

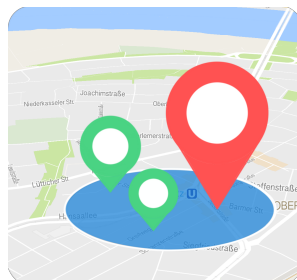
Heinrich-Heine-Universität Düsseldorf  
Sommersemester 2016  
Institut für Sprache und Information  
Projektseminar Angewandte Informationswissenschaft  
Dozentin: Maria Henkel  
Student: Paul Degemann  
Github: [padgm](#)  
Matrikelnummer: 2206653

22.09.2016

# Projektplan & Projektdokumentation

## -

## Locations Around You (Instagram)



**LAY!**

# Inhaltsverzeichnis

1 Projektidee & Projektplanung	1 - 5
1.1 Idee	1
1.2 Konzept	1
1.3 Aufbau und Funktionen	2
1.3.1 Suchfunktion	2
1.3.2 Menüfunktion	3
1.4 Bibliotheken	3
1.5 Google & Instagram API	4
1.6 Mögliche Probleme	4
2 Projektumsetzung	5 - 13
2.1 Projektaufbau	5 - 6
2.1.1 Klassen	5 - 6
2.1.1.1 StartActivity.java	5
2.1.1.2 LocationActivity.java	6
2.1.1.3 MapsActivity.java	6
2.1.1.4 DownloadTask.java	6
2.2 Anleitung zur Ausführung/Benutzung	7 - 9
2.2.1 Installation	7
2.2.2 Benutzung	7
2.3 Reflexion	9 - 12
2.3.1 Probleme in der Programmierung	10
2.3.2 Probleme mit der Instagram und Google Maps API	11
3 Fazit	12

# **1 Projektidee & Projektplanung**

## **1.1 Idee**

Das Projekt soll auf Basis einer Android Applikation (App) erstellt werden. Die App soll eine Weltkarte aufzeigen, die die populärsten Hashtags an den Orten, Ländern und Stellen anzeigt, an denen sie von Instagram Nutzern verwendet wurden. Damit kann man anhand der Weltkarte erkennen, wo zu welchem Hashtag die meisten Bilder bzw. Videos gepostet werden. Es ist somit auch möglich lokale Hashtags zu nutzen, was von Interesse von Firmen, Cafés oder auch Personen, wie beispielsweise Bloggern sein kann. Aber es besteht auch die Möglichkeit bedeutende, weltweite Hashtags zu finden, sofern ein breiteres Publikum angesprochen werden soll.

## **1.2 Konzept**

Über die Instagram API lassen sich mittels „location ID“ Orte und die dort genutzten Hashtags ausfindig machen. So können die verwendeten Daten genutzt werden um in der App auf der Karte angezeigt werden zu können. Über einen bestimmten API Link lässt sich eine JSON Liste von Daten über Ort, Nutzernamen und verwendete Hashtags abrufen. Hierbei müssen eine „min\_id“ und eine „max\_id“ eingefügt werden um die dazwischen liegenden location IDs zu extrahieren (instagram.com, o.S.). Diese JSON Liste, stellt ein Array dar, welches mit Java eingelesen werden kann. Mittels der GSON Bibliothek gelangt man an die Geoinformation und an die Hashtags, da diese die Möglichkeit eröffnen auf bestimmte Stellen in der JSON Datei zuzugreifen. Es wird in der Programmiersprache Java programmiert um eine Android App erstellen zu können. Hinzu wird Android Studio von IntelliJ IDEA verwendet, da dies die offizielle IDE zur Entwicklung von Android Apps von Google ist (developer.android.com, o.S.). Damit kann über ein .xml-Dokument alles eingebunden werden, wie Buttons, SearchViews, ImageViews, was benötigt wird. Es wird von Android Studio automatisch Gradle benutzt, obwohl Maven auch eine Alternative

darstellen würde. Welche Bibliotheken benutzt werden wird in Abschnitt 1.4 *Bibliotheken* beschrieben.

## 1.3 Aufbau und Funktionen

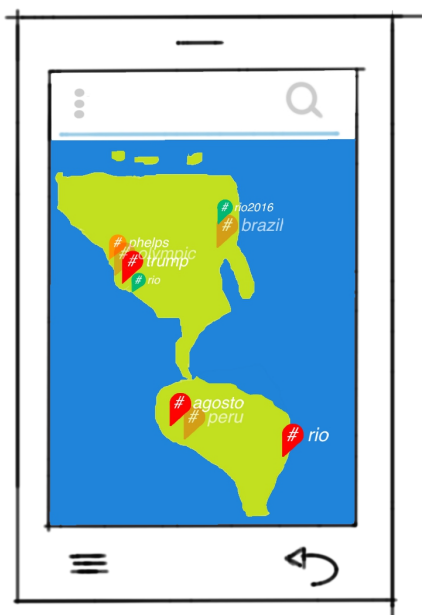


Abb. 1: Skizze Karten-/Markeraufbau

Der Grundaufbau besteht hauptsächlich aus einer Weltkarte. Diese beinhaltet die normalen Google Maps Eigenschaften, sodass Orte, Länder und vieles Weitere angezeigt wird. Die Hashtags werden mittels Markierungen angezeigt und je nach Bedeutung eingefärbt und in ihrer Größe variiert. Zudem befindet sich im oberen Bereich des Bildschirm auf der linken Seite das Menü, das symbolisiert wird durch drei einfache Punkte. Auf der Rechten befindet sich die Suchfunktion, die durch eine Lupe kenntlich gemacht wird. Um die Funktion von Google Maps zu erreichen benötigt es einer Google API, wobei der OAuth Key in den Code eingefügt werden muss.

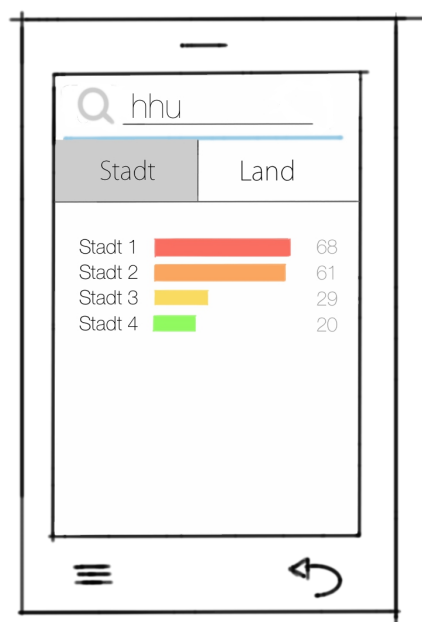


Abb. 2: Skizze Suchfunktion

### 1.3.1 Suchfunktion

Die Suchfunktion ermöglicht es dem Nutzer nach einem speziellen Hashtag zu suchen und dann zwischen Stadt und Land zu wählen. Daraufhin wird ein Ranking erstellt, dass die Städte bzw. Länder aufzeigt, die den Hashtags am häufigsten verwendet haben.

### 1.3.2 Menüfunktion

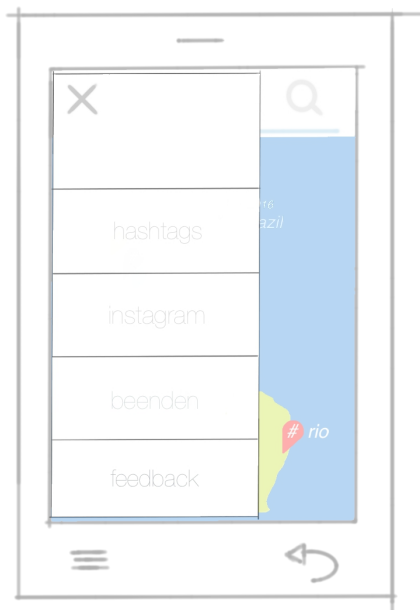


Abb. 3: Skizze Menüfunktion

Die Menüfunktion beinhaltet vier Auswahlmöglichkeiten. Zum Einen „hashtags“, die die aktuell populärsten Hashtags anzeigt. Zum Anderen auch „instagram“, was den Nutzer zum Instagram-Account der App führt. Des Weiteren gibt es den Punkt „beenden“, der die App schließt und zu letzt noch die Auswahlmöglichkeit „feedback“. Feedback erlaubt es dem Nutzer Rückmeldung zu dieser App zu geben. Der Nutzer wird also zum Playstore geleitet, sofern die App dort erscheint, oder kann per Kontaktformular sein Feedback abgeben.

## 1.4 Bibliotheken

Für diese App müssen viele verschiedene offizielle, aber auch eventuell inoffizielle Java-Bibliotheken in Anspruch genommen werden. Eine inoffizielle Bibliothek stellt die Instagram API in Java dar. Eine derartige Bibliothek findet man unter folgendem Link: <https://github.com/sachin-handiekar/jInstagram#instagram-endpoints> (diese nutzt die Bibliothek: <https://github.com/scribejava/scribejava>).

Es wird die Google GSON Bibliothek benötigt, um die JSON Dateien verarbeiten zu können. Auch wird eine Bibliothek benötigt, die die Google Maps Eigenschaften verarbeitet, bspw: <https://dzone.com/articles/google-maps-java-swing>.

Die offizielle Bibliotheken `java.util` sollte importiert werden, um `ArrayLists` oder auch den `Scanner` verwenden zu können. Dies wird unter anderem zum Speichern von Eingaben benötigt.

## **1.5 Google & Instagram API**

Um die Karte von Google Maps verwenden zu können muss die Google API genutzt werden. Die Google Maps Karte kann mit dem vom Android Studio vorgegebenen Google Maps Template eingebunden werden, indem der Authentifizierungsschlüssel in die `google_maps_api.xml` Datei eingefügt wird.

Schwieriger gestaltet sich dies bei der Instagram API. Hier wird wie oben beschrieben eine `max_id` und eine `min_id` verwendet die in den API-Link eingefügt werden muss. Leider ist nicht direkt ersichtlich wo und wie dies geschehen muss. Darüberhinaus sind hier lediglich 500 Anfragen die Stunde erlaubt, sodass es schwierig ist eine Menge von Daten zu erheben, die die ganze Weltkarte abdeckt (`instagram.com`, o.S.).

## **1.6 Mögliche Probleme**

Bei dieser Anwendung ist es möglich, dass die Umsetzung nicht vollständig die Anforderungen erfüllen kann und es zu Problemen kommt. Ein potentiell Problem könnte die Instagram API darstellen, die eventuell nur bedingt alle nötigen Informationen zur Verfügung stellt, sodass nicht alle Elemente der App angezeigt werden können. Darunter fällt auch die Geoinformation der Instagram API, da diese über Location IDs erfolgt, sodass nicht direkt ersichtlich ist wo der Post abgesetzt wurde. Eine mögliche Lösung wäre die Instagram Seite mit einem Crawler zu durchlaufen und so an Ortsnamen zu gelangen und zusätzlich an die verwendeten Hashtags.

Des Weiteren kann nicht vollständig gewährleistet werden, dass die Visualisierung genau dem entspricht, was auf den oben aufgeführten Skizzen zu sehen ist. Der Grund ist, dass zeitliche oder auch programmiertechnische Probleme auftreten können. Ein Problem stellt

beispielsweise die Google Maps Karte dar, die ebenfalls nur durch eine API von Google ihre Funktion erhält. Hier könnte man auch eine normale Weltkarte einbinden, die nicht die interaktiven Eigenschaften von Google Maps besitzt.

## **2 Projektumsetzung**

### **2.1 Projektaufbau**

Da es sich um eine Android App handelt wurde die Programmiersprache Java und die IDE Android Studio basierend auf IntelliJ IDEA verwendet. Java stellt die grundsätzliche Programmiersprache für Android Apps dar ([developer.android.com](http://developer.android.com), o.S.). Android Studio ist eine Entwicklungsumgebung, die von Google als offizielle freie integrierte IDE für Android Projekte zum Download freigegeben ist ([developer.android.com](http://developer.android.com), o.S.).

#### **2.1.1 Klassen**

Es werden ausschließlich die wichtigsten Klassen auf ihre Funktionsweise hin erklärt, sodass die Programmarchitektur verständlich wird.

##### **2.1.1.1 StartActivity.java**

Die StartActivity ist die Klasse, die als erstes aufgerufen wird und alle Einstellungen beinhaltet, die der Nutzer vornehmen kann. Diese Klasse zeigt auf dem Display den Startbildschirm an. Hierbei handelt es zunächst um die Klasse, von der aus das weitere Programm ausgeführt wird. Es werden daraufhin entweder die LocationActivity.java oder die MapsActivity.java, aber immer die DownlaodTask.java ausgeführt.

### 2.1.1.2 LocationActivity.java

Die LocationActivity.java zeigt auf dem Smartphone ein grünes Display und einen Ladebalken. Wenn dieser Ladebalken vollständig geladen ist, wird die MapsActivity.java Klasse gestartet und der Nutzer wurde geortet.

### 2.1.1.3 MapsActivity.java

Android Studio erstellt diese Klasse, sofern man eine App mit Map Funktion auswählt, automatisch. Hierbei handelt es sich um die Klasse, die die Google Maps Karte anzeigt und verarbeitet. Auf einer Google Maps Karte gibt es den Zusatz jegliche Bilddatei als Marker zu verwenden, sodass folgende Marker selbst gestaltet und eingefügt wurden.



*Abb. 4: Marker aktueller Standort*



*Abb. 6: Marker um aktuellen Standort*



*Abb. 5: Marker um eingegeben Ort*

### 2.1.1.4 DownloadTask.java

Die Instagram Daten werden durch die DownloadTask.java zur Verfügung gestellt. Diese Klasse verarbeitet die Instagram Schnittstelle und lädt über den Instagram API Link, in Form einer JSON-Datei, die Daten herunter und verarbeitet diese, sodass die Daten auf der MapsActivity.java veranschaulicht werden können.



## 2.2 Anleitung zur Ausführung/Benutzung

### 2.2.1 Installation

Voraussetzung um die App entsprechend nutzen zu können, ist ein Android Smartphone. Hierbei ist zu beachten, dass die Android Version 6.0 mit Google API 23 installiert ist, da ansonsten gewisse Funktionen der App, wie das Lokalisieren des Smartphones, nicht korrekt ausgeführt werden können. Möglich ist auch Android Version 5.1 mit der Google API 22, jedoch wurde dies nur auf einem Emulator getestet. Ein Emulator wäre somit ebenfalls eine Möglichkeit die App zu nutzen.

Zunächst muss unter den Einstellungen des Smartphone das Installieren von unbekannten Quellen aktiviert werden. Daraufhin kann die App in Form einer apk-Datei 'aroundU.apk' bspw. von Github heruntergeladen werden und durch einen Klick installiert werden. Daraufhin ist die App auch ausführbar, indem man auf 'Öffnen' klickt. Dies sollte soweit alles ohne weitere Probleme funktionieren. Sofern dies nicht der Fall ist, gibt es von [Chip](#) eine ausführliche Anleitung inklusive Video.

### 2.2.2 Benutzung

Die App startet mit der Startseite, wo sich von oben nach unten, ein Textfeld, ein Button, zwei Einstellleisten und wieder ein Button befinden. Das Textfeld dient dazu eine Ortssuche durchzuführen. Wenn man den gewünschten Ort eingegeben hat, lässt sich über die zwei Einstellleisten noch der Radius einstellen, der um den Ort herum untersucht werden soll und die Anzahl der anzuzeigenden Orte um den eingegebenen Ort. Nun kann man auf den Button „Eingeben“ klicken, der einem zu der GoogleMaps Karte führt. Man kann auch die Möglichkeit via GPS nutzen.

Wenn man die Parameter Anzahl und Radius eingestellt hat kann man auf den Button Lokalisieren klicken und gelangt über eine

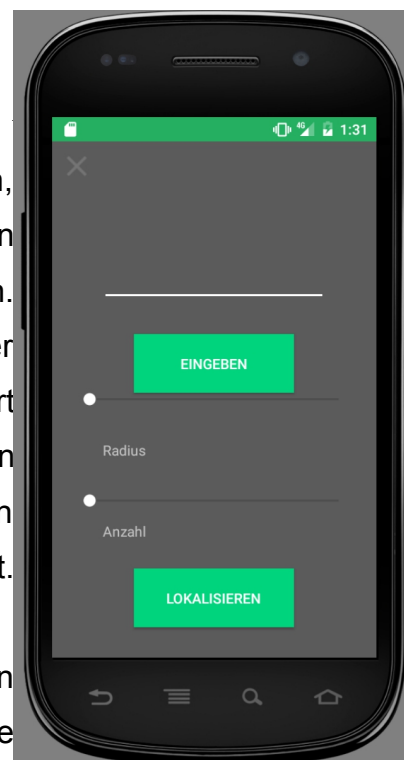
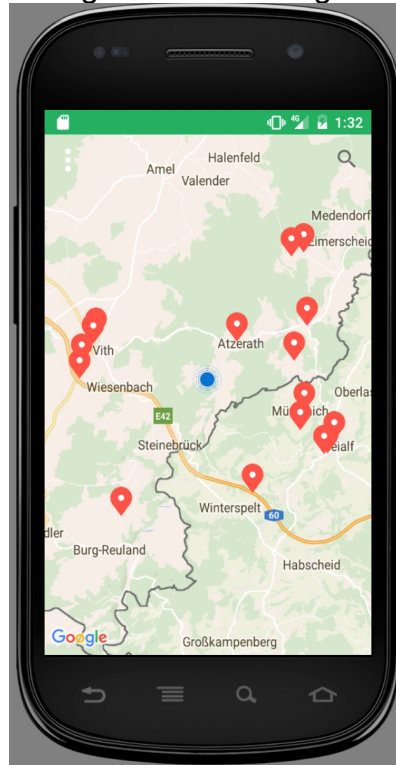


Abb. 7: Startbildschirm

kurze Phase des Positionssuchens zu der GoogleMaps-Karte.

Dort werden um den, ob nun per Eingabe oder Ortung ermittelten Ort, die von Instagram



*Abb. 8: Karte mit Markern*

und deren Nutzern genutzten Orte angezeigt. Die Marker sind genau auf diesen Orten gesetzt und anklickbar. Sofern man einen Marker anklickt erscheint der Titel über dem Marker mit dem Namen des Ortes. Wenn man nun sogar noch den Titel anklickt gelangt man über den Browser zu Instagram, wo nach diesem Ort als Hashtag gesucht wird. Zurück zur LAY! App gelangt man mit dem Zurück-Button des Smartphones. Nun gibt es auf der GoogleMaps-Karte noch weitere Eigenschaften. In der oberen, rechten Ecke befindet sich ein Lupe, die bei Klick eine Suchleiste öffnet. Diese Suchleiste ermöglicht das Suchen nach einem weiteren Ort. Im Grunde verhält sich alles wie zuvor, jedoch werden durch diese Suchleiste gesuchte Orte mit grünen Markern umgeben. Im Endeffekt sind alle Marker grün, die nicht durch den Startbildschirm gesucht bzw. geortet wurden oder angeklickt sind.

In der oberen, linken Ecke befindet sich der Button für das Menü. Dieser beinhaltet vier weitere Buttons. Mit dem ersten Button „Radius“ kommt man zurück zum Start, wo man erneut Radius und weitere Einstellungen vornehmen kann. Der zweite Button erfüllt den

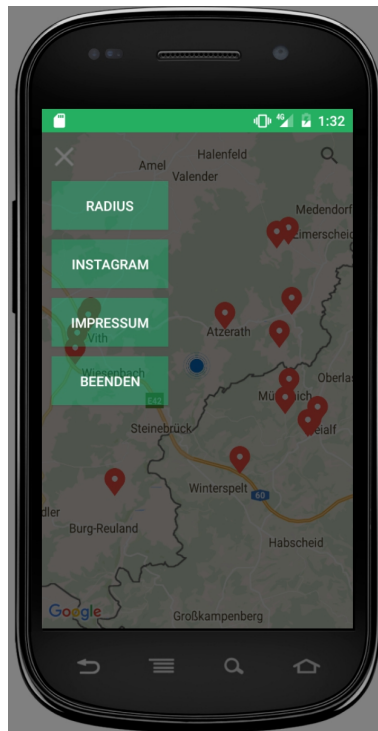


Abb. 9: geöffnetes Menü

Zweck, dass man zum Instagram Profil der App gelangt. Der dritte Button zeigt das Impressum der App und der vierte beendet diese.

## 2.3 Reflexion

Mit Blick auf den zuvor aufgeführten Projektplan und der zuvor aufgeführten Projektidee, zeigt sich deutlich, dass sich das Projekt zum Teil geändert hat. Es war angedacht eine Weltkarte mit Hashtags zu versehen, die von der Instagram API bereitgestellt werden und demnach von Instagram genutzt werden und eingetragen sind. Der Grund für die Veränderung liegt in den Problemen die unter 8.1 und speziell 8.2 folgen werden. Dennoch blieb der Grundgedanke erhalten. Fertiggestellt wurde eine Android App, die die Instagram API nutzt und über den Location Endpoint, Orte auf eine Google Maps Karte markiert.

Diese App hatte das Ziel für Blogger, Unternehmen, Cafés oder andere Instagram-Nutzer

eine Übersicht zu bieten, um zu sehen was in der Umgebung geschieht und aktuell ist. Dieses Ziel wurde zwar nicht komplett erfüllt, aber auch nicht verfehlt. Denn die App zeigt die Orte in der Umgebung an, die viel Aufschluss für die zuvor genannten Instagram-Nutzer bieten. Hinzu gelangt man von den Orten zu Instagram und kann sich die, mit dem Ort getagten Bilder anschauen und interagieren. So bleibt das Ziel erhalten und kann sogar einen Mehrwert bieten, da sonst die Orte wegfallen würden und nur die Hashtags angezeigt werden würden.

### 2.3.1 Probleme in der Programmierung

Während der Programmierung der App traten vereinzelt Probleme im Bereich von Location Tracking und XML auf.

Einer der aufwendigsten Anteile in der Programmierung war die Ortung des Smartphones. Nach Jakuben (2014) gibt zwei Wege, um an die Daten des mobilen Endgeräts zu gelangen. Zum Einen die Location API, die von Android mitgeliefert wird und der hier gewählte Weg mit den Google Play Services. Letzteres hat aber auch den Nachteil, dass auf die jeweilige Android Version geachtet werden muss. Google Play Services benötigen unter anderem einen Authentifizierungsschlüssel für eine Google API, die auch für die Google Map und andere Google Services genutzt werden kann. Diesen Authentifizierungsschlüssel kann man unter diesem [Link](#) erstellen. Bei dem Erstellen des Schlüssels, ist es möglich sich mittels SHA1-Fingerabdruck seinen Schlüssel einschränken zu lassen. Um diesen ganzen Vorgang nachvollziehen zu können, sollte die [Google Dokumentation](#) gelesen werden.

Ebenfalls benötigt wird die Erlaubnis vom Smartphone bzw. Smartphonebenutzer den Standort über das Internet oder GPS zu ermitteln. Hier hört der eigentliche Java Programmiererteil auf, sodass man sich mit der XML Struktur vertraut machen muss. Es gibt dabei zwei Berechtigungen, die Access Coarse Location und die Access Fine Location. Erstere „ermöglicht die Verwendung von WLAN- und/oder Mobiltelefonaten durch die API, um den Gerätestandort zu ermitteln. Der Standort wird von der API mit einer Genauigkeit von etwa einem Häuserblock zurückgegeben“ (developers.google.com, o.S.) und die zweite Berechtigung „ermöglicht eine möglichst präzise Standortbestimmung

durch die API unter Verwendung von Daten von Standort-Providern, einschließlich GPS (Global Positioning System) sowie WLAN- und Mobiltelefonaten“ (developers.google.com, o.S.) . Die sogenannten Permissions müssen in die AndroidManifest.xml Datei eingebunden werden. An diesem Punkt kam es zu gewissen Problemen, da durch das Einbinden des externen Moduls von [lorensiuswlt](#) ebenfalls AndroidManifest.xml vorhanden sind. Darüber hinaus gab es durch die Einbindung weitere AndroidManifest.xml und build.gradle Dateien, die um alle Module herum gelagert wurden. Es wurden für das Projekt auch nicht alle in 1.4 aufgeführten Bibliotheken verwendet. Für die Nutzung der Instagram Daten wurde das externe Modul von lorensiuswlt als externe Bibliothek verwendet und um die JSON-Dateien zu verarbeiten wurde nicht GSON sondern org.json von Android genutzt.

Unter anderem gab es auch Konflikte bei TextViews mit dem API Level. Da mindestens API 21 gefordert wurde, konnten die TextViews nur für diese API erstellt werden. Zudem wurde von der IDE darauf verwiesen, anstatt einer direkten Eingabe eines Strings für ein Textfeld, über die Resource Datei strings.xml einen String einzutragen und diesen zu verlinken. Dies konnte aber ignoriert werden, da es sich nur um Warnungen handelte.

Um eine externe Bibliothek nutzen zu können, musste das externe Projekte zunächst als externes Modul importiert werden und konnte darauf hin als Dependency hinzugefügt werden, was Anfangs auch eine gewisse Herausforderung darstellte.

Des Weiteren wurde zuvor nicht mit Activities programmiert, sondern die verschiedenen Schaltflächen und Elemente entweder sichtbar oder unsichtbar gesetzt. Dies wurde durch das Starten von verschiedenen Activities abgeändert und deutlich vereinfacht. Dennoch ist es manchmal hilfreich mit Sichtbarkeit bzw. Unsichtbarkeit zu arbeiten. Diese Variante wurde bspw. für das Menü auf der MapsActivity benutzt.

### **2.3.2 Probleme mit der Instagram und Google Maps API**

Das größte Problem stellt wohl die Instagram API dar. Hierbei wurde nicht nur die Zugriffsrate begrenzt, wodurch die App nicht mehr als 500 Anfragen pro Stunde tätigen darf (instagram.com, o.S.), sondern auch durch den Sandbox Modus, der weitere Einschränkungen wie die Nutzung nur bestimmter API-Links zur Folge hatte. Die Idee, an

den Instagram Orten die benutzten Hashtags anzuzeigen konnte aus diesem Grund ebenfalls nicht verwirklicht werden, da die Anfrage:

[https://api.instagram.com/v1/locations/{location-id}/media/recent?access\\_token=XXXX](https://api.instagram.com/v1/locations/{location-id}/media/recent?access_token=XXXX)

als Antwort eine inhaltslose JSON-Datei zurück gibt. Um dennoch sinnvoll die Orte in der Umgebung zu nutzen, konnte folgende API Anfrage verwendet werden:

[https://api.instagram.com/v1/locations/search?  
lat=48.858844&lng=2.294351&access\\_token=XXXX&distance=10&count=20](https://api.instagram.com/v1/locations/search?lat=48.858844&lng=2.294351&access_token=XXXX&distance=10&count=20)

Damit gab es keine inhaltslose JSON-Datei, sodass diese Informationen genutzt werden konnten, um die Orte und ihre Titel auf der Karte anzeigen zu können. Zudem ist es bei dieser Anfrage möglich Radius und Menge der Ergebnisse einzugeben, was auf der App mit den Einstellleisten eingestellt werden kann. Leider scheint hier ein gewisser Fehler in der API vorzuliegen. Der Radius, im Link als „distance“ bezeichnet, wird laut instagram.com (2016) in Metern angegeben, ist bei einer leeren Angabe 500m und hat ein Maximum von 750m. Dennoch werden Orte um den Standort angezeigt, die deutlich weiter entfernt liegen. Dazu konnte auch bei stackoverflow keine Antwort gefunden werden: [stackoverflow.com/questions/39561943/correct-link-instagram-api](https://stackoverflow.com/questions/39561943/correct-link-instagram-api). Aus diesen Gründen war es auch nicht möglich alle Funktionen, die vorher beschrieben wurden zu verwirklichen.

Mit der Google API gab es nicht direkt Probleme, jedoch musste darauf geachtet werden in jede google\_maps\_api.xml Datei den Schlüssel einzusetzen und auch im values Ordner in den richtigen Dateien die ID einzufügen.

### **3 Fazit**

Die App wird den zuvor gestellten Anforderungen nicht vollständig gerecht. Speziell in Bezug auf die Anzeige der Hashtags muss diese Application Einbußen machen. Dennoch kann diese Anwendung den meisten Anforderungen gerecht werden, sodass sie den

Zweck erfüllt, den es zu erfüllen galt.

Man kann festhalten, dass dieses Projekt eine App hervorgebracht hat, die es in dieser Form noch nicht gibt. Zusätzlich kann diese App noch enorm ausgebaut werden und mit weiteren Features und Funktionen versehen werden. Im Hinblick auf die Programmierung erbrachte diese einen tiefen Einblick in die Welt der Android App Entwicklung, die wahrscheinlich deutlich größer, aber auch deutlich interessanter ist, als es auf den ersten Blick scheint.

## Literaturverzeichnis

- Jakuben, B. (2014, Copyright 2016). *The Beginner's Guide to Location in Android*. blog.teamtreehouse.com. Abgerufen am 22.09.2016, von <http://blog.teamtreehouse.com/beginners-guide-location-android>.
- Standortdaten (2016). developers.google.com. Abgerufen am 22.09.2016, von <https://developers.google.com/maps/documentation/android-api/location?hl=de>
- Rate Limits (2016). [instagram.com](https://www.instagram.com). Abgerufen am 22.09.2016, von <https://www.instagram.com/developer/limits/>
- Location Endpoints (2016). [instagram.com](https://www.instagram.com). Abgerufen am 22.09.2016, von <https://www.instagram.com/developer/endpoints/locations/>
- Fundamentals (2016). developer.android.com. Abgerufen am 22.09.2016 von <https://developer.android.com/guide/components/fundamentals.html>
- Meet Android Studio (2016). developer.android.com. Abgerufen am 22.09.2016 von <https://developer.android.com/studio/intro/index.html>
- <https://github.com/lorensiuswt/AndroidInstagram>