# Anleitung Teilnehmer

Die Studie soll zwei unterschiedliche Methoden zur Entwicklung von Android App vergleichen.

**Aufgabe Umsetzung einer einfachen App (Fokus auf User Interface und Interaction)**

Die Aufgabe wird durch den Entwurf (Sketch) einer App definiert.

**Ziel:** klickbares UI, Wechsel auf Screens möglich (keine Backend Funktion wie GPS nötig)

App zur Aufzeichnung von Points-of-Interest

Name: POI App

Package: org.accapto.poiapp



1 Dashboard:

Auswahl-Buttons, auf entsprechenden Screen wechseln

2 New POI: neuen POI (Point of Interest) hinzufügen

Checkbox GPS und Public

Texteingabe Name

Funktion Foto machen und sichern

Zurück

3 Liste der POIs

Texteingabe Filter suchen, zurück

Neuen POI

4 Settings

Checkbox: GPS immer aktiv

Texteingabe: Username

# Aufgaben für die Teilnehmer

1. **Vorbefragung**

Umfrage zu Kenntnissen und persönlichen Daten

Link: <https://goo.gl/forms/NErq06nJVoyiQi9B2>

1. **Umsetzung eines App Prototypes**

**Methode A**

* Erstellen Sie ein Android Studio Projekt in gewohnter Weise

**Methode B**

- Machen Sie sich mit dem Beispiel und dem accapto.jar vertraut

siehe Dokumentation und Beispiele

- Erstellen Sie ein passendes XML Model für die Aufgabenstellung

- Erzeugen Sie mit Hilfe von accapto.jar und dem XML Model ein Android Studio Projekt

Führen Sie eventuelle Updates (Gradle, Android Studio, etc.) durch.

Führen Sie den Build-Prozess durch.

1. **Evaluierung:**

Überprüfen Sie das User Interface der entstandenen App folgen Sie dazu der Anleitung für die Evaluierung

1. **Nachbefragung**

Rückmeldung zum Workload und der entstandenen App

Link: <https://goo.gl/forms/dLmSLAtsTBuvausU2>

# Entwicklungsmethoden

## Methode A – Konventionelle Entwicklung

Verwenden von Android Studio (Activitiy, XML Layout und Intent, etc.)

Siehe Basis Tutorial

<https://developer.android.com/training/basics/firstapp/creating-project.html>

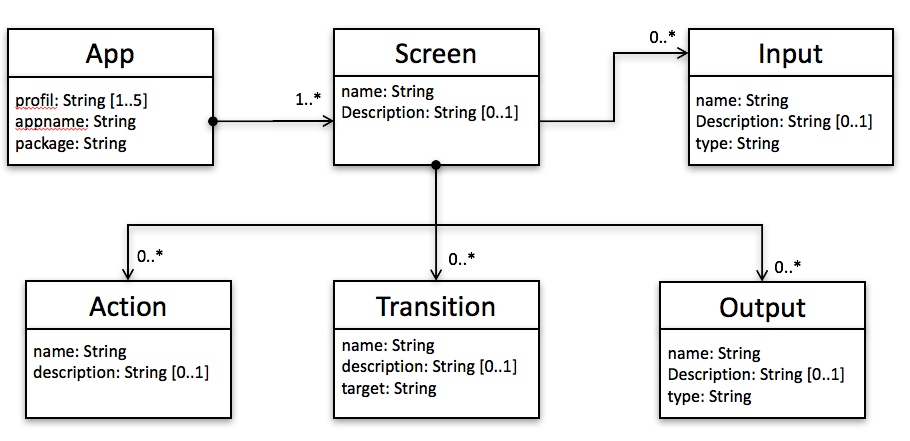
Intent zum Starten einer neuen Activity

Intent i = new Intent(this, OtherActivity.class);

startActivity(i);

## Methode B – Model-driven Development mit Accapto

**Model**



**XSD Definition**

Siehe accapto\_model\_v1\_2.xsd

**XML Model**

Beispiel in XML



Anmerkungen zum XML:

*input*:

möglich bei *type*

text, checkbox, radio, button

*output*:

möglich bei *type*

text, image

*transition*

*target* ist ein im Model enthaltener Screen

### Aufruf accapto

1.) create xml model file, see example new\_model.xml or <https://github.com/elmarkrainz/accapto/blob/master/model/accapto_model.xsd>

2.) create App Project from Model

**java -jar accapto.jar -i new\_model.xml**

3.) open Project in Android Studio or build it with Gradle

# Evaluierung der entstandenen App

Accessibility-Check der entstandenen App. Die Teilnehmer überprüfen jeweils Ihre eigene App. Für eine einheitliche Evaluierung der App Accessibility wird ein 14 Punkte Accessibility- Check in drei Bereichen durchgeführt.

Nutzung des Quick Accessibility Checks (QAC) für mobile Anwendungen

1. AT (z.B. Talkback)

2. Accessibility Scanner

3. manueller Check

Das Ergebnis ist ein einheitliches Accessibility Rating für alle entstandenen Apps.

## Vorbereitung Accessibility Check auf Android

Installation (falls nicht vorhanden von) via

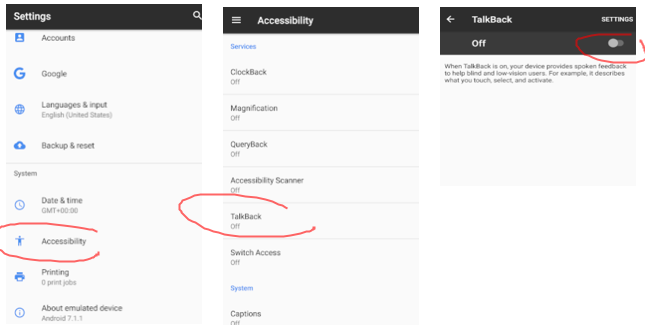
* Talkback <https://play.google.com/store/apps/details?id=com.google.android.marvin.talkback>
* Accessibility Scanner <https://play.google.com/store/apps/details?id=com.google.android.apps.accessibility.auditor>

bzw. siehe .apk Files für AVD

## Aktivierung von Talkback & Accessibility Scanner

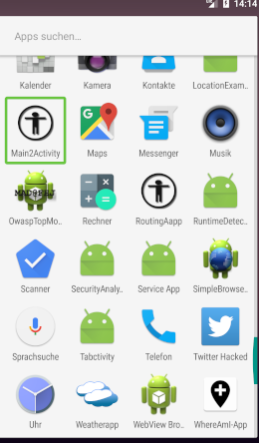
### Anwendung – Talkback

Settings -> Accessibility ->Talkback



Bei aktivierten *Talkback* ist *Explore-By-Touch* aktiv, d.h. die aktuelle Position (Icon, Widget, etc.) wird vorgelesen und ist markiert.

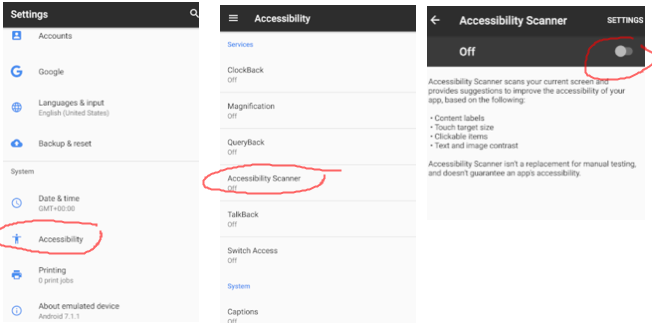
Durch Doppelklicken wird das aktuelle Element ausgewählt, Swipe Links oder rechts unterstützen Tabbing. Siehe Bild unten mit den wichtigsten Gesten.

### Anwendung - Accessibility Scanner

Siehe auch <https://support.google.com/accessibility/android/answer/6376570>

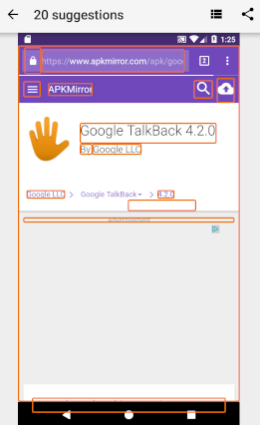
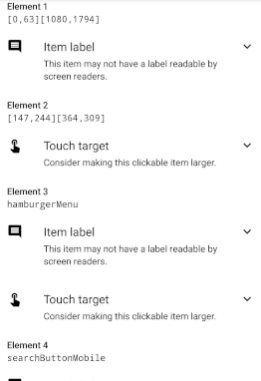
Starten via Settings -> Accessibility -> Accessibiliy Scanner



Wenn Accessibility Scanner aktiv ist, befindet sich ein Overlay Element am Bildschirm. Beim Klicken wird der aktuelle Screen (aktive App) gescannt und die Accessibility geprüft. Die geprüften Screens werden im Scanner gespeichert und können später wieder aufgerufen werden.



## Ergebnisse des Accessibility Scanners

Gefundene Fehler in den Bereichen

* Touch target
* Item label
* Item description
* Text contrast
* Image contrast

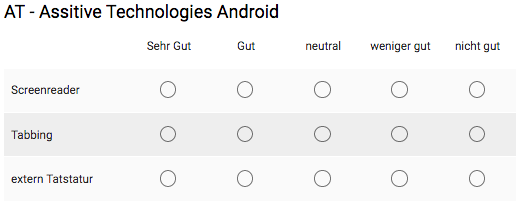
Siehe <https://support.google.com/accessibility/android/answer/6376559>

## Durchführung Accessibility Check auf Android

Die Bewertung erfolgt manuell und wird online in das Formular eingetragen. Link: <https://goo.gl/forms/SfKvn9do3s7NRscj1>

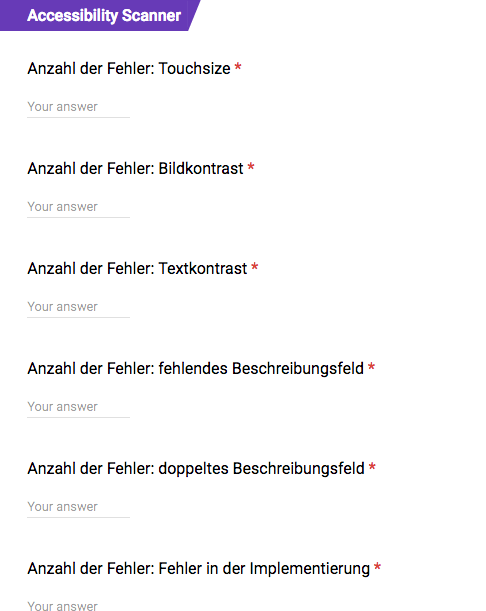
**1. Assitive Technologien (z.B. Talkback)**

Überprüfen Sie die Bedienung mit Talkback, achten Sie dabei auf die Rückmeldungen vom Screenreader, die Bedienung durch Tabbing und evtl. die Nutzung einer externen Tastatur (Bluetooth Tastatur).



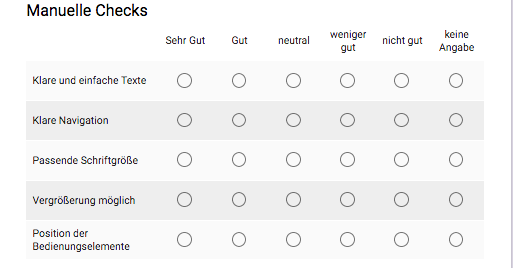
**2. Accessibility Scanner**

Verwenden Sie den Accessibility Scanner und tragen Sie die gefundenen Ergebnisse ein (0 bei keinem Fehler, ansonsten die Anzahl der Fehler in den verschiedenen Bereichen)

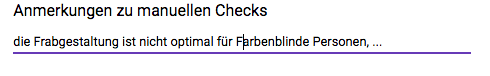


**3. manueller Check**

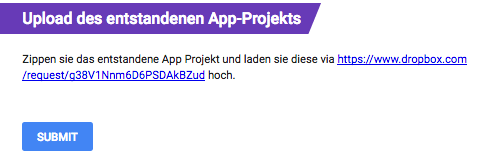
Überprüfen Sie die App anhand der Kriterien und bewerten Sie dieses.



Sie können in den drei Kategorien auch noch eigene Anmerkungen hinzufügen.



Zippen Sie das entstandene Projekt und laden Sie diesen über den Upload-Link hoch. Schicken Sie das Formular zum Abschluss ab.



# Anhang: Fragebogenteil Links

**Vorbefragung**

<https://docs.google.com/forms/d/e/1FAIpQLSfhkozWEBS-AR7vYv-8s3rolItkVQs-Bx548q-AC6ZZfzGtVA/viewform?usp=sf_link>

**Accessibility Bewertung**

<https://docs.google.com/forms/d/e/1FAIpQLScz64xFbaaEqXcPuA_bMj9jOtNnh00qezvaj_sE2TrSJK2xAg/viewform?usp=sf_link>

**Nachbefragung**

<https://goo.gl/forms/dLmSLAtsTBuvausU2>