Bachelor MKI

Mobile Computing

SS 2020

Prof. Dr. Natividad Martínez Madrid

- Seminar Documentation -

Geolocation and maps integration

Fanni Tamara Marosi, 764345

[Fanni\_Tamara.Marosi@Student.Reutlingen-University.De](mailto:Fanni_Tamara.Marosi@Student.Reutlingen-University.De)

Ebru Selin Özcelik, 764349

[Ebru\_Selin.Oezcelik@Student.Reutlingen-University.De](mailto:Ebru_Selin.Oezcelik@Student.Reutlingen-University.De)  
  
  
6. Semester

Vorgelegt am: 24.06.2020

Inhaltsverzeichnis

[1. Motivation (Ebru) 3](#_Toc43477393)

[2. Grundlagen 3](#_Toc43477394)

[2.1 GPS 3](#_Toc43477395)

[2.2 Funktionsweise 3](#_Toc43477396)

[3. Mobile 3](#_Toc43477397)

[3.1 Zugriff 3](#_Toc43477398)

[3.1.1 GPS 4](#_Toc43477399)

[3.1.2 WLAN-Information 4](#_Toc43477400)

[3.1.3 Cell Tower / Mobilfunkmasten 4](#_Toc43477401)

[3.1.4 (Bluetooth) 4](#_Toc43477402)

[3.2 Geokoordinaten 4](#_Toc43477403)

[3.2.1 Sexagesimale und dezimale Darstellung 4](#_Toc43477404)

[3.2.2 Das Location-Objekt 4](#_Toc43477405)

[4. Android: LocationManager vs Google Play Services (Fanni) 5](#_Toc43477406)

[4.1 Location Manager 5](#_Toc43477407)

[4.1.1 Verfügbarkeit feststellen 5](#_Toc43477408)

[4.1.2 Daten empfangen 5](#_Toc43477409)

[4.1.3 Empfänger abmelden 5](#_Toc43477410)

[4. 2FusedLocationProviderAPI – Google Play 5](#_Toc43477411)

[4.2.1 FusedLocationProviderClient 5](#_Toc43477412)

[4.2.2 LocationRequest 5](#_Toc43477413)

[5. Maps SDK für Android 5](#_Toc43477414)

[6. Eine GPS-Tracker-App 6](#_Toc43477415)

[7. Summary (Ebru & Fanni) 6](#_Toc43477416)

[Quellenverzeichnis 7](#_Toc43477417)

# 1. Motivation (Ebru)

Wir haben uns für das Thema Geolocation interessiert, da es sich bei unserer Application hauptsächlich um eine Anwendung für die Nachbarschaft handelt. In der ersten Version ist geplant, dass der Nutzer bei der Registrierung als Wohnort die Stadt und Postleitzahl angibt. Für eine spätere Version haben wir überlegt, eine Positionsbestimmung über GPS-Daten einzubauen. Der Use Case würde so umgesetzt werden, dass die Nutzer bei der Suche nach einer Anzeige von ihrem Wohnort ausgehend einen bestimmten Radius einstellen können, in dem gesucht werden soll. Hierfür würde es sich anbieten, die Positionsbestimmung und Map Integration einzubauen.

# 2. Grundlagen

In diesem Kapitel werden die Grundlagen von dem Global Positioning System (GPS) vorgestellt. Hierbei handelt es sich um ein globales System, bestehend aus Satelliten, welches dazu dient, die globale Position zu bestimmen [1, S.666].

## 2.1 GPS

Capderou schreibt in dem Buch „Handbook of Satellite Orbits: From Kepler to GPS”, dass das amerikanische Verteidigungsministerium das System, welches als Navigation Satellite Time and Ranging / Global Positioning System (Navstar/GPS) bekannt wurde, entwickelte. Mithilfe dieses Systems sollte die globale Bestimmung der Position umstrukturiert werden.

Erste Studien haben bereits seit den 1960er Jahren stattgefunden. Seit 1978 gibt es die ersten funktionsfähigen Satelliten für die Bereiche Forschung und Entwicklung [1, S. 666].

## 2.2 Funktionsweise

Das GPS-System besteht aus 24 Satelliten, welche die Erdumlaufbahn umkreisen. Die Satelliten haben eine Entfernung von 20.000 Kilometer zur Erde. Diese Entfernung wird vom Meeresspiegel gemessen. Alle Satelliten enthalten eine präzise Uhr. Alle Uhren sind miteinander synchronisiert [2, S.23].

Eine Voraussetzung für das Funktionieren des Systems ist, dass an jedem Ort auf der Erde jederzeit die Daten von vier Satelliten empfangen werden können sollten. Durch die Konstellation der Satelliten wird dies, wie Tobias Schüttler in „Satellitennavigation. Wie sie funktioniert und wie sie unseren Alltag beeinflusst“ erläutert, gewährleistet [3, S.46]. Die Position des Empfängers wird relativ zu dessen Entfernung zu den Satelliten ermittelt. Anhand der Position mehrerer (mind. vier) Satelliten lässt sich die Position des Empfängers ableiten [3, S.2 ff.].

Sie versenden periodisch Signale über ihren eigenen Standort. Funktionsweise zusammenfassen

Mai 200 für alle gratis

Bodenstationen überwachen die Positionen der Satelliten und sorgen dafür, dass das System wie vorgesehen funktioniert. Hierfür ist es unter anderem notwendig, die Satellitenbahnen genau zu vermessen, die eingebauten Uhren zu überwachen und mögliche Störungen zu ermitteln [3, S.44].

# 3. Mobile

Einige Anwendungen für Smartphones erfordern, dass die Daten über den Standort des Nutzers erfragt werden können. In diesem Kapitel wird aufgezeigt, welche Möglichkeiten es gibt, die Standortdaten eines Nutzers zu erhalten.

## 3.1 Zugriff

In diesem Kapitel werden drei Möglichkeiten aufgezeigt, woher das Android SDK Standortdaten eines Nutzers beziehen kann. Zu diesen drei Möglichkeiten zählen GPS, WLAN-Netzwerke und Mobilfunkmasten.

### 3.1.1 GPS

Android bietet Programmierern die Möglichkeit die Lokalisierung eines Geräts über GPS oder Netzwerke vorzunehmen. Der große Vorteil der GPS-Variante besteht in dessen Genauigkeit. Jedoch lassen sich hier auch einige Nachteile feststellen. GPS funktioniert nicht immer zuverlässig in Gebäuden. Die Antwortzeit nach der ersten Anfrage kann länger dauern. Der Stromverbrauch des Geräts ist ziemlich hoch, wenn der GPS-Sensor aktiviert ist [4].

### 3.1.2 WLAN-Information

Bei der zweiten Variante wird die Position des Geräts über ein Netzwerk oder Funkzellen des Mobilfunknetzes bestimmt. Die Bestimmung der Koordinaten ist jedoch ungenauer als bei der GPS-Variante. Vor allem in flachen Gebieten, in denen die Funkzellen größer sind. Als ein Vorteil lässt sich aufführen, dass die Positionierung über Netzwerke innerhalb von Gebäuden möglich ist. Des Weiteren ist die Antwortzeit kürzer und der Akkuverbrauch nicht so hoch wie bei der GPS-Ortung [4].

### 3.1.3 Cell Tower / Mobilfunkmasten

### 3.1.4 (Bluetooth)

## 3.2 Geokoordinaten

### 3.2.1 Sexagesimale und dezimale Darstellung

Eine geografische Angabe der Position besteht aus einem Längen- und einem Breitengrad. Die Längengrade werden von 0 bis 180 Grad östlich oder westlich von Greenwich in England definiert. Die Breitengrade werden 0 bis 90 Grad nördlich oder südlich ausgehend vom Äquator definiert. Ein Grad besteht aus 60 Winkelminuten. Eine Winkelminute wiederrum aus 60 Winkelsekunden. Winkelminuten werden mit einem Apostroph, Winkelsekunden mit zwei Apostrophen abgekürzt.

Für die Darstellung der Position wird im Computer die Dezimalschreibweise genutzt, da diese für den Computer praktischer ist. In dem Location-Objekt, welches in Kapitel 3.2.2 erläutert wird, werden der Längen - und Breitengrad als double-Werte gespeichert.

Louis und Müller zeigen zu der Dezimalschreibweise ein Beispiel in dem Buch „Android“ auf. Für die Golden Gate Bridge wird der Breitengrad in der Dezimalschreibweise als 37.816667 und der Längengrad als -122.483333 definiert. Die Gradzahlen werden als Vorkommastellen dargestellt, die Nachkommastellen sind die Minuten und Sekunden. Anhand des Vorzeichens lässt sich bestimmen, um welche Richtung es sich handelt. Norden und Osten werden durch ein positives Vorzeichen, Süden und Westen durch ein negatives repräsentiert [4].

### 3.2.2 Das Location-Objekt

Bei dem Location-Objekt handelt es sich um eine Instanz der Klasse Location. Diese Klasse repräsentiert einen geografischen Standort. Das Android SDK stellt Programmierern diese Klasse in dem Paket android.location.Location zu Verfügung. Ein Location-Objekt kann Attribute wie einen Längen- und Breitengrad, Zeitstempel und die Angabe der Höhe besitzen.

Android garantiert, dass alle Location-Objekte, welche von einem LocationManager erzeugt werden, über einen gültigen Breiten- und Längengrad und Zeitstempel verfügen. Der LoactionManager wird in dem nachfolgenden Kapitel 4.1 vorgestellt.

Location-Objekte verfügen über einige public Methoden. Einige von ihnen werden in der nachfolgenden Tabelle 1 aufgezeigt. Die Auswahl erfolgte anhand der subjektiven Einschätzung, wie wichtig/häufig genutzt sie werden.

Tabelle 1: Public Methoden des Location-Objekts

|  |  |  |
| --- | --- | --- |
| **Methodenname/-signatur** | **Rückgabetyp** | **Beschreibung** |
| getAltitude() | double | Mithilfe dieser Methode lässt sich die Höhe (falls vorhanden) ermitteln. Gemessen wird sie in Metern in dem Referenzsystem „World Geodetic System 1984“ (WGS 84). |
| getLatitude() | double | Der Rückgabewert dieser Methode ist der Breitengrad des geografischen Standorts (gemessen in Grad). |
| getLongitude() | double | getLongitude() liefert den Längengrad des geografischen Standorts zurück. Dieser wird in Grad gemessen. |
| distaneTo(dest: Location!) | float | Diese Methode ermöglicht es, die ungefähre Entfernung zwischen dem Location-Objekt, auf welchem sie aufgerufen wird und dem als Parameter übergebenen Location-Objekt zu bestimmen. |
| getTime() | long | An dieser Stelle wird Zeitstempel (UTC) zurückgegeben. Es handelt sich um die Anzahl der Millisekunden, welche seit dem 01.01.1970 vergangen sind. |

<https://developer.android.com/reference/kotlin/android/location/Location>

# 4. [Android: LocationManager vs Google Play Services](https://stackoverflow.com/questions/33022662/android-locationmanager-vs-google-play-services) (Fanni)

Es werden zwei Möglichkeiten, zum einen die Klassen, das Android SDK anbietet und zum anderen Google Play Services näher betrachtet.

https://stackoverflow.com/questions/33022662/android-locationmanager-vs-google-play-services

## 4.1 Location Manager

Der LocationMangager ist eine Klasse im Android SDK. Mithilfe des LocationManagers bekommt man Zugang zu den Ortungsservices des Systems, auf welchem die Application ausgeführt wird. Die Application kann periodisch den Standort des Geräts abfragen oder eine Benachrichtigung erhalten, wenn der Nutzer sich in der Nähe eines angegebenen Standorts befindet.

### 4.1.1 Verfügbarkeit feststellen

Es gibt in Android bereits eine Klasse für die Bestimmung der Position des Geräts. Diese Klasse heißt GeolocationManager und befindet sich im android.hardware-Paket. Ein Objekt dieser Klasse kann vom Programmierer in einer Activity wie folgt initialisiert werden:

LocationManager locationManager = (LocationManager) this.getSystemService

### 4.1.2 Daten empfangen

### 4.1.3 Empfänger abmelden

## 4. 2FusedLocationProviderAPI – Google Play

https://developers.google.com/location-context/fused-location-provider

### 4.2.1 FusedLocationProviderClient

https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderClient

### 4.2.2 LocationRequest

<https://developers.google.com/android/reference/com/google/android/gms/location/LocationRequest>

# 5. Maps SDK für Android

<https://developers.google.com/maps/documentation/android-sdk/intro?hl=de>

# 6. Eine GPS-Tracker-App

Android Studio GPS location tracker tutorial 01: <https://www.youtube.com/watch?v=V62sxpyxapU>

# 7. Summary (Ebru & Fanni)

# Quellenverzeichnis

[ 1 ] Capderou, M.: Handbook of Satellite Orbits. From Kepler to GPS. Springer, 2014. DOI 10.1007/978-3-319-03416-4

[ 2 ] Doberstein, D.: Fundamentals of GPS Receivers. A Hardware Approach. Springer, 2012. DOI 10.1007/978-1-4614-0409-5

[ 3 ] Schüttler, T.: Satellitennavigation. Wie sie funktioniert und wie sie unseren Alltag beeinflusst. Springer, 2014. DOI 10.1007/978-3-642-53887-2

[ 4 ] Louis, D., Müller, L.: Android, Kapitel 17. Hanser, 2016. DOI 10.3139/978-3-446-45112-4