

# Tweebo

## Ein Geo-Basierter Twitter Client für das iPad

### BACHELORARBEIT

zur Erlangung des akademischen Grades

### Bachelor of Science

im Rahmen des Studiums

### Medieninformatik und Visual Computing

eingereicht von

**Matthias Schötta**

Matrikelnummer 0326357

und

**August Nikolaus Kampfer**

Matrikelnummer 0326465

an der

Fakultät für Informatik der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller

Mitwirkung: Dipl.-Ing. Johanna Schmidt

Wien, 11.01.2014

\_\_\_\_\_  
(Unterschrift Verfasser)

\_\_\_\_\_  
(Unterschrift Betreuung)



# Erklärung zur Verfassung der Arbeit

Matthias Schötta  
Lerchengasse 25, 1080 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

---

(Ort, Datum)

---

(Unterschrift Verfasser)



# Erklärung zur Verfassung der Arbeit

August Nikolaus Kampfer  
Semperstraße 18, 1180 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

---

(Ort, Datum)

---

(Unterschrift Verfasser)



# Kurzfassung

Twitter ist eine online Plattform, welche sich darauf spezialisiert hat Kurznachrichten, auch Tweets genannt, zu erstellen und in weiterer Folge online zu publizieren. Sobald ein Benutzer ein entsprechendes Twitter-Konto erstellt hat, hat er die Möglichkeit Kurznachrichten zu verfassen und per Favoritensystem andere Nachrichten-Streams zu verfolgen. Jeder Benutzer erhält seine eigene Twitter-Website, auf welcher seine Nachrichten einzusehen sind. Twitter wird auch gerne als Mikroblogging Plattform bezeichnet. Neben Facebook ist Twitter einer der großen Namen im Bereich der sozialen Netzwerke.

Kurznachrichten, welche auf 140 Zeichen limitiert sind, bilden die Basis der Plattform. Diese enthalten neben der eigentlichen Nachricht, Metadaten, welche weitere Verarbeitungsebenen ermöglichen. Unter anderem stellt Twitter eine umfangreiche REST-API zur Verfügung. Diese ermöglicht es Drittanbietern die Funktionalität von Twitter in andere Applikationen zu integrieren. Die Zurverfügungstellung dieser Schnittstelle wirkt einer Kapselung der Twitter-Umgebung entgegen und motiviert neue Einsatzmöglichkeiten.

Tweebo stellt eine dieser neuen Einsatzmöglichkeiten dar, fokussiert sich speziell auf den Standortaspekt eines Tweets und kann als geo-basierter Twitter Client für iOS, welcher eine wertvolle Kombination aus reinen Nachrichtenaspekten und Geolokalisierungen kombiniert, bezeichnet werden. Er wurde speziell für iOS entwickelt. Durch die außergewöhnliche Darstellung von Tweets auf einem dreidimensionalen Globus, wird es dem Benutzer ermöglicht, schnell Veränderungen im Weltgeschehen zu erkennen. Verschiedene Visualisierungsmöglichkeiten stellen erstmals eine geobasierte Darstellung von Tweets, auf einem mobilen Gerät ansprechend zur Verfügung. Tweebo versucht einen globalen Bezug hinsichtlich der Twitteraktivitäten eines Benutzers zu visualisieren. Anstelle einer gewohnten Listenansicht ermöglicht Tweebo, durch Verarbeitung der zusätzlichen Lokalisierungsdaten pro Kurznachricht, eine angepasste Ansicht auf einem dreidimensionalen Globus. Neben der eigentlichen Lokalisierung der Kurznachrichten

auf dem Globus ermöglicht Tweebo auch die Visualisierung der Aktivitätsintensität eines Nachrichtenstreams.

Der Tweebo iOS Client unterstützt die primären Twitter-Funktionalitäten. Natürlich können Kurznachrichten erstellt und abgerufen werden, diese Funktionalitäten werden aber mit einer zusätzlichen Visualisierungsebene, dem dreidimensionalen Globus erweitert.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Problemstellung . . . . .	2
1.2	Methodischer Ansatz . . . . .	3
<b>2</b>	<b>Verwandte Arbeiten</b>	<b>5</b>
2.1	TwitInfo . . . . .	5
2.2	Limosa . . . . .	7
2.3	TwitterReporter . . . . .	8
2.4	Erdbebenerkennung mit Hilfe von Twitter . . . . .	9
2.5	TwitterHitter . . . . .	9
2.6	Aktuelle Twitter iOS Apps . . . . .	10
<b>3</b>	<b>Implementierung</b>	<b>13</b>
3.1	iOS Entwicklung . . . . .	13
3.2	Verwendete Technologien . . . . .	15
3.3	Tweebo Funktionalität . . . . .	23
<b>4</b>	<b>Ergebnisse</b>	<b>31</b>
4.1	Funktion 1 - Die Suche . . . . .	32
4.2	Funktion 2 - Die Aktivitätsanzeige/Hotspots . . . . .	32
4.3	Funktion 3 - Message browsing . . . . .	34
4.4	Einschränkungen . . . . .	35
<b>5</b>	<b>Schlussfolgerungen und ein Ausblick</b>	<b>39</b>
	<b>Literaturverzeichnis</b>	<b>41</b>



## Einleitung

Twitter [10] ist ein soziales Echtzeit-Informationsnetzwerk, welches 2006 gegründet wurde und dann rasch an Popularität gewonnen hat. Es basiert auf Kurznachrichten, sogenannten „Tweets“, welche eine Länge von 140 Zeichen nicht überschreiten dürfen und zwischen den Usern von Twitter ausgetauscht werden. Tweets können über eine Webseite, mobile Geräte oder Applikationen von Drittherstellern gesendet werden. Der User hat die Möglichkeit Tweets privat oder öffentlich zu senden. Wird ein Tweet als öffentlich markiert erhalten alle User, welche dem Tweet-Autor „folgen“, die Nachricht. Wem gefolgt wird, ist den Twitter-Usern selbst überlassen. Durch dieses System wird Twitter hauptsächlich benutzt um Nachrichten an alle „folgenden“ User zu senden. Diese Nachrichten können jedoch auch noch später von allen anderen Usern durchsucht werden, jeder Twitter-User hat somit Zugriff auf sämtliche Diskussionen und Themen, welche auf der ganzen Welt diskutiert werden.

Aufgrund der eingeschränkten Länge der Tweets enthalten diese meist aktuelle Nachrichten, dienen Marketingzwecken, sind Teile von Konversationen, oder sind persönliche Statusmeldungen privater Personen. Tweets können Antworten auf ältere Tweets, sogenannte „Retweets“ sein, und Links zu weiteren Informationen, Bildern oder Videos enthalten.

Alles in allem kann Twitter also als „Mikro-Blogging“ Plattform angesehen werden, da die User ihre Gedanken, Meinungen, Erlebnisse oder Neuigkeiten austauschen und der Öffentlichkeit zur Verfügung stellen können.

Twitter ist in den letzten Jahren zu einem mächtigen Kommunikationswerkzeug herangewachsen. Schneller als auf traditionellen Plattformen, können sich User hier über

Nachrichten, globale Interessen und Diskussionen, oder einfach nur über Neuigkeiten bei Freunden und Verwandten informieren.

Mittlerweile hat Twitter über 200 Millionen aktive User welche mittlerweile über 170 Milliarden Tweets produziert haben [13]. All diese Informationen sind öffentlich zugänglich und zusätzlich für Entwickler über eine Programmierschnittstelle (API) erreichbar. Dies ermöglicht es, die riesige Menge an Informationen in weiteren Anwendungen zu verarbeiten und wie im Falle von Tweebo, dem User auf alternativen Wegen zu präsentieren.

Nicht nur der veröffentlichte Text wird über die API übertragen, sondern es werden auch weitere Informationen über den Autor des Tweets werden zur Verfügung gestellt. 8,2% der Tweets enthalten Daten über den genauen Aufenthaltsort des Autors, und in 30% der Tweets kann die ungefähre Ursprungsposition eines Tweets auf anderen Wegen, wie zum Beispiel der angegebenen Heimatstadt des Tweet-Autors, herausgefunden werden. Die Öffentlichmachung dieser Daten ist freiwillig und muss vom Autor des Tweets explizit zugelassen werden, was zur Folge hat, dass nicht alle Tweets Daten über den Aufenthaltsort des Tweet-Autors beinhalten. Betrachtet man aber die Daten und den Trend der letzten Jahre, so erkennt man schnell, dass immer mehr Twitter-User ihr Position sichtbar machen. Somit ist zu erwarten, dass eine eindeutige Positionierung der Tweets in Zukunft noch einfacher möglich sein wird.

## **1.1 Problemstellung**

Das Ziel von Tweebo ist es, die geobasierten Daten in Twitter zu nutzen um dem User aktuelle News und Themen auf einem alternativen Weg zu präsentieren. In Tweebo werden Tweets in einer iOS App, welche speziell für das iPad entwickelt wurde, auf einem dreidimensionalen Globus angezeigt und lassen sich nach Kategorien filtern. So kann schnell ein Überblick über die wichtigsten Themen in der jeweiligen Region unserer Erde gewonnen werden. Allerdings wird in diesem Projekt das Hauptaugenmerk nicht auf das Sammeln der Daten, sondern auf die Anzeige der Daten gelegt. Andere Projekte wie zum Beispiel Fujita [2] haben sich bereits ausgiebig mit der Problemstellung der Sammlung der Daten beschäftigt.

## **1.2 Methodischer Ansatz**

Nach heutigen Standards wird eine iOS App entwickelt, welche den Grundstein für dieses Projekt legt. Ein dreidimensionaler Globus, welcher unsere Erde repräsentiert dient als Hauptnavigationselement. Sobald die App gestartet wird, werden im Hintergrund neue Tweets von Twitter abgerufen, verarbeitet, gefiltert, und auf dem Gerät persistiert. Sollte der User seinen Twitter-Login auf dem iPad hinterlegt haben, macht sich Tweebo das zu nutze und berücksichtigt die persönlichen Interessensgebiete des Users. Sobald diese Informationen verfügbar sind, werden dem User, nun erstmals auf einem mobilen Gerät, Tweets je nach Kategorie und Position auf dem Globus angezeigt. Schnell können so aktuelle Trends und Themen sichtbar werden. Weitere Informationen zu den Tweets kann sich der User natürlich auch gleich in der App holen.



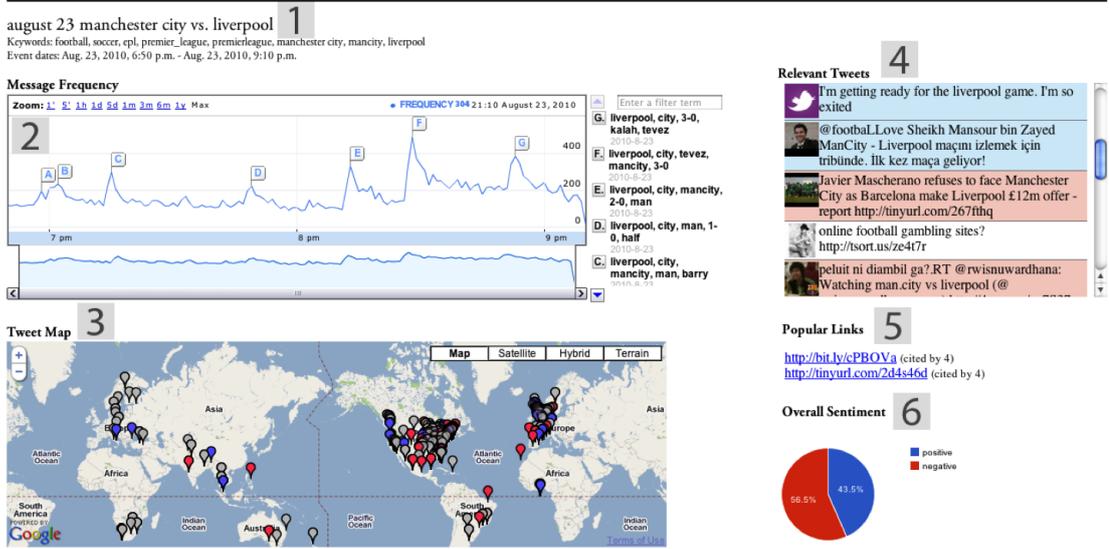
## Verwandte Arbeiten

Da Twitter sehr viele, teilweise sehr wertvolle und vor allem sehr schnell aktuelle Informationen bereitstellt und dies, wie oben schon erwähnt, mittels einer Programmierschnittstelle, existiert mittlerweile eine große Menge an wissenschaftlichen Arbeiten und Projekten, welche versuchen diese Informationen zu analysieren, auszuwerten und für den Menschen in eine lesbare und erfassbare Form zu bringen. Dies ist jedoch keines Falls einfach, da eine solche Flut an Informationen erstmal gefiltert und dargestellt werden will.

Einige dieser Projekte beschäftigen sich, so wie Tweebo, vor allem mit der Lokalisierung der Tweets. Die wichtigsten Arbeiten wollen wir an dieser Stelle kurz erwähnen, da diese Denkanstöße gaben und auch zur Lösung des ein oder anderen Problems beigetragen haben.

### 2.1 TwitInfo

TwitInfo [14] basiert auf realen Events. Ein intelligenter Algorithmus macht es möglich besonders hohe Twitter-Aktivitäten zu erkennen. Sollten zu einem bestimmten Zeitpunkt sehr viele Tweets mit ähnlichem Inhalt gefunden werden, ist meist ein reales Event dafür verantwortlich. Genau diese Events erkennt TwitInfo und stellt diese dem User webbasiert da. User können sogar Events noch genauer untersuchen und weitere Informationen sammeln, oder dazugehörige kleinere Events erforschen. Natürlich spielt bei realen Events meist auch der Standort eine Rolle, weswegen User die Möglichkeit bekommen, auf dem Standort des Events weiter nach Informationen zu suchen.



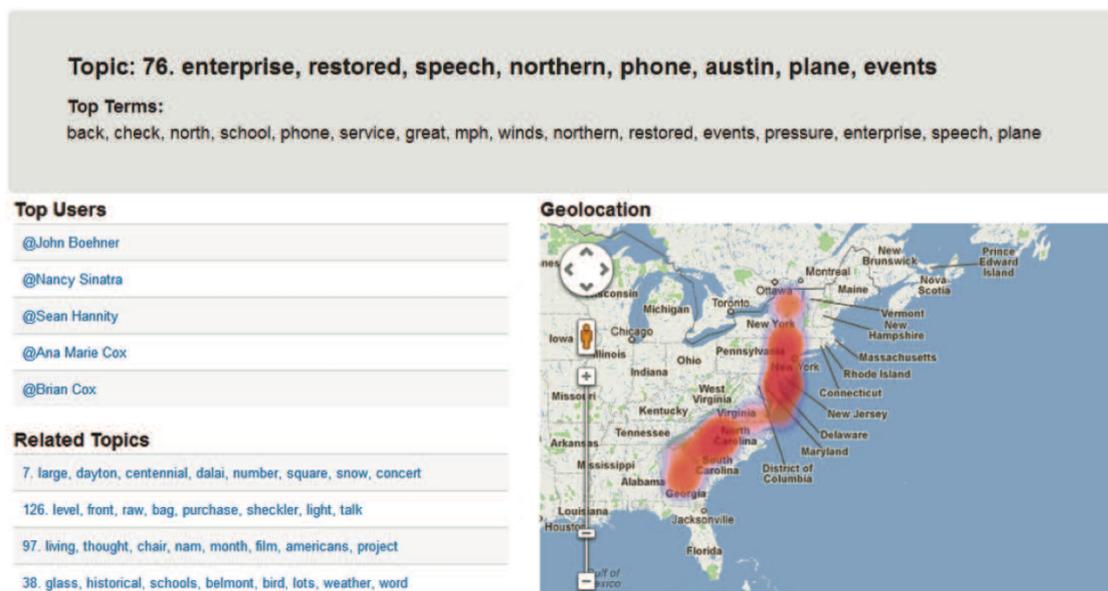
**Abbildung 2.1:** TwitInfo Benutzeroberfläche: 1) Der Name des vom Benutzer erstellten Events und dessen Stichworte. 2) Die Zeitlinie, die X-Achse repräsentiert die Zeit die Y-Achse die Anzahl der relevanten Tweets zum gegebenen Zeitpunkt. 3) Relevante Tweets, welche genaue Daten über die Position enthalten werden auf einer zweidimensionalen Karte angezeigt. 4) Relevante Tweets werden in einer Liste angezeigt. Rot erscheinen Tweets mit einer negativen Stimmung, blau Tweets mit einer positiven Stimmung 5) Die beliebtesten Links werden zusätzlich angezeigt. 6) Die Stimmung aller relevanten Tweets wird durch ein Kreisdiagramm dargestellt.

Um TwitInfo zu verwenden müssen User zunächst ein Event mittels eines Keywords erstellen, mit Hilfe dessen relevante Tweets gesucht, gespeichert, und auf einer Webseite, sowohl auf einer Zeitachse als auch auf einer Karte dargestellt werden (siehe Abbildung 2.1). Auch die Stimmung eines Tweets kann durch Farben schnell abgelesen werden, so lassen sich positive und auch negative Tweets auf einen Blick erkennen. Regelmäßig wird die Ansicht upgedatet um stets die neuesten Informationen zu liefern. Um den Standort eines Tweets zu bestimmen verwendet TwitInfo entweder die genaue geographische Position, welche wie in Kapitel 1 erwähnt, in ca. 8% der Fälle mitgeliefert wird, oder sollte diese nicht mitgeliefert werden, die Heimatstadt des Tweet-Autors, welche dann mit Hilfe eines weiteren Webservices in genaue geografische Länge und Breite umgewandelt werden kann.

Es hat sich herausgestellt, dass User mit diesem System sehr schnell Zusammenfassungen und Überblicke rekonstruieren können. Gerade lang anhaltende Events können so besser verstanden werden und auch Augenzeugen lassen sich schnell durch TwitInfo finden.

## 2.2 Limosa

Auf einer anderen Idee basiert Limosa [21]. Das System stellt Interessen von Twitter-Usern geographisch dar. Limosa besteht im wesentlichen aus zwei Komponenten. Das Backend verwendet die Twitter API um auf sämtliche Tweets zuzugreifen und filtert relevante Informationen aus den gesammelten Tweets. Das Frontend ist ebenfalls web-basiert und ermöglicht dem User eine visuelle Darstellung der Tweets, und somit eine Analyse der Daten.



**Abbildung 2.2:** Limosa Frontend: Visualisierung der meistgesuchten Themen. *Besuchte* und *erwähnte* Standorte werden angezeigt.

Da Limosa, genauso wie Tweepo, auf geographische Daten der Tweets angewiesen ist, nützt das System, falls vorhanden, die geographische Position der Tweets einerseits aus dem mitgelieferten genauen Standort, greift aber auf eine weitere Methode zurück, falls

dieser nicht verfügbar sein sollte. Im Text des Tweets erwähnte Standorte werden analysiert und entweder *besuchten*, oder nur *erwähnten* Standorten zugeordnet. Dies geschieht, in dem der Text auf weitere Schlagwörter, wie zum Beispiel „Hotel“ untersucht wird. Werden vergleichbare Schlagwörter gefunden, so ist es sehr wahrscheinlich, dass sich der Tweet-Autor gerade an diesem Ort befindet. Sollte dies nicht der Fall sein handelt es sich wahrscheinlich um einen erwähnten Standort. Ausgefeilte Algorithmen werden verwendet um die jeweilige Wahrscheinlichkeit zu berechnen.

Ist die Position nun erkannt, muss der Tweet noch einem Thema zugeordnet werden und schon kann man sich im Frontend auf einer Karte mit Hilfe von Heatmaps die Interessen von Twitter Usern in bestimmten Regionen der Erde ansehen (siehe Abbildung 2.2). Limosa zielt darauf ab, eine interessante Demonstration von Sichtweisen der Twitter-User, Regionen oder speziellen Zeiträumen zur Verfügung zu stellen.

## 2.3 TwitterReporter

Ein weiteres sehr interessantes Projekt ist TwitterReporter [15], welches neue Methoden präsentiert um Daten zu sammeln, aktuelle Nachrichten beziehungsweise Eilmeldungen selbstständig zu erkennen und diese unter Berücksichtigung des Standorts und des Zeitraums zu visualisieren.

TwitterReporter hat Zugriff auf sämtliche Tweets, welche Informationen über den genauen Standort des Tweet-Autors zum Zeitpunkt der Erstellung enthalten und innerhalb der Vereinigten Staaten von Amerika erstellt wurden. Dieser Zugriff wurde speziell von Twitter genehmigt und findet in Echtzeit statt. Dies dient als Grundlage um schnell relevante Ereignisse zu erkennen. Weitere Methoden um die große Menge an Daten zu analysieren sind jedoch notwendig. So filtert TwitterReporter ausschließlich englische Tweets aus den Daten, welche dann weiter vereinfacht werden, um Schlagwörter schnell erkennen zu können. Werden Schlagwörter gefunden, wird weiter in anderen Tweets nach diesen gesucht und die Häufigkeit berechnet. Wird der Tweet nun als relevant markiert, so existiert eine neue Eilmeldung. Eilmeldungen werden auf einer Karte angezeigt, sie sind in drei Kategorien (Vom Menschen ausgelöst, Naturkatastrophen, Sonstige) unterteilt und auch die Relevanz / Häufigkeit der Tweets an einer bestimmten geographischen Position wird mit Hilfe von Farben repräsentiert (siehe Abbildung 2.3). TwitterReporter hat bewiesen, dass zum Beispiel Erdbeben durch diese Technik sehr schnell erkannt werden und meist mindestens 2-3 Stunden früher durch TwitterReporter berichtet werden, als von traditionellen Medien. Um das System weltweit einzusetzen

müsste allerdings noch eine Unterstützung sämtlicher Sprachen implementiert werden.

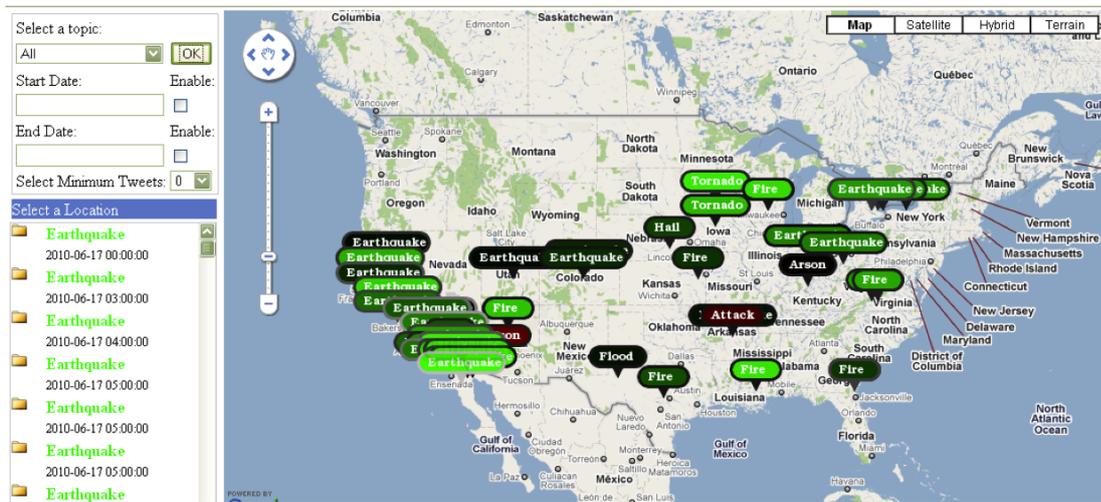


Abbildung 2.3: TwitterReporter: Benutzeroberfläche mit entsprechenden Daten

## 2.4 Erdbebenerkennung mit Hilfe von Twitter

Ähnlich wie TwitterReporter funktioniert auch „Twitter Earthquake Detection“ [1]. Wie der Name schon sagt, werden hier jedoch ausschließlich Erdbeben registriert. Auch hier wird die Frequenz an Tweets registriert. Wird eine ungewöhnlich hohe Anzahl an Tweets erkannt, werden die Tweets weiter nach dem Schlagwort „Erdbeben“ beziehungsweise dessen Übersetzung, in anderen Sprachen durchsucht. Wird das Schlagwort in einer bestimmten Anzahl von Tweets gefunden, gilt ein Erdbeben als registriert. Natürlich werden von diesem System nur Erdbeben erkannt, welche auch von Menschen wahrgenommen werden, oder in besiedelten Gebieten stattfinden. Der Großteil an Erdbeben wird nicht registriert, da diese von Menschen nicht registriert werden, dafür werden aber auch fast keine Falschmeldungen produziert. Von 48 weltweit registrierten Erdbeben gab es im Testzeitraum nur 2 Falschmeldungen.

## 2.5 TwitterHitter

Selbst Projekte, welche sich mit Verbrechensbekämpfung, beziehungsweise mit der Analyse von Verbrechen auseinandersetzen nutzen Daten, welche von Twitter bereit-

gestellt werden. TwitterHitter [22] nutzt die geographische Position der Tweets von Verdächtigen und untersucht Zusammenhänge und Verbindungen zu deren „Twitter-Freunden“, um ganze Netzwerke von Verbrechern beziehungsweise die Machthaber solcher Banden ausfindig zu machen (siehe Abbildung 2.4). Ermittler der Polizei haben die Möglichkeit Bewegungen und Gewohnheiten von vermeidlichen Verdächtigen schnell zu erkennen. Auch in diesem Projekt werden die Ergebnisse auf einer 2D-Karte präsentiert.

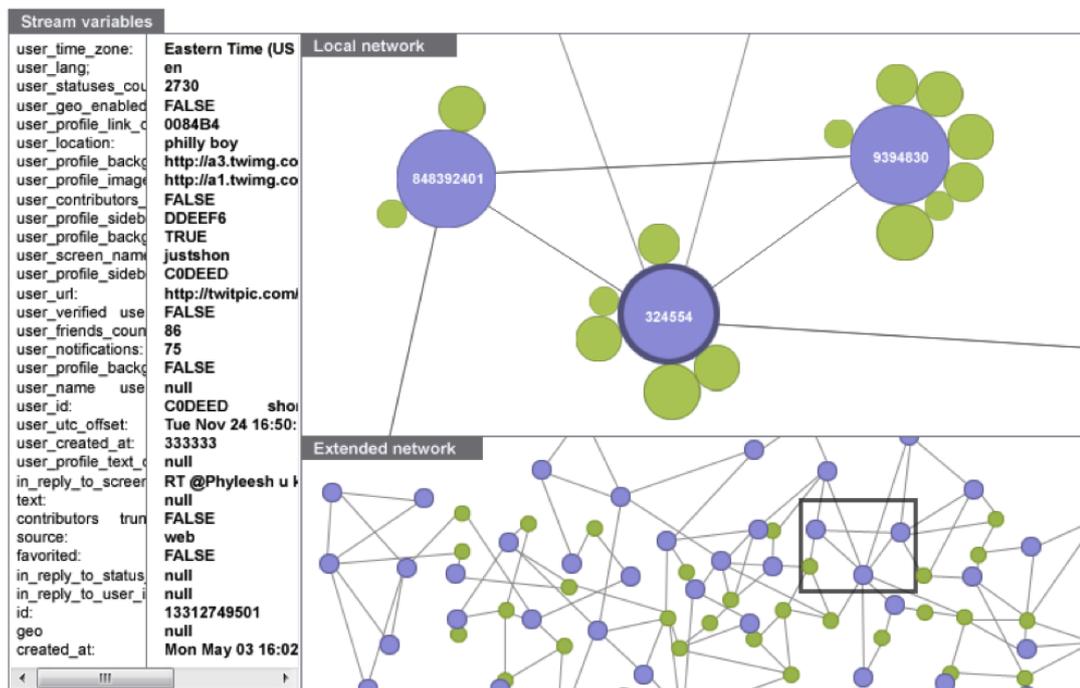


Abbildung 2.4: TwitterHitter: mögliche Verbindungen zu andern Verdächtigen

## 2.6 Aktuelle Twitter iOS Apps

Natürlich existieren zum aktuellen Zeitpunkt schon einige Twitter Clients für iOS. Die Unterschiede zwischen den angebotenen Apps sind aber relativ gering. Viele der Clients unterstützen zwar das Senden eines Tweets unter Berücksichtigung der aktuellen geographischen Position des Tweet-Autors, jedoch konnte keine App gefunden werden, welche auch bei der Anzeige fremder Tweets die Position darstellt. In Apps wie

Twitter [11], HootSuite Für Twitter & Facebook [9] oder Tweetbot [19] sind meist die gleichen Funktionalitäten implementiert.

So kann der User beispielsweise in allen genannten Apps das eigene Profil betrachten und bearbeiten. Personen oder Institutionen welchen gefolgt wird, oder diejenigen, welche einem selbst folgen, können eingesehen und gegebenenfalls aus der Liste der gefolgt Personen oder Institutionen gestrichen werden. Alle drei Apps unterstützen die Einbindung der aktuellen Position, Bilder, oder Links auf Webseiten in Tweets, welche direkt in der jeweiligen App sehr schnell und einfach erstellt werden. Auch private Tweets, Konversationen und Diskussionen lassen sich meist leicht abrufen. Das Herzstück ist in allen genannten Apps die sogenannte „Timeline“, welche einen schnellen Überblick über die aktuellsten Tweets der gefolgt Personen oder Institutionen verschafft. Schnell können auch weitere Details und verwandte Tweets zu aktuellen Tweets, welche in der „Timeline“ angezeigt werden, abgerufen werden.

Die meisten dieser Funktionalitäten werden noch nicht von Tweebo zur Verfügung gestellt, da hier das Hauptnavigationselement der dreidimensionale Globus ist und das Hauptaugenmerk auf der geographischen Positionen der Tweets gelegt wurde. Eine Einbindung der genannten Punkte wäre aber wäre eine interessante Weiterentwicklung der bestehenden Funktionalität.



## Implementierung

iOS [6] ist wohl eines der bekanntesten mobilen Betriebssysteme welche sich heutzutage auf dem Markt befinden und sollte der Leser dem Hersteller Apple [4] glauben, auch das am weitesten entwickeltste mobile Betriebssystem der Welt.

Apple legt in der Tat großen Wert auf sehr hohe Benutzerfreundlichkeit und versucht es Entwicklern zu erleichtern, ausgereifte und ansprechende mobile Applikationen zu kreieren. Eine große Erleichterung für Entwickler ist die Tatsache, dass iOS zumindest offiziell nur auf Hardware, welche von Apple hergestellt wird, zur Verfügung steht. So ist die Anzahl an zu unterstützenden Geräten sehr überschaubar, was eine starke Optimierung der Apps auf genau diese Geräte ermöglicht.

Alles in allem ist iOS die perfekte Plattform um schnell ansprechende und benutzerfreundliche Apps zu entwickeln. So fiel die Entscheidung bei der Suche nach dem richtigen Betriebssystem für Tweebo auch eben genau auf dieses.

### 3.1 iOS Entwicklung

Grundsätzlich sei erwähnt, dass iOS Applikationen nur auf Hardware, welche von Apple vertrieben wird, entwickelt werden kann. Gerade, wenn es an das Testen der App geht, ist man auf Werkzeuge wie XCode [8] angewiesen, um die App auf ein Gerät zu spielen. XCode ist eine integrierte Entwicklungsumgebung, welche nur für Apples aktuelles Betriebssystem Mac OS X zur Verfügung steht. Mac OS X darf laut Lizenzvertrag wiederum nur auf entsprechender Apple - Hardware installiert werden.

Mit XCode ist es möglich Apps für Macs, also Computer mit einer installierten Version

von Mac OS X, iPhones und iPads zu kreieren. Es besteht neben den Standardwerkzeugen wie Editor, Compiler und Debugger aus einigen weiteren Komponenten, die es dem Entwickler ermöglichen sehr schnell und effizient Ideen in ansprechende Programme umzusetzen. Natürlich erkennt der Compiler schon während der Eingabe Fehler. Xcode ist grundsätzlich für die Programmiersprachen C, C++ und Objective-C gedacht, aber auch in anderen Programmiersprachen kann damit entwickelt werden. Objective-C<sup>1</sup> und die Cocoa-Frameworks<sup>2</sup> sind die wichtigsten Bestandteile um native iOS Apps zu entwickeln.

Der sogenannte Interface Builder<sup>3</sup> wird dazu benutzt die grafische Oberfläche der Mac- oder iOS-Programme zu designen und gleichzeitig die Verbindungen zum Code, also der Logik zu erstellen. Auch Instruments<sup>4</sup> ist ein mächtiges Werkzeug, welches zum Analysieren der bestehenden Apps dient. Bei richtiger Anwendung können hier Probleme mit dem Speichermanagement oder Geschwindigkeitseinbußen, welche sich vermeiden lassen, schnell erkannt werden. Natürlich ist auch das iOS Software Development Kit<sup>5</sup> (SDK) gleich in XCode integriert und auch die API - Dokumentation ist sehr schnell über XCode zu erreichen.

Einen sehr wichtigen Teil bei der Entwicklung entsprechender Apps übernimmt der iOS Simulator<sup>6</sup>, welcher ebenfalls zu den Bestandteilen der XCode Werkzeuge gehört. Er ermöglicht es den Entwicklern die Software ohne Umwege direkt auf dem Gerät, welches zur Entwicklung verwendet wird, zu testen. Gerade die Benutzeroberfläche lässt sich in dieser Form gut und schnell überprüfen. Grundsätzlich ist es möglich, die gesamte Applikation mit Hilfe des iOS-Simulators zu entwickeln. Da dieser aber nur die Software der iOS Geräte simuliert, nicht aber die Hardware, (wie es beispielsweise ein Emulator machen könnte), darf keinesfalls auf Tests am echten Gerät verzichtet werden. Nicht nur Probleme beim Speicher Management oder beim Zusammenspiel mit anderen Apps, auch limitierte Hardwareressourcen fallen meist erst auf dem echten Gerät auf. Auch Probleme, welche durch eine langsame, oder nicht vorhandene Internetanbindung entstehen können, werden meist erst zu diesem Zeitpunkt erkannt. Nicht zu vergessen ist

---

<sup>1</sup><http://de.wikipedia.org/wiki/Objective-C>

<sup>2</sup><https://developer.apple.com/technologies/mac/cocoa.html>

<sup>3</sup>[http://de.wikipedia.org/wiki/Interface\\_Builder](http://de.wikipedia.org/wiki/Interface_Builder)

<sup>4</sup><https://developer.apple.com/library/mac/documentation/DeveloperTools/Conceptual/InstrumentsUserGuide/Introduction/Introduction.html>

<sup>5</sup><https://developer.apple.com/technologies/ios/>

<sup>6</sup>[https://developer.apple.com/library/ios/documentation/IDEs/Conceptual/iOS\\_Simulator\\_Guide/Introduction/Introduction.html](https://developer.apple.com/library/ios/documentation/IDEs/Conceptual/iOS_Simulator_Guide/Introduction/Introduction.html)

auch die Benutzerfreundlichkeit, welche erst getestet werden kann, wenn die Applikation mit den Fingern statt mit der Maus, und vielleicht sogar unterwegs, (beispielsweise in einem öffentlichen Verkehrsmittel), benutzt wird.

Spätestens zu diesem Zeitpunkt, um also Anwendungen auf iPhone und iPad zu installieren, muss der Entwickler Mitglied in einem der kostenpflichtigen iOS Developer Programme [7] sein. Wird die Applikation lediglich auf dem iOS Simulator ausgeführt, ist dies nicht notwendig. Diese Tatsache ist gerade für Einsteiger sehr hilfreich, da die gesamte Software, welche zur Entwicklung notwendig ist, von Apple's Webseite kostenlos heruntergeladen werden kann und somit die nötigen Kenntnisse ohne zusätzliche Kosten erworben werden können.

Verschiedene Varianten des Entwicklerprogramms stehen zur Verfügung. Hier wollen wir nur kurz auf das Standardprogramm eingehen, da die restlichen hauptsächlich für größere Firmen und Institutionen interessant sind.

Das iOS Developer Programm wird wie eine Mitgliedschaft gehandhabt und ermöglicht das Testen der entwickelten Applikationen auf bis zu einhundert Testgeräten, auch durch Beta-Tester. Weiters ist das weltweite Bereitstellen der, vom Apple Review Team freigegebenen, kostenlosen oder kostenpflichtigen Apps über den iTunes App Store, im Preis inbegriffen. Die Mitgliedschaft kostet jährlich 99,00 US Dollar.

## 3.2 Verwendete Technologien

Bei der Entwicklung wurde, um einerseits Zeit zu sparen und andererseits das Benutzererlebnis zu optimieren, auf Komponenten, welche von Apple zur Verfügung gestellt werden, als auch auf Open Source Komponenten zurückgegriffen. Die wichtigsten wollen wir in diesem Kapitel vorstellen.

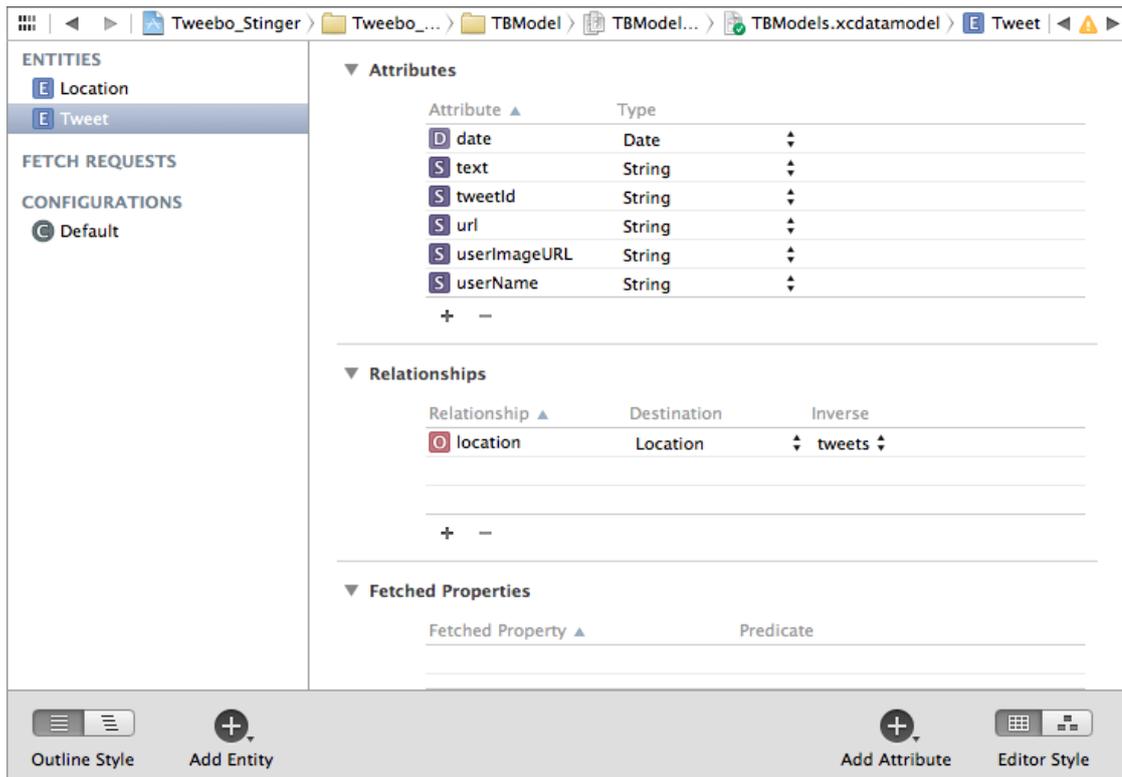
### Core Data

Core Data [5] wird direkt von Apple zur Verfügung gestellt und ist ein Framework für die Verwaltung und Persistierung von Objekten. Es serialisiert Objective-C Objekte und legt dieses in einer SQLite<sup>7</sup> Datenbank, im Arbeitsspeicher oder als XML, ohne weiteres zutun des Entwicklers, ab. Es agiert ähnlich wie das Enterprise Objects Framework<sup>8</sup>,

---

<sup>7</sup><http://www.sqlite.org/>

<sup>8</sup>[http://en.wikipedia.org/wiki/Enterprise\\_Objects\\_Framework](http://en.wikipedia.org/wiki/Enterprise_Objects_Framework)



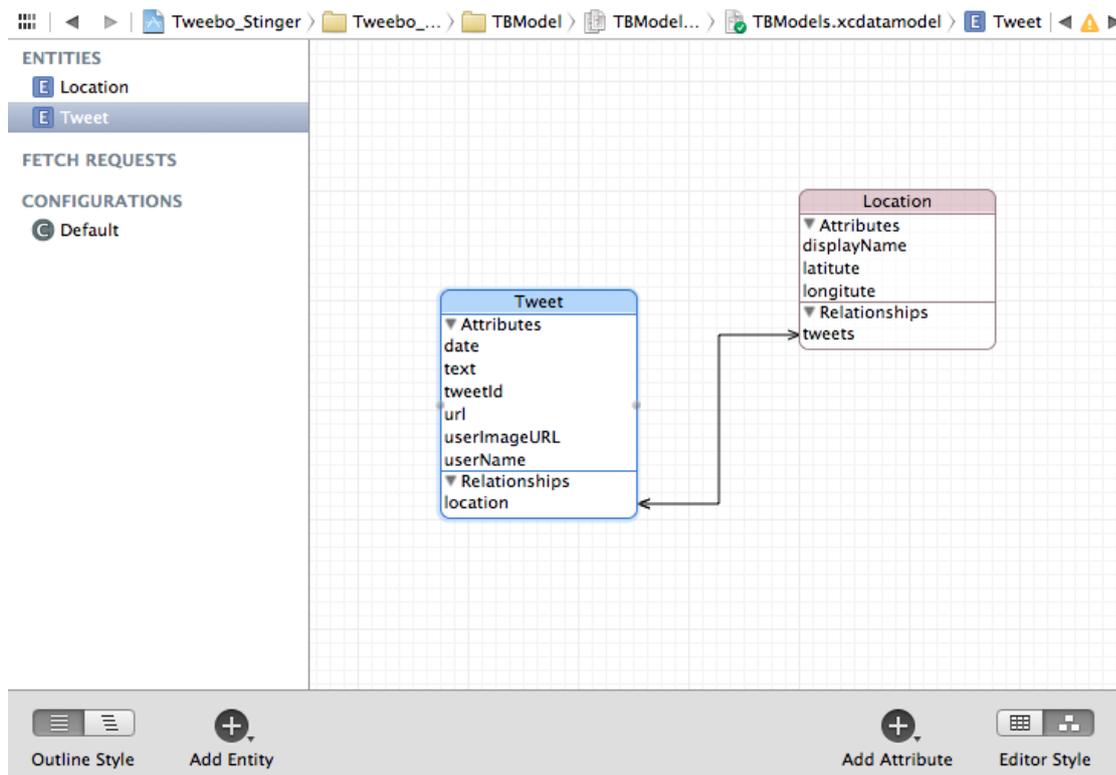
**Abbildung 3.1:** Core Data Editor: In der linken Spalte werden alle Entitäten aufgelistet. Wählt man eine aus, so können Attribute und Relationen zwischen Entitäten in der rechten Spalte erweitert und editiert werden. Auch das Hinzufügen von weiteren Entitäten ist einfach möglich.

jedoch ist es nicht möglich SQL Abfragen direkt zu schreiben oder mehrere Abfragen parallel durchzuführen.

Im Falle von Tweebo wird Core Data verwendet, um Tweets und Standorte (Locations) auf dem Gerät zu speichern. XCode erleichtert das Erstellen der Entitäten ungemein. Mit dem *XCDatamodel* Editor (siehe Abbildung 3.1) lassen sich einfach Entitäten definieren und in der Folge auch *NSManagedObjects* generieren. Mit dem *XCDatamodel* Schema Editor (siehe Abbildung 3.2) werden danach die entsprechenden Relationen definiert. Im Hintergrund arbeiten einige Klassen um den leichten Zugriff auf die SQLite Datenbank zu ermöglichen (siehe Abbildung 3.3, Quelle: Ralf Ebert<sup>9</sup>).

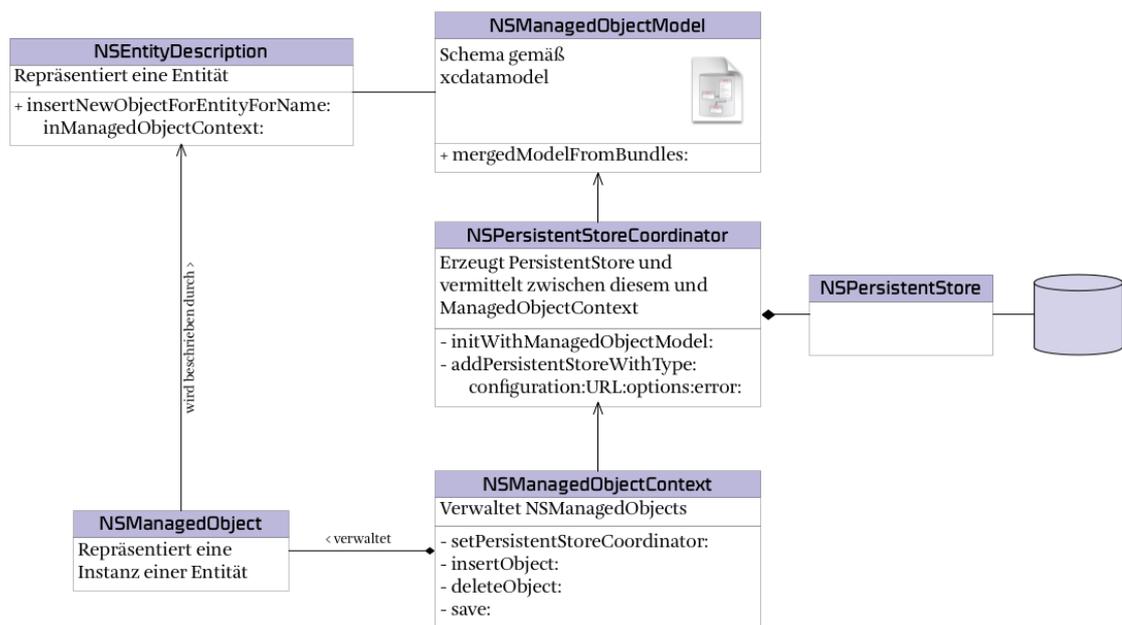
Mit wenigen Zeilen Code können Objekte abgespeichert werden (siehe Code-Ausschnitt 3.1),

<sup>9</sup><http://www.ralfebert.de/ios/coredata/> August, 2013



**Abbildung 3.2:** Core Data Schema Editor: Eine alternative Darstellung des Core Data Editors. Relationen können hier sehr gut abgebildet werden.

diese sechs Zeilen legen den Datensatz sicher in der SQLite Datenbank ab und auch die Abfrage von Objekten ist sehr einfach und übersichtlich gehalten (siehe Code-Ausschnitt 3.2). Core Data unterstützt bidirektionale 1:m und n:m Beziehungen zwischen Objekten und eine automatische Schema-Migration bei Änderungen am Schema. Jedoch hat sich in der Praxis herausgestellt, dass es sicherer ist die SQLite Datenbank nach jeder Änderung am Schema zu löschen um Inkonsistenzen zu vermeiden.



**Abbildung 3.3:** Core Data Klassen: Für die Verwendung von Core Data sind ein NSManagedObjectContext, ein NSPersistentStoreCoordinator und ein NSManagedObjectModel zu konfigurieren. Das NSManagedObjectModel lädt und repräsentiert die Modelldefinition zur Laufzeit. Der NSPersistentStoreCoordinator verwaltet die Persistierung der Daten und wird typischerweise mit einem SQLite-Persistent Store mit Ablage im Dokumentenverzeichnis der Anwendung konfiguriert. Alle Datenbankobjekte, die als NSManagedObject repräsentiert werden, werden zur Laufzeit in einem NSManagedObjectContext verwaltet

```

1 // Eine neue NSEntityDescription wird erstellt, welche in diesem Fall
  // die Entität "Location" und deren Attribute beschreibt.
2 NSEntityDescription *entityDesc = [NSEntityDescription entityForName:
  @"Location" inManagedObjectContext: context];
3 // Ein neues NSManagedObject wird mit Hilfe der oben erstellten
  // NSEntityDescription erstellt welches die Daten in der Datenbank
  // repräsentiert.
4 NSManagedObject *newLocation = [[NSManagedObject alloc]
  initWithEntity:entityDesc insertIntoManagedObjectContext:context]
5 // Nun werden die Attribute gesetzt, so bekommt beispielsweise das
  // Attribute "latitute" den Inhalt der Variable _lat.
6 [newLocation setValue:_lat forKey:@"latitute"];
7 [newLocation setValue:_lon forKey:@"longitute"];
8 [newLocation setValue:_displayName forKey:@"displayName"];
9 NSError *error;
10 // Das neu erstellte NSManagedObject wird persistiert
11 [context save:&error];

```

**Code-Ausschnitt 3.1:** Core Data - Objekt speichern

```

1 // Eine neue NSEntityDescription wird erstellt, welche in diesem Fall
    die Entitaet "Location" und deren Attribute beschreibt.
2 NSEntityDescription *entityDesc = [NSEntityDescription entityForName:
    @"Location" inManagedObjectContext: contextt];
3 // Ein neuer Request wird mit Hilfe der oben erstellten
    NSEntityDescription erstellt.
4 NSFetchedRequest *request = [[NSFetchedRequest alloc] init];
5 [request setEntity:entityDesc];
6 // Als Praedikat wird "displayName like (Inhalt der Variable
    _displayName)" mitgegeben.
7 NSPredicate *predicate = [NSPredicate predicateWithFormat:@"
    displayName like %@", _displayName];
8 [request setPredicate:predicate];
9 NSError *error;
10 // Nun wird nach Resultaten gesucht. Dies liefert einen Array mit den
    entsprechenden Ergebnissen.
11 NSArray *matchingData = [contextt executeFetchRequest:request error:&
    error];

```

**Code-Ausschnitt 3.2:** Core Data - Abfrage

## Whirly Globe

Whirly Globe [3] ist das Herzstück von Tweebo. Nach einigen Versuchen selbst einen 3D Globus zu implementieren wurde dieses durchdachte Open Source - Framework entdeckt und in Tweebo eingebaut. Whirly Globe ist eine eigenständige interaktive 3D View, welche leicht in iOS Apps integriert werden kann. Der Globus wird mit Hilfe von OpenGL ES<sup>10</sup> gerendert. Die Textur des Globus ist austauschbar und besteht in unserem Fall aus OpenStreetMap<sup>11</sup> Karten Kacheln, sogenannten Tiles, welche dynamisch und je nach Bedarf über ein Webservice geladen werden. Der Globus unterstützt das Zeichnen von Vektoren, Bildern, Labels, Markern und vielen weiteren visuellen Instrumenten um Daten oder Logik auf der 3D View abzubilden. Das Framework ist relativ schnell und flexibel, wodurch der Entwickler meist mit einer konstanten Framerate rechnen kann, und unterstützt MultiTouch - Gesten. Whirly Globe ist daher das perfekte Werkzeug um schnell geografische Positionen auf einem 3D Globus anzuzeigen, da auch Längen- und Breitengrade schnell berechnet werden können.

Sobald dieses Objective-C Framework in dem vorhandenen XCode-Project integriert

<sup>10</sup><http://www.khronos.org/opengles/>

<sup>11</sup><http://www.openstreetmap.org/>

wurde, kann der Entwickler sehr schnell eine Abbildung unseres Heimatplaneten erstellen (siehe Code-Ausschnitt 3.3), welche durchaus ansprechende Resultate liefert (siehe Abbildung 3.4).



**Abbildung 3.4:** WhirlyGlobeViewController

```

1 // Ein neuer WhirlyGlobeViewController wird initialisiert. Dieser ist
   fuer die Anzeige des 3D Globus hauptverantwortlich
2 globeViewC = [[WhirlyGlobeViewController alloc] init];
3 // Der ViewController, welcher den WhirlyGlobe erstellt wird als
   Delegate mitgegeben, sodass auf entsprechende Calls reagiert
   werden kann.
4 globeViewC.delegate = self;
5 // Auf die aktuelle View wird nun der WhirlyGlobe gezeichnet.
6 [self.view addSubview:globeViewC.view];
7 // Die Abmessungen der WhirlyGlobe-View werden gesetzt.
8 globeViewC.view.frame = self.view.bounds;
9 [self addChildViewController:globeViewC];
10
11 MaplyQuadEarthWithRemoteTiles *layer = [[
   MaplyQuadEarthWithRemoteTiles alloc] initWithBaseURL:@"http://a.
   tiles.mapbox.com/v3/mousebird.map-2ebn78d1/" ext:@"png" minZoom:0
   maxZoom:12];
12 [globeViewC addLayer:layer];

```

**Code-Ausschnitt 3.3:** Whirly Globe - Initialisierung

## AFNetworking

AFNetworking [20] ist wohl die bekannteste iOS Open Source Library um Netzwerkauf-rufe abzuarbeiten. Sie wurde von Matt Thompson entwickelt und verwendet viele mo-dere Bestandteile von Objective-C, wie zum Beispiel Blocks<sup>12</sup>. Blocks machen zwar den geschriebene Code nicht unbedingt lesbarer, sind aber ein mächtiges Werkzeug um beispielsweise asynchrone Netzwerkauf-rufe zu tätigen. AFNetworking deckt die wich-tigsten Funktionalitäten im Zusammenhang mit Netzwerkaufrufen ab. Einige wenige davon sind einfache Get- oder Post- Aufrufe / Abfragen, Sessionmanagement, Down-loads, Serialisation oder auch einfach das Überprüfen der Erreichbarkeit im Netzwerk (siehe Code-Ausschnitt 3.4).

<sup>12</sup>[https://developer.apple.com/library/ios/DOCUMENTATION/Cocoa/Conceptual/Blocks/Articles/00\\_Introduction.html](https://developer.apple.com/library/ios/DOCUMENTATION/Cocoa/Conceptual/Blocks/Articles/00_Introduction.html)

```

1 AFHTTPRequestOperationManager *manager = [
    AFHTTPRequestOperationManager manager];
2 // Mit dem zuvor erstellten AFHTTPRequestOperationManager wird eine
    GET-Abfrage erstellt. Die entsprechende URL wird mitgegeben.
3 [manager GET:@"http://example.com/resources.json" parameters:nil
    success:^(AFHTTPRequestOperation *operation, id responseObject) {
4     // Ist der Aufruf erfolgreich, so wird das gefundene JSON-Objekt
        in der Konsole ausgegeben.
5     NSLog(@"JSON: %@", responseObject);
6 } failure:^(AFHTTPRequestOperation *operation, NSError *error) {
7     // Schlaegt der Aufruf fehl, so wird der entsprechende Fehler in
        der Konsole ausgegeben.
8     NSLog(@"Error: %@", error);
9 }]];

```

**Code-Ausschnitt 3.4:** AFNetworking - Get-Request

## UzysSpringBoardView

Die UzysSpringBoardView [12] eine Komponente welche einer sogenannten GridView sehr ähnelt, auch vergleichbar mit dem HomeScreen von iOS Geräten. Die einzelnen Zellen sind also in einem Raster angeordnet und es besteht die Möglichkeit zu blättern. Sie wird verwendet um Tweets einer geografischen Lage anzuzeigen und unterstützt das Hinzufügen, Umsortieren und Löschen von einzelnen Zellen, welche im Falle von Tweebo einzelne Tweets darstellen. Die einzelnen Zellen der GridView lassen sich, ähnlich wie iOS-Entwickler es auch von ListViews kennen, je nach Belieben visuell anpassen.

## OpenStreetMap

Die Integration der OpenStreetMap-Schnittstelle [17] bildet weiters einen Grundstein der Tweebo Applikation. Wie zuvor erwähnt setzt sich ein Tweet-Objekt aus unterschiedlichen Komponenten zusammen. Eine dieser Komponenten ist die Lokalisierungsinformation oder weiters auch bezeichnet als das *Positionierungselement*, welche den Ursprung der Tweetnachricht repräsentiert.

Der eigentliche Informationsgehalt dieses Datenelements besteht meist aus einem Namensstring, welcher weiters auf die Position zurückzuführen ist. Werte wie London, Chicago oder auch Wien sind mögliche Inhalte.

Weiters gilt es nun die stringbasierende Lokalisierungsinformation in entsprechende

Breiten- und Längengradinformationen zu übersetzen um ein Mapping auf den Globus zu ermöglichen.

OpenStreetMap und dessen frei zugängliche Abfrage-Schnittstelle [18] ermöglichen es mittels einer Parameterübergabe die gesuchten Längen- und Breitengradinformation abzufragen. Tweebo parst das weitere Abfrageresultat (JSON) und übergibt die entsprechenden Koordinaten für ein Tweetobjekt dem *Applikationmanager*, welcher weiters das Tweetobjekt und dessen Loklisierungsinformationen persistiert (CoreData) und später das Tweetobjekt auf dem Globus positionieren zu können.

### 3.3 Tweebo Funktionalität

#### Datenverarbeitung

Um auf die Twitter API zuzugreifen benötigt Tweebo zu aller erst einen Twitter Login. Dieser wird seit iOS 6 direkt vom Betriebssystem zur Verfügung gestellt. User haben also die Möglichkeit den eigenen Twitter Account global zu hinterlegen und müssen der jeweilige App dann nur noch den Zugriff gewähren. Da Tweebo für iOS 6 und der vor kurzem erschienen Version iOS 7 entwickelt wurde, wird diese Funktionalität natürlich unterstützt.

Werden die Daten des Twitter Accounts freigegeben, greift Tweebo zu Beginn auf die Twitter-Timeline<sup>13</sup> des Users zu. Die Twitter Timeline beinhaltet die neuesten Tweets von anderen Twitter Useren, welchen gefolgt wird, und spiegelt so die interessanten Beiträge für den jeweiligen User wieder. Natürlich gibt es während dieser API-Abfrage eine visuelle Rückmeldung. Während sämtlichen API-Aufrufen und Datenbankzugriffen, welche alle im Hintergrund ablaufen, wird die Benutzeroberfläche nicht geblockt und ist weiterhin uneingeschränkt benutzbar. Sind neue Daten vorhanden, wird die Benutzeroberfläche dementsprechend aktualisiert. Nach erfolgreichem Laden der Timeline-Tweets werden diese einerseits in der Datenbank mittels Core Data persistiert und gleichzeitig nach Standorten sortiert.

Einerseits beinhalten viele Tweets, wie in der Einleitung schon erwähnt, im besten Fall den genauen Längen- und Breitengrad. Diese Position spiegelt den Standort des Twitter - Users während der Erstellung des Tweets wieder und kann ohne weiters übernommen werden. Falls Tweets diese Information nicht beinhalten, verwendet Tweebo einen kleinen Trick um den Tweet doch noch einer bestimmten Position zuzuordnen zu können.

---

<sup>13</sup><https://support.twitter.com/articles/164083-what-is-a-timeline>

Wieder wird eine OpenStreetMap Schnittstelle<sup>14</sup> herangezogen, um aus dem Namen der Stadt oder dem Staat, welcher bei den meisten Tweets als Standort des Tweet-Autors mitgeliefert wird, die genauen Koordinaten zu bestimmen (siehe Code-Ausschnitt 3.5). Natürlich speichert Tweebo sich all diese Standort Daten auch lokal, um die Anzahl der Netzwerkaufrufe möglichst gering zu halten. Wurde ein Standort mit genau diesem Namen schon einmal gefunden, ist ein weiterer Netzwerkaufruf somit nicht mehr notwendig. Sobald alle Standorte bestimmt und Tweets dem jeweiligem Standort zugeordnet worden sind, werden diese auf der Benutzeroberfläche angezeigt.

```
1 // Gibt Koordinaten fuer einen entsprechenden String zurueck, welcher
   // eine Location repraesentiert. Asynchron, erstellt ein TBLocation
   // Objekt und gibt dieses im completionBlock zurueck, sollte eine
   // entsprechende Position gefunden werden.
2 + (void) getCoordinatesForString: (NSString *)locationSearchString
   completion: (void (^) (TBLocation *location, NSError *error))
   completionBlock; {
3     // Sollte der String nil sein oder keinen Text beinhalten wird
   // abgebrochen.
4     if (!locationSearchString || [locationSearchString length] == 0)
5         return;
6     // Mit Hilfe eines Beispiel Objekte wir mittels Core Data in der
   // lokalen Datenbank nach der Location gesucht.
7     TBLocation *location = [[TBLocation alloc] init];
8     location.displayName = locationSearchString;
9     if ([location getCoordinates]) {
10        completionBlock(location, nil);
11        return;
12    }
13    // Sollte die Suche in der lokalen Datenbank fehlgeschlagen sein
   // wird ueber http://nominatim.openstreetmap.org/ nach den
   // Koordinaten gesucht.
14    NSLog(@"getCoordinatesForString: %@", locationSearchString);
15    // Der gegeben String wird escaped und somit fuer den Request
   // vorbereitet.
16    NSString* escapedUrlString =
17    [locationSearchString stringByAddingPercentEscapesUsingEncoding:
18     NSASCIIStringEncoding];
19    // Der Request wird erstellt
20    NSString *urlString = [NSString stringWithFormat:@"%%%%",
```

<sup>14</sup><http://nominatim.openstreetmap.org/>

```

kOpenStreetMapBaseUrl, escapedUrlString,
kOpenStreetMapSearchParams];
21  NSURL *url = [NSURL URLWithString:urlString];
22  NSURLRequest *request = [NSURLRequest requestWithURL:url
23                          cachePolicy:
NSURLRequestReturnCacheDataElseLoad
24                          timeoutInterval:
kOpenStreetMapTimeout];
25  // Mit Hilfe von AFNetworking wird der Request asynchron
abgesetzt.
26  AFJSONRequestOperation *operation = [AFJSONRequestOperation
JSONRequestOperationWithRequest:request success:^(NSURLRequest *
req, NSHTTPURLResponse *response, id JSON) {
27      TBLocation *location;
28      // Im Falle einer Antwort wird ein TBLocation Objekt erstellt
29      if ([JSON isKindOfClass:[NSArray class]]) {
30          for (NSDictionary *locDict in JSON) {
31              location = [TBLocation instanceFromDictionary:locDict
];
32              location.displayName = locationSearchString;
33              break;
34          }
35          // Das TBLocation Objekt wird mittels Block
zurueckgegeben.
36          completionBlock(location, nil);
37      }
38  } failure:^(NSURLRequest *req, NSHTTPURLResponse *response,
NSError *error, id JSON) {
39      // Im Falle eines Fehlers wird dieser ebenfalls mittels Block
zurueckgegeben.
40      completionBlock(nil, error);
41  }];
42  [operation start];
43 }

```

**Code-Ausschnitt 3.5:** Standort wird ermittelt

## Interaktion

Wie schon in Kapitel 3.2 erwähnt ist das Herzstück von Tweebo ein dreidimensionaler Globus, welcher mit Kartenkacheln überzogen ist. Etliche Zoomstufen sind verfügbar,



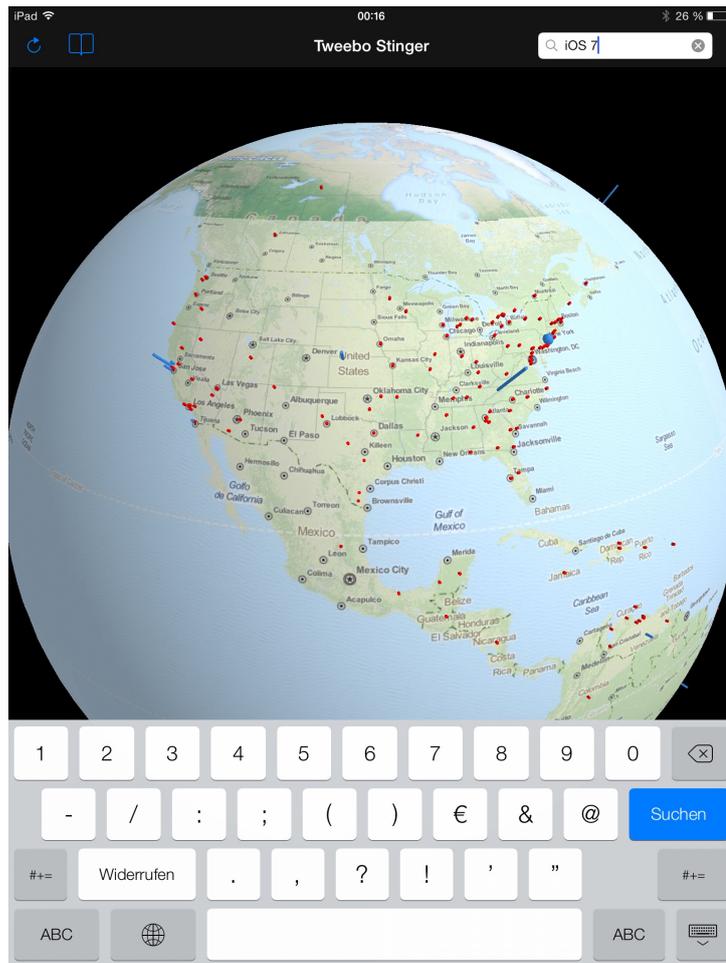
**Abbildung 3.5:** 3D Darstellung der Erde mit Tweets

die Abbildung unserer Erde kann einerseits als kleiner Ball dargestellt werden, andererseits kann so genau zu einem Teil der Erde herangezoomt werden, dass die kleinsten Gassen und deren Namen sichtbar werden. Nun werden Tweets auf dem Globus entweder in Form von Zylindern oder in Form von einer Halbkugel auf dem Globus angezeigt. Ist nur ein Tweet an einem bestimmten Standort verfügbar, so wird ein kleiner roter Zylinder eingeblendet. Sollten mehrere Tweets an einem Standort verfügbar sein, so wird ein blauer Zylinder angezeigt, wobei die Höhe des Zylinders der Anzahl Tweets der verfügbaren Tweets entspricht. Sind mehr als einhundert Tweets verfügbar, blendet Tweebo eine blaue Halbkugel über dem Standort ein (siehe Abbildung 3.5, sowie 4.2 bzw. 4.3).



**Abbildung 3.6:** Detailansicht für Wien

Der User hat die Möglichkeit zu einer bestimmten Position zu navigieren. Dies passiert mittels Touch und Multi-Touch Gesten. Der Globus kann intuitiv gedreht werden, wobei der Nordpol immer nach oben zeigt, um eine entsprechende Orientierung zu gewährleisten. Durch eine Pinch-Bewegung kann gezoomt werden. Hat sich der User für einen bestimmten Standort entschieden, kann er mit einem einfachen Touch auf den Globus die Detailansicht des Standorts aufrufen (siehe Abbildung 3.6). Wird auf einen Bereich ausgewählt, welcher keinen Standort besitzt, (z. B. die Mitte des Atlantiks) so werden weltweite Tweets angezeigt, welche keinem Standort zugeordnet werden konnten. Wie oben schon erwähnt werden die Tweets der jeweiligen Location in einer GridView (UzysSpringBoardView) angezeigt, welche aus mehreren Seiten bestehen kann. Durch einen



**Abbildung 3.7:** Suche

Swipe in die entsprechende Richtung kann die Seite gewechselt werden. Jeder Tweet wird durch eine Zelle in der GridView repräsentiert und beinhaltet das Bild des Tweet Autors, dessen Namen und den Text selbst. Wurde dem Tweet ein Link hinzugefügt, so kann mit einem Touch auf den Tweet der Link im iOS-Browser Safari geöffnet werden.

Natürlich sind für den Benutzer nicht nur die Tweets seiner eigenen Twitter Timeline interessant. Tweebo ist vor allem dazu gedacht, schnell auf dem gesamten Globus Trends und wichtige Ereignisse zu erkennen. Hierzu dient die Suche. Wird ein Suchbegriff eingegeben, werden die aktuellen Tweets der Timeline vorläufig ausgeblendet und der Globus mit Tweets aus dem Suchergebnis befüllt. Wird zum Beispiel nach „iOS 7“

gesucht siehe Abbildung 3.7), welches im Moment in aller Munde ist, so wird schnell sichtbar, wo auf der Welt dieser Begriff gerade am meisten diskutiert wird. Schnell kann der Benutzer durch die Ergebnisse browsen und bei Bedarf weitere Informationen über gegebenenfalls mitgelieferte Webseiten abrufen. Ist die Suche beendet und will der User wieder zu seinen Timeline Tweets zurückkehren, so kann er diese schnell über das Lesezeichen-Symbol rechts oben erreichen. Jederzeit können mit dem Aktualisierungssymbol, welches sich auch rechts oben befindet, die Daten aktualisiert und auf neue Tweets geprüft werden.



## Ergebnisse

In dem folgenden Kapitel möchten wir unsere grundlegenden Ergebnisse nach der finalen Implementierung und entsprechender Testphase präsentieren. Weiters werden wir in drei Use-Cases die primären Funktionen der Tweebo Applikation beschreiben und mit entsprechendem Bildmaterial untermauern.

Die eigentliche Suche nach Kurznachrichten auf Basis eines Volltextsuchkriteriums stellt einen der Kernaspekte der Applikation dar. Mittels der Twitter REST API wird es ermöglicht den Suchbegriff zu übermitteln und entsprechende Resultate als Antwort zurückzuerhalten. Die Suche nach Twitter-Nachrichten in ihrem derzeitigen Umsetzungsstatus ermöglicht eine performante Analyse von Themen und Begriffen. Intuitiv und schnell kann nach einem speziellen Suchbegriff gesucht werden.

Es ist sehr interessant zu beobachten, dass ein Begriff, welcher weltweit in aller Munde ist, auch in den entsprechenden Regionen mit ausreichend Tweets repräsentiert wird. Sensationelle und Aufmerksamkeit erregende Themen verbreiten sich fast so schnell wie ein Lauffeuer über alle Twitterkanäle. Schnell wird ersichtlich, in welchen Regionen der Erde und unserer Gesellschaft Twitter häufig benutzt wird und als ein aktives Kommunikationsmedium Verwendung findet.

Zu diesen Regionen gehören vor allem die Vereinigten Staaten von Amerika, Europa, Brasilien, Süd Afrika, Indien, Japan, Indonesien, Teile Chinas und die großen Städte Australiens. Wir können entsprechende Parallelen ziehen, dass die kontinuierliche Verbreitung von mobilen Endgeräten eine schnelle und fast schon uneingeschränkte Nutzung und nahtlose Integration an den Puls der Zeit ermöglicht, welcher in gewisser Art und Weise über Twitter reflektiert wird.

Binnen weniger Sekunden wird es dem Benutzer ermöglicht einen Überblick zu erhalten hinsichtlich der Hotspot Regionen, welche den Begriff zur Zeit am häufigsten diskutieren.

## **4.1 Funktion 1 - Die Suche**

Die Suche steht dem Benutzer durchgehend zur Verfügung. Die Suchleiste wurde gezielt in die Navigationsleiste positioniert um einerseits dem Users fortwährend und ergonomisch die Suche einzelner Begriffe zu ermöglichen und andererseits dem User auch an gewohnte Paradigmen zu erinnern. Somit hat sich entsprechend in der Testphase gezeigt, dass Benutzer nicht viel umdenken mussten, da jenes Verhalten aus schon bekannten Umgebungen (z.B.: Browser) übernommen werden konnte. Diese Umsetzungsart wurde gewählt, um die Einarbeitungszeit und den eigentlichen Lernaufwand so gering wie möglich zu halten. Abbildung 4.1 gibt die Suchergebnisse einer Suche nach „iOS“ auf den verschiedenen Kontinenten wieder. Einerseits kann schnell erkannt werden, in welchen Teilen der Erde Twitter benutzt wird, andererseits kann sogar die Uhrzeit (18:00 UTC+1) an der Anzahl der Tweets erahnt werden, da die Dichte der Suchergebnisse auf den Kontinenten am größten ist, auf denen die meisten Menschen gerade munter sind.

## **4.2 Funktion 2 - Die Aktivitätsanzeige/Hotspots**

Neben der eigentlichen Suche stand für uns im Vordergrund das Twittergefühl und das daraus resultierende Geschehen aus der zwei-dimensionalen Welt zu heben und einen globalen Bezug zu erstellen. Hier bedienten wir uns eines entsprechenden Mappings zwischen den Standortmetadaten einer Twitternachricht und der Projektion auf einen drei-dimensionalen Globus.

Das entsprechende Ergebnis war, verblüffend. Eines unserer aussagekräftigsten Resultate ergab sich in der entsprechenden Testphase, in welcher wir die Applikation von begeisterten Twitter-Benutzern testen liesen. Es hat sich herausgestellt, dass sich durch das Mapping der Kurznachrichten auf den Globus das entsprechende Verständnis und die weitere Interpretationen einer eigentlichen Nachricht ein wenig verändert hat.

Der Informationsgehalt der Kurznachricht ist natürlich derselbe, aber dadurch, dass der Benutzer aktiv einen lokalen Bezug auf die Nachricht visualisiert bekommt, werden mehr Faktoren im Verständnisprozess wahrgenommen. Einflüsse wie Kultur, Sprache und Tageszeit werden aktiv Teil des Interpretationsprozesses. Wir konnten so manche

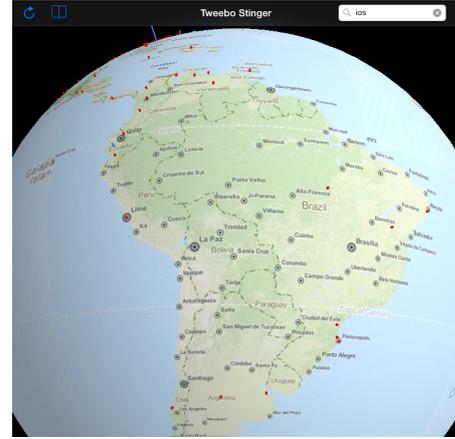
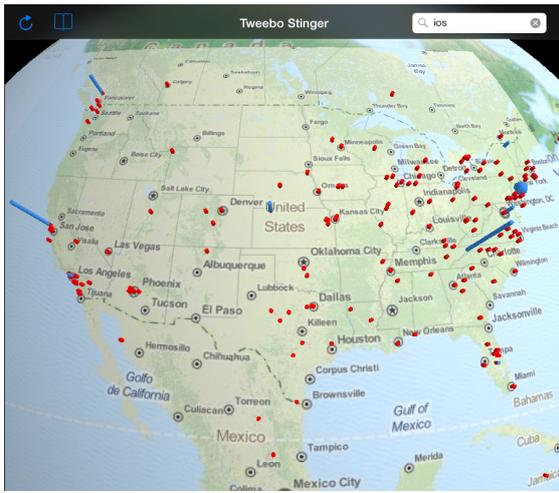


Abbildung 4.1: Suchergebnis „ios“ auf den verschiedenen Kontinenten

Äha-Momente aufzeigen, in welchen Benutzer erstaunt feststellen mussten aus welcher Region der ursprüngliche Tweet verfasst wurde.

Neben der eigentlichen Visualisierung der Kurznachrichten auf dem drei-dimensionalen Globus wurde noch eine Clustering-Mechanismus hinzugefügt. Dieser Clustering-Mechanismus ermöglicht es entsprechende Hotspots aufzuzeigen. Ein Hotspot definiert einen Standort in welchem die Twitteraktivität ungewöhnlich hoch ist.

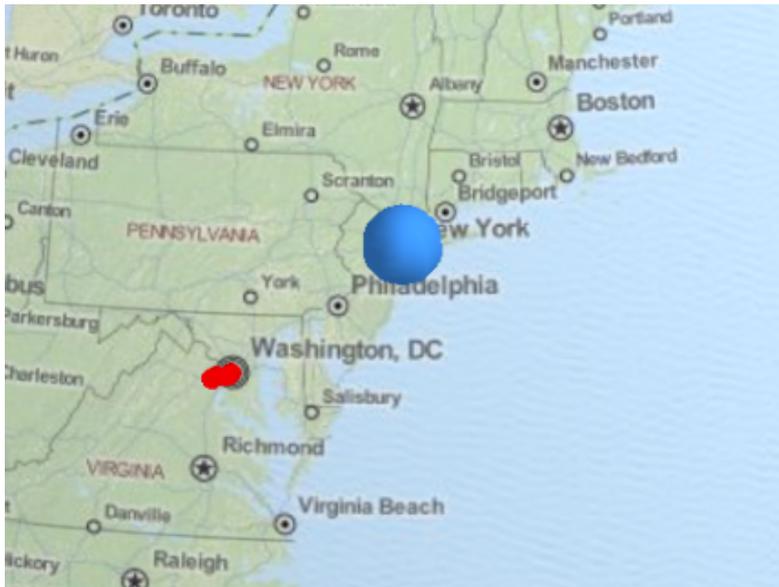
In der derzeitigen Implementierung werden Hotspots durch einen Zylinder auf dem Globus visualisiert (siehe Abbildung 4.2). Die Höhe eines Zylinder repräsentiert weiters den Aktivitätsgrad beziehungsweise wird eine Halbkugel angezeigt, sobald mehr als hundert Tweets auf einem Hotspot existieren (siehe Abbildung 4.3). Hier hatten wir das bestätigende Ergebnis, dass globale Metropolen und dichte Ballungsgebiete dediziert durchschnittlich die größte Menge an Kurznachrichten produzieren.



**Abbildung 4.2:** Clustering: Die verschiedenen Zylindergrößen repräsentieren die unterschiedliche Anzahl von Tweets an den jeweiligen Hotspots. Wird nur ein Tweet gefunden wird der Zylinder zusätzlich rot eingefärbt um die Sichtbarkeit zu erhöhen.

### 4.3 Funktion 3 - Message browsing

Der dritte Kernaspekt der Tweebo Applikation stellt die eigentliche Visualisierung der Kurznachrichten in Bezug auf die zuvor erwähnten Hotspots dar. Tweebo ermöglicht auf gewisse Art und Weise standortbezogen benutzerorientierte Kurznachrichtenstreams zu verarbeiten. Hier lag uns speziell im Vordergrund eine einfache und elegante Methodik



**Abbildung 4.3:** Clustering: Sobald mehr als einhundert Tweets zu einem Hotspot gefunden werden, wird eine blaue Halbkugel angezeigt.

einzuführen dies zu ermöglichen. Unterschiedliche Darstellungsebene wie Listenelemente ermöglichen die gezielte Darstellungen von systematischen Datensätzen.

In weiterer Folge mussten dediziert den mobilen Gebrauchsaspekt berücksichtigen und motivieren. Entsprechende Diskussions- und Testergebnisse, welche entsprechend aus dem User Driven Development Konzept motiviert wurden, ergaben in der Konzeptionsphase, dass die Kombination eines gut strukturierten Rasters in Verbindung eines mobil üblichen Paging-Mechanismus zu einem ergonomischen und ansprechenden Resultat führt. Zusätzlich bildet es den technischen Vorteil, dass nicht zu viele Elemente (kompletter Twitterstream) zeitgleich erstellt und dargestellt werden müssen (siehe Abbildung 3.6).

## 4.4 Einschränkungen

In dem folgenden Kapitel möchten wir weiters auch auf ein paar Einschränkungen hinweisen, welche sich im Zuge der entsprechenden Entwicklungstätigen aufgezeigt haben. Auf die wichtigsten drei wollen wir hier kurz eingehen.

## **Genauer geografischer Standort**

Nur sehr selten wird der genaue geografische Standort mitgeliefert. Viel zu oft ist Tweepo darauf angewiesen, den Längen- und Breitengrad des Tweets allein nur durch den Namen des mitgelieferten Heimatortes des Tweet Autors zu bestimmen. Diese Information ist natürlich oft nicht richtig. Einerseits werden falsche Positionen erzeugt, da Benutzer jedes beliebige Wort als Heimatort angeben können, andererseits ist nicht garantiert, dass sich der Benutzer auch wirklich auf diesem Ort zur Zeit der Erstellung des Tweets aufhält. Dies sind Probleme, welche in erster Linie mit Datenschutz zu tun haben. Jeder Tweet Autor muss seine genaue Position aktiv freigeben um diese dem Tweet beizufügen. Ob dieses Problem in Zukunft gelöst wird, weil User aufgeschlossener werden und die Privatsphäre im Internet weniger wichtig wird oder ob wir in ein paar Jahren noch immer mit ähnlichen Problemen zu kämpfen haben ist fraglich.

## **Verschiedenen Sprachen**

Das Problem der unterschiedlichen Sprachen ist zwar weiterhin ein großes Thema (siehe Abbildung 4.4), ist aber, wenn man es mit dem Problem der unzureichenden Standortinformation vergleicht, geradezu lächerlich. Leider war aber im Projektzeitraum keine Zeit um dieses Feature zu berücksichtigen. Es existieren einige Webservices welche es möglich machen, zum Beispiel Suchbegriffe und auch wieder die Texte der Resultate in mehrere Sprachen zu übersetzen. Allerdings werden diese Services sehr schnell kostenpflichtig und Tweepo wurde von Anfang an nicht als kommerzielle App geplant. Schön wäre es trotzdem zumindest in den fünf bis sechs verbreitetsten Sprachen suchen zu können und dann pro Sprache 1500 Suchergebnisse von der Twitter API zu erhalten, welche wieder in die Ausgangssprache zurückübersetzt werden könnten. Viel genauere Suchergebnisse mit sicher sehr wertvoller Information wären dadurch möglich. Leider ist dieses Feature aber nicht im Projektrahmen und wird wenn, dann mit einem Update nachgereicht.

## **Begrenzte Anzahl an Tweets**

Jede Suchanfrage liefert maximal 1500 Ergebnisse, besteht dann aber auch aus 15 einzelnen Requests. Zu viele Requests sollte Tweepo nicht absetzen. Eine gewisse Anzahl an Tweets, um eine möglichst akkurate Darstellung der Verteilung des Themas auf dem Gobus zu sehen ist aber notwendig. Hier wurde ein entsprechendes Verhält-



Abbildung 4.4: Fremdsprachen

nis gewählt, welches beide Faktoren berücksichtigt. Allerdings wäre es natürlich ideal mit nur einem Request möglichst viele Ergebnisse zu erhalten. Twitter lässt dies allerdings nicht zu. Um dieses Problem ein wenig abzuschwächen wurde unter anderem der Aktualisieren-Button hinzugefügt. Wird die selbe Suche erneut gestartet, oder der Aktualisieren-Button geklickt, so werden die bestehenden Resultate der vergangenen

Suche nicht gelöscht, sondern Tweets, welche mit der erneuten Anfrage hinzugekommen sind, zusätzlich angezeigt. So ist es theoretisch Möglich über einen längeren Zeitraum hinweg eine Suchabfrage durchzuführen um ein möglichst genaues Ergebnis zu erhalten.

## Schlussfolgerungen und ein Ausblick

Die Entwicklung und Konzeption der iOS Applikation namens Tweebo war ein sehr lehrreiches und interessantes Unterfangen. Das unterliegende Konzept hat uns die ungewöhnliche Möglichkeit geboten eine wertvolle Kombination aus reinen Nachrichtenaspekten und Geolokalisierungen zu kombinieren. Weiters hatten wir von Anfang an eine spezielle technologische Zielgruppe im Visier, den mobilen Endpoint. Wir erachten es als sehr wichtig, wenn nicht sogar als verpflichtend, die weitere intensive Entwicklung und Ausschöpfung der Möglichkeiten von mobilen Endgeräten voranzutreiben.

Diesbezüglich muss natürlich erwähnt werden, dass die eigentliche Plattform, ob es jetzt Google Android oder Apple iOS ist, als zweitrangig zu erachten ist. Apple's iOS und die entsprechende Programmiersprache Objective-C haben sich als sehr hochwertige, fortgeschrittene und als sehr gut dokumentierte Entwicklungsumgebung präsentiert.

Die immer weiter aufstrebende Verknüpfung, Verarbeitung und Bereitstellung unterschiedlichster Daten wird sich unserer Meinung nach immer mehr auf der mobilen Ebene abspielen. Diese Ansicht und Einstellung bildet auch die grundlegende Motivation der Umsetzung dieses Projekts. Wie anfangs erläutert, befassen sich schon mehrere Plattformen mit der Thematik Twiternachrichten auszuwerten, und eine weitere Geolokalisierungskomponente hinzuzufügen. Doch keine dieser Plattformen kombiniert den mobilen und persönlichen Charakter. Tweebo schließt diese Lücke und ermöglicht es dem User seine Interessensgebiete visuell und global aufbereitet zu bekommen.

Es war überaus interessant zu beobachten, dass der grundlegende Entwicklungs- und

Konzeptionsprozess sich primär an die gewohnten Designkriterien und Paradigmen orientiert. Diese Behauptung trifft dediziert auf die Backendfunktionalitäten der Applikation und der genutzten Schnittstellen (OpenStreetMap, CoreData, etc.) zu.

Mehrere Iterationen waren nötig bis das schlussendliche Designkonzept ausgearbeitet war. Wichtig war, dass der User sich schnell in der App zurechtfindet, und immer die wichtigsten Informationen bereitgestellt bekommt. Unterschiedliche Motivationsstufen haben uns dazu verleitet, den Globus als das primäre Navigationselement der Tweebo Applikation heranzuziehen. Uns ist bewusst, dass die Auswahl eines 3D-Elements als primäres Navigationselement durchaus nicht immer ideal ist. Doch in diesem Fall haben wir uns erlaubt eine Ausnahme zu machen. Wir möchten dediziert das globale Bewusstsein im User erwecken und ihn dazu motivieren weltweit herumzustoßern. Der Globus ermöglicht es das Einarbeitungslevel hinsichtlich der Nutzung der Applikation auf ein Minimum zu reduzieren und das Vertrauensgefühl von Anfang an auf ein Maximum zu heben, da einem Großteil der User der Globus und dessen Bedeutung und Konzept vertraut ist. Weiters erhält der User das visuelle und globale Feedback, dass seine Twitertätigkeiten einem globalen Netzwerk ähneln und andererseits können wir auch die Neugier eines Users erwecken und ihn motivieren in neue und unerforschte Gebiete vorzudringen.

## **Ausblick**

Wir erachten die Tweebo Applikation in ihrem jetzigen Zustand als einen dedizierten Release-Candidate. Doch müssen wir auch erwähnen, dass ein zusätzliches Featureset angedacht und angestrebt wird. Uns würde es entsprechend reizen weitere Aspekte in Richtung des Gamification-Aspekt zu inkludieren, um die Motivationskurve der User weiters aufrecht zu halten. Zusätzlich ist es unser Ziel das Tweebo-Projekt mit weiteren Twitterfunktionalitäten auszustatten, um zusätzliche Features der Twitter API zu nutzen. Die Erschaffung einer Tweebo-Global-Community und deren Kollaboration oder die Einführung es globalen Domination-Features ala Ingress [16] wäre sehr interessant. Weiters können wir uns vorstellen den Globus zu adaptieren, sodass unterschiedliche Darstellungsmodelle unterstützt werden. Die Tweebo Applikation birgt sehr aussichtsreiches Potential in sich, welches wir in weiteren Releases preisgeben möchten.

# Literaturverzeichnis

- [1] Paul Earle, Daniel Bowden, and Michelle Guy. Twitter earthquake detection: earthquake monitoring in a social world. *Annals of Geophysics*, 54(6), 2012.
- [2] Hideyuki Fujita. Geo-tagged Twitter collection and visualization system. *Cartography and Geographic Information Science*, 40(3):183–191, 2013.
- [3] Steve Gifford. WhirlyGlobe Component, August 2013. <http://mousebird.github.io/WhirlyGlobe/>.
- [4] Apple Inc. Apple Inc, August 2013. <http://www.apple.com/>.
- [5] Apple Inc. CoreData, August 2013. <https://developer.apple.com/library/ios/documentation/cocoa/conceptual/CoreData/cdProgrammingGuide.html>.
- [6] Apple Inc. iOS, August 2013. <https://developer.apple.com/technologies/ios/>.
- [7] Apple Inc. iOS Developer Program, August 2013. <https://developer.apple.com/programs/ios/>.
- [8] Apple Inc. XCode, August 2013. <https://developer.apple.com/xcode/>.
- [9] Hootsuite Media Inc. Hootsuite Für Twitter & Facebook (iOS App), August 2013. <https://itunes.apple.com/at/app/hootsuite-for-twitter/id341249709?mt=8>.
- [10] Twitter Inc. Twitter, August 2013. <https://twitter.com/>.
- [11] Twitter Inc. Twitter (iOS App), August 2013. <https://itunes.apple.com/at/app/twitter/id333903271?mt=8>.
- [12] Jaehoon Jung. UzysSpringBoardView, August 2013. <https://github.com/uzysjung/UzysSpringBoardView>.

- [13] Kalev H. Leetaru, Shaowen Wang, Guofeng Cao, Anand Padmanabhan, and Eric Shook. Mapping the global Twitter heartbeat: The geography of Twitter. *First Monday*, 18(5), May 2013. <http://firstmonday.org/ojs/index.php/fm/article/view/4366/3654>. Zugriff: 2013-08-18.
- [14] Adam Marcus, Michael S. Bernstein, Osama Badar, David R. Karger, Samuel Madden, and Robert C. Miller. Twitinfo: aggregating and visualizing microblogs for event exploration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 227–236, Vancouver, BC, Canada, May 2011. ACM.
- [15] Brett Meyer, Kevin Bryan, Yamara Santos, and Beomjin Kim. TwitterReporter: Breaking news detection and visualization through the geo-tagged Twitter network. In *Proceedings of the ISCA 26th International Conference on Computers and Their Applications*, CATA '11, pages 84–89, New Orleans, Louisiana, USA, March 2011. ISCA.
- [16] NianticLabs. Ingress, August 2013. <http://www.ingress.com>.
- [17] OpenStreetMap. OpenStreetMap, August 2013. <http://www.openstreetmap.org>.
- [18] OpenStreetMap. OpenStreetMapApi, August 2013. <http://wiki.openstreetmap.org/wiki/API>.
- [19] Tapbots. Tweetbot - Der Twitter Client mit Charme für das iPhone (iOS App), August 2013. <https://itunes.apple.com/at/app/tweetbot-der-twitter-client/id428851691?mt=8>.
- [20] Matt Thompson. AFNetworking, August 2013. <https://github.com/AFNetworking/AFNetworking>.
- [21] Jan Vosecky, Di Jiang, and Wilfred Ng. Limosa: a system for geographic user interest analysis in Twitter. In *Proceedings of the 16th International Conference on Extending Database Technology*, EDBT '13, pages 709–712, Genoa, Italy, March 2013. ACM.
- [22] Jeremy J. D. White and Robert E. Roth. TwitterHitter: Geovisual analytics for harvesting insight from volunteered geographic information. In *Extended Abstracts*

*Volume of the 6th International Conference on Geographic Information Science,*  
GIScience '10, Zurich, Switzerland, September 2010. GIScience.