

Hausarbeit

1.1 Beschreibung

Die Hausarbeit wird in diesem Semester anstelle einer schriftlichen SEP durchgeführt. Untenstehende Aufgaben müssen Sie in 5 Stunden lösen. Bitte lesen Sie zunächst die ganze Aufgabenstellung und teilen sich die Zeit entsprechend ein.

Die Aufgaben unter 1.2 werden bei der Bewertung gleich gewichtet. Wesentlich dabei: Wurde die Aufgabe erfüllt? Ist der geforderte Funktionsumfang umgesetzt? Und mit geringerem Gewicht: Ist die App gut gestaltet und gut benutzbar? Ist der Code lesbar und gut strukturiert. Wie ist die Qualität der Dokumentation?

Ihr Resultat senden Sie als ZIP-Datei zurück (siehe Abgabe 1.3 unten).

1.2 Aufgabenstellung

Sie erhalten den Auftrag, eine Android App zu entwickeln, in der ein mobiler Benutzer offline nach den nächstgelegenen Restaurants suchen kann. Basis Ihrer App ist eine JSON Datei mit sämtlichen Restaurants aus der Schweiz (sowie einige aus den angrenzenden Nachbarländern), welche aus OpenStreetMap extrahiert wurde. Diese Datei finden Sie mitgeliefert.

Technologie: native Android-Applikation, implementiert in der Programmiersprache Kotlin.

1.2.1 Aufgabe 1

Erstellen Sie ein Wireframe für die Detailanzeige genau eines Restaurants. Sie können dazu ein Tool Ihrer Wahl verwenden. Auf folgender Webseite finden Sie die bei einem Restaurant erlaubten Tags:

<https://wiki.openstreetmap.org/wiki/DE:Tag:amenity%3Drestaurant>

Beachten Sie, dass in der Datenbasis nicht jedes Restaurant jedes Tag gesetzt hat. Die folgenden Tags werden, in absteigender Reihenfolge, am häufigsten verwendet:

`name, cuisine, addr:street, addr:housenumber, addr:postcode, website, addr:city, opening_hours, phone, wheelchair`

Zeigen Sie diese Tags auf Ihrer Detailanzeige an. Die Webseite soll sich von der Detailseite öffnen lassen. Die Öffnungszeiten sollen formatiert angezeigt werden.

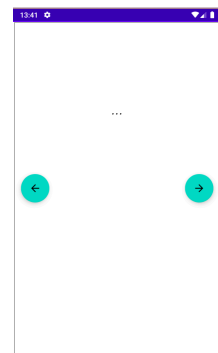
1.2.2 Aufgabe 2

Implementieren Sie Ihre Anzeigeseite des Wireframes als Startseite einer Android App. Laden Sie die mitgelieferte JSON-Datei am Start der App und zeigen Sie ein Restaurant Ihrer Wahl auf der Seite an. Hier einige gute IDs aus dem JSON, welche viele Attribute verwalten:

5926368521, 344919749, 618354424, 3625214583, 280528098

1.2.3 Aufgabe 3

Erweitern Sie die App durch zwei Floating Action Buttons (Vor und Zurück), mit denen durch die vorhandenen Daten geblättert werden kann. Die Buttons sollen rechts und links mittig erscheinen und ein Pfeil Icon enthalten. Beim Klick auf einen Button soll der nächste (vorherige) Datensatz angezeigt werden.



1.2.4 Aufgabe 4

Bauen Sie Ihre App „location aware“. Die App soll die aktuelle Position (latitude und longitude) lesen und beim Start direkt das nächstgelegene Restaurant anzeigen. Die Latitude und Longitude Werte jedes Restaurants sind im JSON hinterlegt. Die Entfernung können Sie über die „Manhattan Distanz“ berechnen.

Die Abfrage der aktuellen Location soll mit dem `FusedLocationProviderClient` implementiert werden. Dieser Google Play Dienst kann im Build-Gradle eingefügt werden:

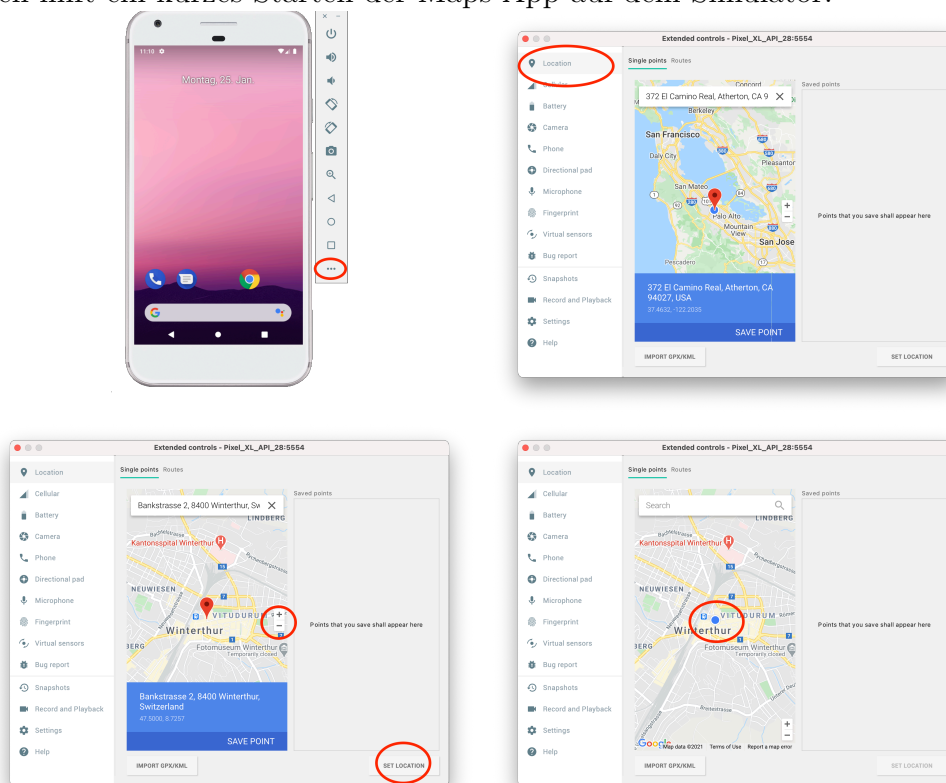
```
implementation
    'com.google.android.gms:play-services-location:17.1.0'
```

In der offiziellen Google Dokumentation wird die Abfrage beschrieben. Im Android Simulator ist der Ort standardmässig auf „Palo Alto“ gesetzt. Sie können diesen einfach auf Ihren aktuellen Standort ändern:

1. Öffnen Sie die Einstellungsseite des Simulators (drei Punkte, Bild 1 oben links).
2. Wechseln Sie auf *Locations* und verwenden Sie die Plus/Minus-Knöpfe um die Anzeige auf Ihren Standort zu verschieben (Bild 2 oben rechts).

3. Markieren Sie Ihren Standort und klicken Sie auf *Set Location* (Bild 3 unten links).
4. Der blaue Punkt sollte jetzt in der Karte an Ihrem Ort erscheinen (Bild 4 unten rechts).

Hinweis: Sollte für die Location immer noch „Palo Alto“ zurückgeliefert werden hilft ein kurzes Starten der Maps App auf dem Simulator.



1.2.5 Aufgabe 5

Erstellen Sie eine Kurzdokumentation als PDF. Diese sollte mindestens folgendes enthalten:

- Das Wireframe aus Aufgabe 1
- Zwei Screenshots mit zwei unterschiedlichen Restaurants in der Detailansicht
- Ein paar Angaben zur Implementierung (welche Aufgaben wurden gelöst / nicht gelöst, welche Schwierigkeiten gab es, ...)

1.3 Abgabe

Für die Abgabe der Prüfung erstellen Sie eine ZIP Datei mit Ihren Ergebnissen und nennen Sie diese Datei `<ihr-kürzel>.zip`. Senden Sie diese ZIP Datei per E-Mail an `bkrt@zhaw.ch`; `stme@zhaw.ch` (beide Personen als Empfänger). Die Abgabe muss bis spätestens 13:05 Uhr erfolgt sein, zu spät zurückgesendete Arbeiten werden als „verspätete Abgabe“ mit 1.0 bewertet. Wir bestätigen Ihnen den Erhalt Ihrer Abgabe in einer kurzen, formlosen Mail. Bitte warten Sie diese Bestätigung ab. Sollten Sie 10 Minuten nach Ihrer Abgabe keine Bestätigung von uns erhalten haben, kontaktieren Sie uns bitte per Telefon (siehe unten).

Bei technischen Problemen oder Fragen sind wir die ganze Zeit per E-Mail oder MS-Teams Chat erreichbar.

Sollten weitergehende Probleme auftreten (Internet fällt aus, Rechnercrash, etc.) nehmen Sie bitte umgehend mit uns telefonisch Kontakt auf:

- Gerrit Burkert: 079 571 69 72
- Henrik Stormer: 079 860 10 95