbildung 2.7 und Abbildung 2.8 visualisiert. Der Benutzer hat die Möglichkeit das Foto entweder aus der eigenen Galerie auszuwählen oder mit der Kamera aufzunehmen. Noch während dem Hochladevorgang kann der Benutzer dem Feed eine Beschreibung und die GPS-Informationen für die Standortangabe hinzufügen.



Abb. 2.7: PopUp-Fenster für neuen Feed

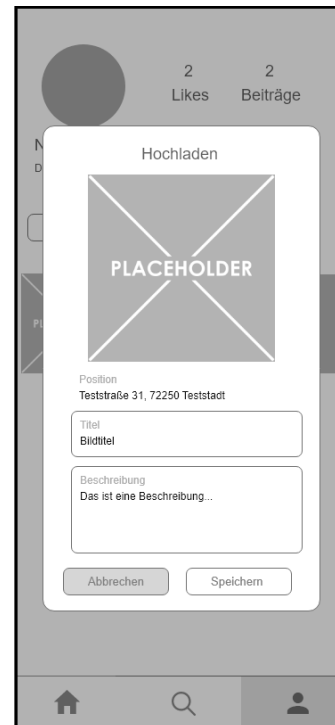


Abb. 2.8: Neuen Feed bearbeiten

## 2.2 MVP-Kriterien

Folgende Kriterien müssen für eine erste, minimal funktionsfähige Iteration der mobilen Anwendung „Geogram“, während der Implementierung umgesetzt werden:

- Login/Nutzerverwaltung
- Upload-Funktion für die Bilder mit Standortdaten
- Zugriff auf die Kamerafunktion des Mobilgerätes
- Zugriff auf die GPS-Daten (GPS-Modul des Mobilgerätes)
- Datenbank für Benutzerkonten, Standortdaten und Bildern
- Anzeigen der Beiträge

## 2.3 Soll- und Kann-Kriterien

Neben den relevanten MVP-Kriterien wird die Konzeption noch mit Soll- und Kann-Kriterien ergänzt.

### **Soll-Kriterien:**

- Löschen von Bildern bzw. Beiträgen
- Hinzufügen von Bildbeschreibungen
- Ansicht des eigenen Kontos
- Einbindung von Google Maps für die Standortdaten

### **Kann-Kriterien:**

- Anmelden per Fingerabdruckssensor
- Anderen Benutzer „folgen“ („folgen“-Funktion)
- Benutzerdefinierte Filterfunktion für Beiträge
- Kommentare, Likes und Hastags für die Beiträge
- Dark-Mode abhängig von Systemeinstellungen

# Kapitel 3

## 3 Architektur

Für die Architektur der mobilen Anwendung „Geogram“, wurden die zwei Frameworks „Ionic“ und „Express“ und die Entwicklungs-Plattform „Firebase“ verwendet.

Wie in Abbildung 3.1 veranschaulicht, wird das Web-Framework „Ionic“ für das Frontend verwendet. Das Backend kann in zwei Bereiche unterteilt werden. Zum einen wird die Entwicklungs-Plattform „Firebase“ und das Node.js Web-Framework „Express“ verwendet. Genauere Informationen bezüglich der Aufteilung des Backends wird in Abschnitt 3.2 beschrieben.

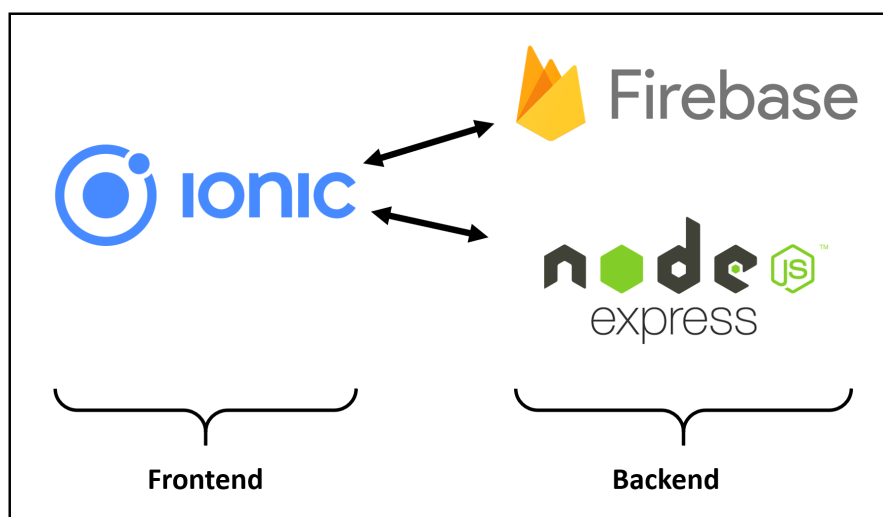


Abb. 3.1: Technologieübersicht

### 3.1 Frontend

Für die Umsetzung des Frontend wurde das Framework „Ionic“ verwendet.

Ein Vorteil von Ionic ist es, dass der Code nicht für jede Plattform einzeln geschrieben werden muss. Als Entwickler ist man dadurch nicht verpflichtet nativen Code zu implementieren.

Jene Plattform unabhängige Entwicklung ist durch den Einsatz von „Ionic Tags“ möglich. Diese speziellen HTML-Elemente gewährleisten, dass das „Look and Feel“ der App auf allen Plattformen gegeben ist.

In Abbildung 3.2 sind die in der App „Geogram“ verwendeten Ionic Tags abgebildet.

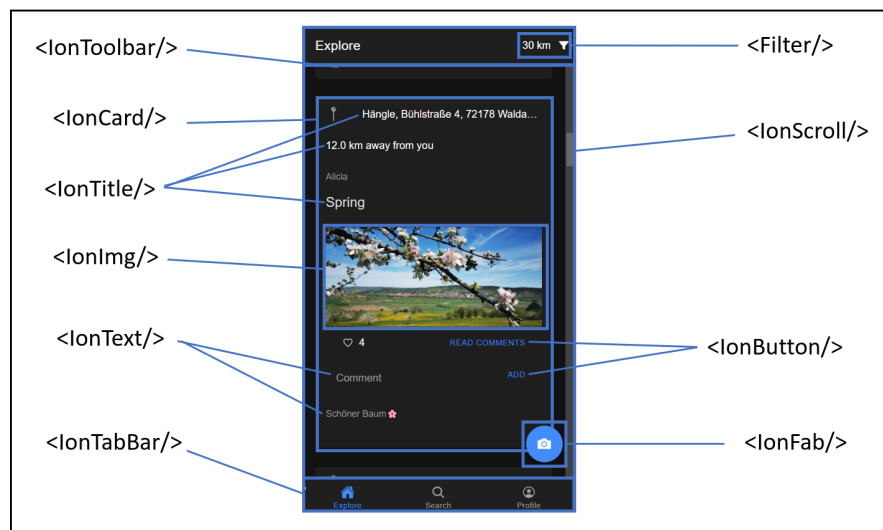


Abb. 3.2: Ionic Tags

Damit die Bilder beim Hochladen auch geografische Informationen besitzen, wird mittels Ionic auf die GPS-Sensoren der Geräte zugegriffen. In Listing 3.1 ist genau jene Abfrage zu finden. Um die erhaltenen Längen- und Breitengrade in bekannte Straßen und Städte umzuwandeln, wird in Zeile 19 auf eine externe Dienstleistung zurückgegriffen.

```

1  ...
2  // get current geolocation
3  Geolocation.getCurrentPosition().then((location) => {
4      setLocation({
5          coords: {
6              accuracy: location.coords.accuracy,
7              altitude: location.coords.altitude,
8              altitudeAccuracy: location.coords.altitudeAccuracy,
9              heading: location.coords.heading,
10             latitude: location.coords.latitude,
11             longitude: location.coords.longitude,
12             speed: location.coords.speed,
13         },
14         timestamp: location.timestamp,
15     });
16     // reverse geolocation
17     axios
18     .get(
19         `https://api.opencagedata.com/geocode/v1/json?q=${location?.coords.
20         latitude}+${location?.coords.longitude}&key=bb44e70b618844faba7c
21         b23b580419e0`
22     )
23     .then((res) => {
24         setPosition(res.data.results[0].formatted);
25         setLocation((state: any) => {

```

```
26     return {
27         ...state,
28         position: res.data.results[0].formatted,
29     };
30     });
31 });
32 });
33 ...
```

---

Listing 3.1: Implementierung der GPS-Abfrage

## 3.2 Backend

Den Großteil der Backend-Aufgaben übernimmt die Entwicklungsplattform Firebase. Verwendet wird hierfür die kostenlose „Spark Plan“-Version von Firebase.

Die **Authentifizierung** und das **Credentialmanagent** wird hierbei vollständig von Firebase übernommen und verwaltet. Über vordefinierte Programmierschnittstellen kann man anschließend auf die Funktionen von Firebase zugreifen. Ebenso wird im kostenlosen Lizenzmodell „Spark Plan“ eine Cloud-Speicherung namens „**Firestore**“ angeboten. In der darin enthaltenen **Echtzeitdatenbank** stehen jedoch nur 1 GiB Speicherplatz zur Verfügung. Da sich das Kernkonzept von Geogram hauptsächlich um Fotos dreht und die finanziellen Mittel der Gruppe nicht für ein kostenpflichtiges Lizenzmodell ausreichen, wurde zusätzlich ein **Foto-Server** mithilfe des Web-Frameworks „Express“ implementiert. Der Foto-Server wird von der Projekt-Gruppe selbst gehostet und beinhaltet mehr als 1 Gib Speicherplatz.

### 3.2.1 Credentialmanagent

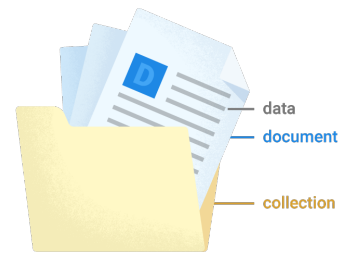
Die Verwaltung von Anmeldeinformationen wird in Geogram von der Entwicklungs-Plattform Firebase übernommen. Firebase bietet verschiedene Sign-in Methoden an. Neben der klassischen E-Mail/Passwort Kombination stellt Firebase noch die Sign-in Methoden von anderen Anbietern zur Verfügung. So kann man die Benutzerkonten von zum Beispiel Google, Facebook, Twitter oder GitHub für die Verwendung verwenden.

Für die mobile Anwendung Geogram wird die klassische Sign-in Methode „E-Mail und Passwort“ und die Sign-In Methode „Google“ verwendet.

### 3.2.2 Cloud Firestore

Die verwendete Firestore-Datenbank ist eine in der Cloud gehostete NoSQL-Datenbank. Über native SDKs können die iOS-, Android- und Web-Apps auf Firestore zugreifen.

In dem NoSQL-Datenmodell von Firestore, werden die Daten in Dokumente gespeichert, welche Felder enthalten, denen Werte zugeordnet sind. Diese Dokumente wiederum werden in Sammlungen (collections) abgespeichert. Diese hierarchische Rangordnung der Datenspeicherung ist nochmals in Abbildung 3.3 abgebildet.



Für Geogram werden die zwei Sammlungen „images“ und „users“ benötigt. Wie die Bezeichnungen schon vermuten lassen, werden darin die Fotos und die Benutzer von Geogram abgespeichert.

Abb. 3.3: Speicherstruktur Firestore

Die Dokumente der Sammlung „users“ bilden die verschiedenen Benutzerprofile von Geogram ab. Ein Benutzerprofil wird durch folgende Felder definiert:

- **biography:** Eine kurze Beschreibung über den Benutzer
- **email:** Die E-Mail Adresse des Benutzers
- **likedImage:** Array mit den Image-IDs der Bilder, welche ein Like bekommen haben
- **profilepic:** URL zu dem Profilbild, welches im Foto-Server abgespeichert ist
- **userFirstName:** Vorname des Benutzers
- **userLastName:** Nachname des Benutzers
- **username:** Username des Benutzers

In Abbildung 3.4 ist ein beispielhaftes users-Dokument abgebildet.

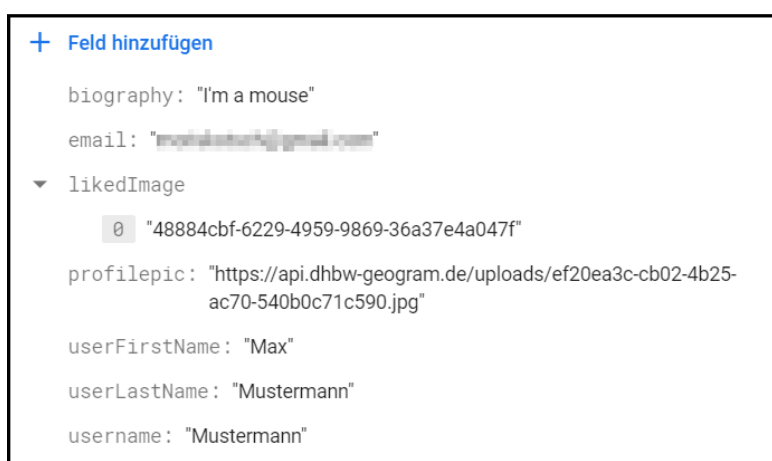


Abb. 3.4: Felder eines „users“-Dokumentes

Die Dokumente der Sammlung „images“ bilden die verschiedenen Bilder von Geogram ab. Ein Bild wird durch folgende Felder definiert:

- **comments:** Array mit allen Kommentaren und dazugehörigen Informationen
- **description:** Beschreibung des Bildes
- **id:** Eindeutige Id für das Bild
- **location:** GPS-Informationen für das Bild
- **timestamp:** Zeitpunkt des Hochladens des Bildes
- **title:** Titel des Bildes
- **url:** URL zu dem Bild, welches im Foto-Server abgespeichert ist
- **user:** Username des Bild-Inhabers

In Abbildung 3.5 ist ein beispielhaftes images-Dokument abgebildet.



Abb. 3.5: Felder eines „images“-Dokumentes

### 3.2.3 Foto-Server

Um das kostenpflichtige Lizenzmodell von Firebase zu umgehen, wurde ein eigener Foto-Server bereitgestellt. Aufgrund des begrenzten Speicherplatzes (max. 1 GiB) im Firestore, werden nur noch die URL-Adressen der jeweiligen Bilder im Firestore abgelegt. Zu sehen ist das Abspeichern der URL-Adressen in den Abbildungen Abbildung 3.4 und Abbildung 3.5. Die URL-Adressen sind alle nach folgendem URL-Schema aufgebaut:

[https://api.dhbw-geogram.de/uploads/{image\\_id}.jpg](https://api.dhbw-geogram.de/uploads/{image_id}.jpg)

Für die Umsetzung des Foto-Servers wurde sich für das Web-Framework „Express“ entschieden. Es zählt zu den bekanntesten Web-Frameworks, welche in der Node.js Laufzeitumgebung beheimatet ist.

Für die Verarbeitung von eingehenden Web-Anfragen, verwendet Express eine Art Pipeline-Mechanismus. In Listing 3.2 ist ein Ausschnitt aus dem Quellcode des Foto-Servers abgebildet. Im Code wird hierfür zunächst ein Express-Objekt erstellt. Mit der `use`-Methode können der Pipeline verschiedene Middlewares hinzugefügt werden. So gibt es zum Beispiel die Möglichkeit, bei der Verarbeitung der Web-Anfrage, einen Logger oder JSON-Parser hinzuzufügen. Die Express-Pipeline orientiert sich am REST-Paradigma. Der Foto-Server hat daher die `get`- und `post`-Methoden implementiert. Mit den beiden Methoden besteht die Möglichkeit ein Foto entweder dem Server hinzuzufügen (`post`) oder abzufragen (`get`).

```
1    ...
2
3    var app = express();
4
5    app.use(logger("dev"));
6    app.use(express.json({ limit: "50mb" }));
7    app.use(express.urlencoded({ extended: false, limit: "50mb" }));
8    app.use(cookieParser());
9    app.use(express.static(path.join(__dirname, "public")));
10   app.use(cors());
11
12   app.get("/upload1/:data", (req, res) => {
13       ...
14   });
15
16   app.post("/upload1", (req, res) => {
17       ...
18   });
19
20   ...
```

Listing 3.2: Codeausschnitt aus dem Foto-Server

Gehostet und verwaltet wird der Foto-Server, wie beschrieben, von Paul Finkbeiner.

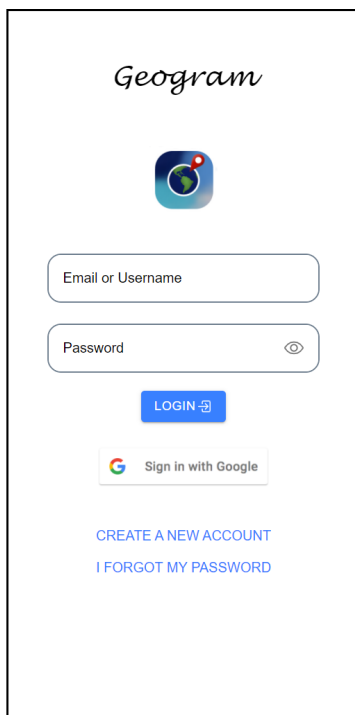


# Kapitel 4

## 4 User-Guide

Im folgenden Kapitel wird ein Überblick über alle implementierten Funktionalitäten aufgezeigt. So dient folgendes Kapitel auch als eine Art „User-Guide“.

### 4.1 Login-/Logout-/Nutzerverwaltung



Für den Anmeldevorgang benötigt der Besucher ein gültigen Account. Für die eigentliche Anmeldung wird das Passwort mit entweder der E-Mail-Adresse oder dem Usernamen benötigt. Zudem besteht auch die Möglichkeit sich mit einem Google-Account anzumelden.

Sind die angegebenen Anmeldeinformationen nicht korrekt wird dem Besucher ein PopUp mit der Nachricht „**Your Login credentials are incorrect**“ angezeigt.

Neben dem Anmelden kann man über diese Ansicht noch die Funktionalitäten „**Registrierung**“ und „**Password vergessen**“ aufrufen.

Abb. 4.1: Login - Ansicht

Register

Username

First Name

Last Name

Email

Password

Confirm Password

REGISTER

Betätigt man auf der Login-Ansicht den Button „**CREATE A NEW ACCOUNT**“, so gelangt man auf die Registrierungs-Ansicht. Für einen neuen Account benötigt man folgende Informationen:

- **Username**
- **Vorname**
- **Nachname**
- **E-Mail**
- **Passwort**

Das Passwort muss folgende Bedingungen erfüllen:

- **Mindestens 6 Zeichen lang**
- **mind. eine Zahl**
- **mind. ein Sonderzeichen**

Abb. 4.2: Registrierungs - Ansicht

Geogram

Please Type In your Email

Email

SEND

Sign in with Google

CREATE A NEW ACCOUNT

I FORGOT MY PASSWORD

Falls der Besucher sein Passwort für seinen Account vergessen hat, so kann er mit der „Password-Vergessen“-Funktionalität ein neues Passwort hinterlegen. Hierfür muss er seine gültige Account-E-Mail angeben und erhält von Geogram eine E-Mail zugesendet. In dieser E-Mail befindet sich ein Link, mit welchem der Besucher seinem Account ein neues Passwort hinterlegen kann.

Abb. 4.3: E-Mail-Vergessen - Ansicht

## 4.2 Profil - Ansicht

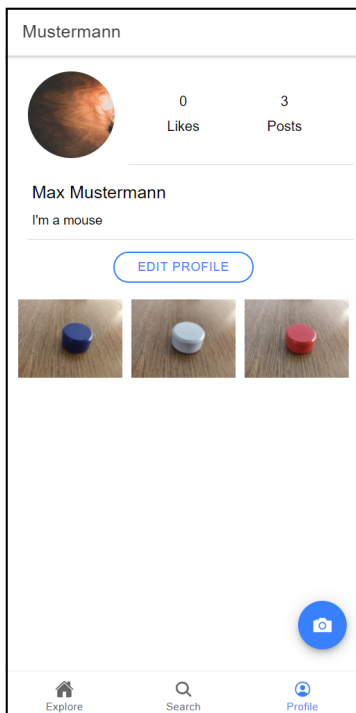


Abb. 4.4: Profil - Ansicht

In der Profil-Ansicht hat der Benutzer die Möglichkeit Informationen über sein eigenes Profil zu sehen.

Zu sehen sind folgende Informationen:

- **Profilbild**
- **Anzahl erhaltener Likes**
- **Anzahl veröffentlichter Posts**
- **Vor- und Nachname**
- **Biographie (Text über sich selbst)**
- **Die eigenen Posts**

Zusätzlich hat man noch die Möglichkeit folgende Funktionalitäten auf der Profil-Ansicht auszuführen:

- **Ändern des Profilbildes (Durch drücken des Bildes)**
- **Profil bearbeiten**
- **Einmalige Verifizierung der E-Mail**

Mit drücken eines Posts öffnet sich das PopUp aus Abbildung 4.5, über welches man auf die Post-Info aus Abbildung 4.6 gelangt.

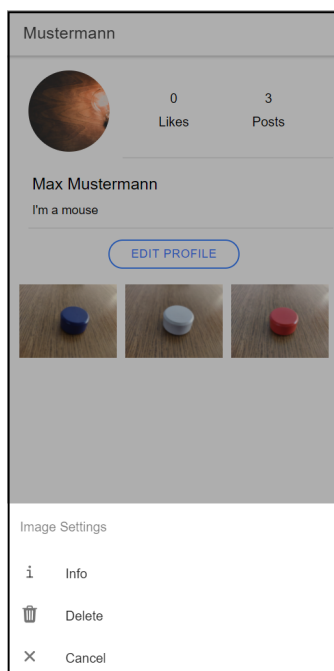


Abb. 4.5: Post-PopUp

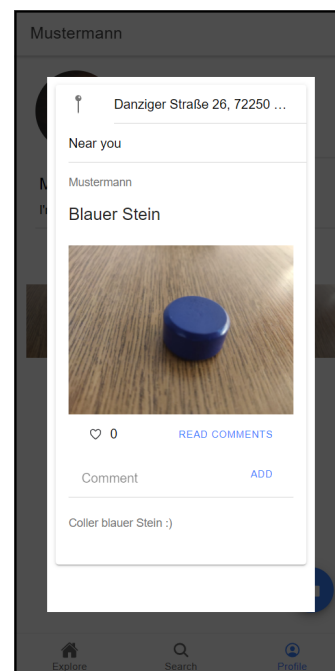


Abb. 4.6: Post Informationen

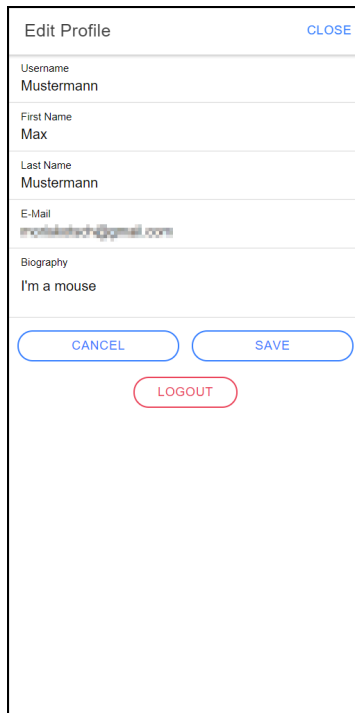


Abb. 4.7: Edit Profil - Ansicht

In dieser Ansicht hat der Benutzer zusätzlich die Möglichkeit die Informationen des Profils zu überarbeiten.

Bearbeitet werden können folgende Informationen:

- **Benutzername**
- **Vorname**
- **Nachname**
- **E-Mail**
- **Biographie (Text über sich selbst)**

Zusätzlich hat man hier die Möglichkeit sich von Geogram abzumelden. Man gelangt dann wieder zur Login-Ansicht.

## 4.3 Suchen - Ansicht

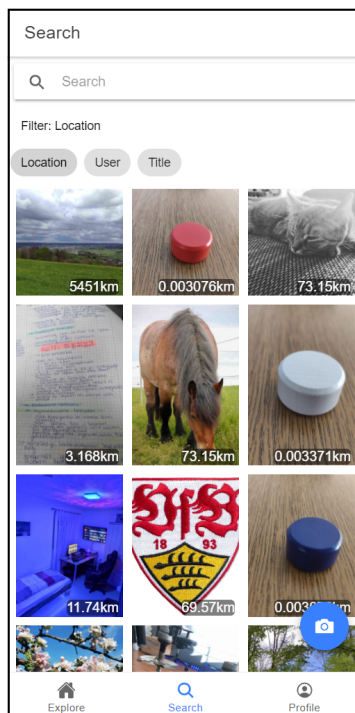


Abb. 4.8: Suchen - Ansicht

Mit dem Tab „**Search**“ gelangt der Benutzer auf die nebenstehende Ansicht. Hier hat der Benutzer die Möglichkeit nach bestimmten Posts zu suchen.

Bei den einzelnen Posts werden jeweils die Bilder und die dazugehörigen Entfernungen zum Benutzer angezeigt und erst beim Drücken eines Posts erhält man eine Info-Ansicht, wie aus Abbildung 4.6.

Zu Beginn werden alle verfügbaren Posts angezeigt. Der Benutzer hat jedoch auch die Möglichkeit die angezeigten Posts zu filtern. Gefiltert werden kann nach der Entfernung, dem User oder dem Titel.

## 4.4 Explore - Ansicht

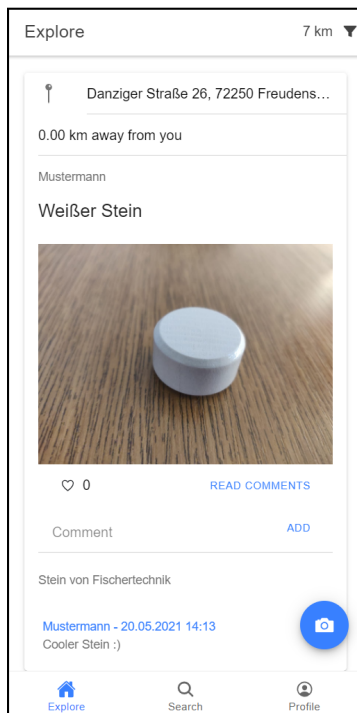


Abb. 4.9: Explore - Ansicht

Mit dem Tab „**Explore**“ gelangt der Benutzer auf die nebenstehende Ansicht. Hier werden dem Benutzer die vollständigen Posts angezeigt.

Dem Benutzer werden die Posts abhängig von der Entfernung entweder angezeigt oder nicht. Um den Radius der Entfernung zu ändern, kann der Benutzer diesen oben rechts einstellen. Wird der Radius auf „**30+**“ gestellt, so werden alle Posts angezeigt.

Bei jedem Post hat der Benutzer die Möglichkeit ein Like und Kommentare abzugeben. Den eigenen Kommentar und die Kommentare von anderen Benutzern kann man ebenfalls unter den Posts sehen.

# Kapitel 5

## 5 Veröffentlichung im Play Store

Ziel war neben der Implementierung der mobilen Anwendung „Geogram“ auch die Veröffentlichung der gleichen App im Google Play Store. Um rechtliche Problematiken bezüglich der Ähnlichkeit zu der App „Instagram“ vorzubeugen, wurden an der App „Geogram“ Änderungen vorgenommen.

Eine Änderung bezieht sich auf das verwendete Logo der App „Geogram“. In Abbildung 5.1 ist oben links das alte App-Logo zu erkennen. Verglichen mit dem Instagram-Logo (rechte Seite) ist eine Ähnlichkeit deutlich zu erkennen. Als Gruppe hat man sich deshalb dafür entschlossen ein neues Logo für die App „Geogram“ zu entwerfen. Das neue Logo der App „Geogram“ ist in Abbildung 5.1 unten links abgebildet. Es gibt nun keine Verwechslungsgefahr mehr mit dem Logo der App „Instagram“.

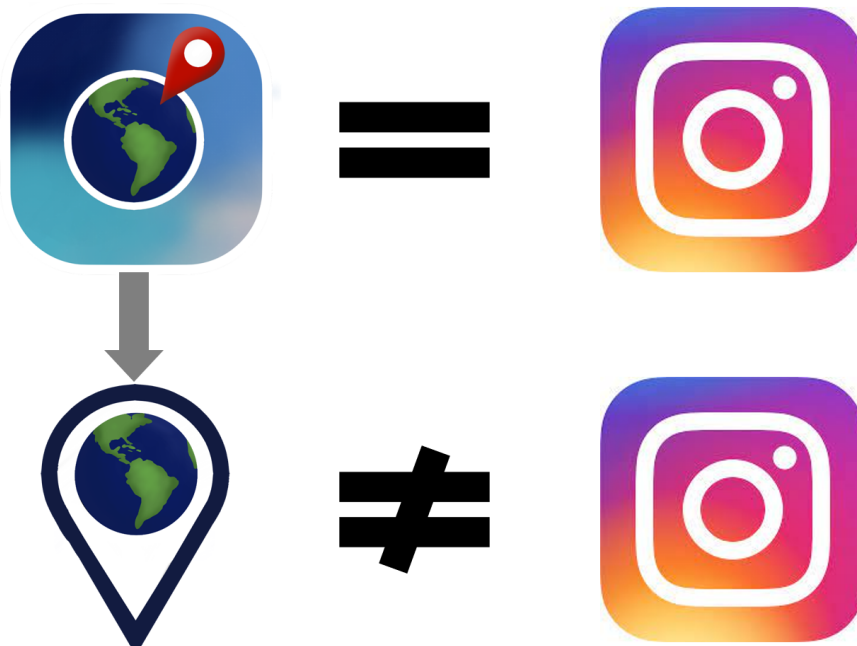


Abb. 5.1: Übersicht der Logos

Neben dem Logo wurde zusätzlich noch der Name der App angepasst. Der bisherige Name „Geogram“ wurde in den Namen „GeoShare“ umgeändert. Grund ist die Ähnlichkeit der Namen „Geogram“ und „Instagram“, bezüglich dem Teil „gram“.

Um die Veröffentlichung der App „GeoShare“ im Google Play Store zu ermöglichen, waren noch zusätzliche Schritte notwendig. Wie in Abbildung 5.2 zu erkennen, wurde

die App mit dem Label „**USK ab 18 Jahren**“ gekennzeichnet. Dies beruht auf der Tatsache, dass das Entwicklerteam keine Verantwortung über die hochgeladenen Inhalte übernimmt und potenzielle Benutzer auch kinderschädigende Bilder posten könnten.

Ebenso notwendig, war die Bereitstellung einer Datenschutzerklärung, welche vom Entwicklerteam zur Verfügung gestellt werden muss. Jene Datenschutzerklärung ist unter dem Link „<https://datenschutz.dhbw-geogram.de/>“ zu finden.

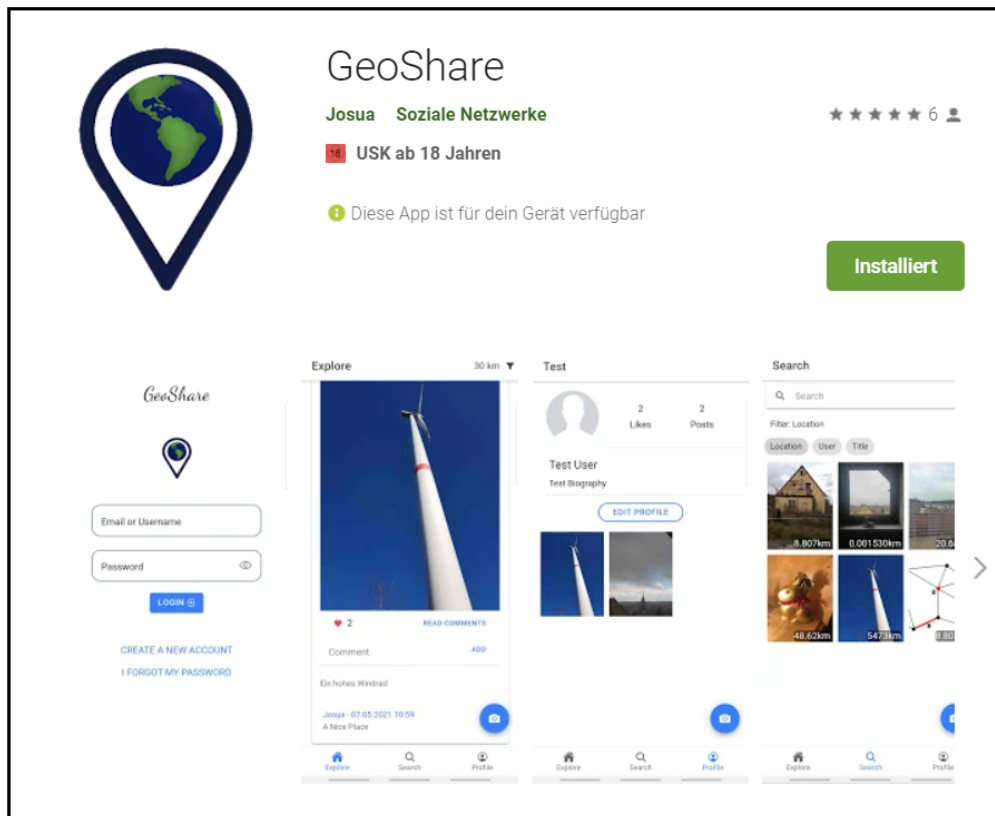


Abb. 5.2: Die App im Play Store

Sollte das Entwicklerteam neue Features umsetzen, ist das Aktualisieren der App im Play Store mithilfe von Updates in wenigen Minuten möglich.