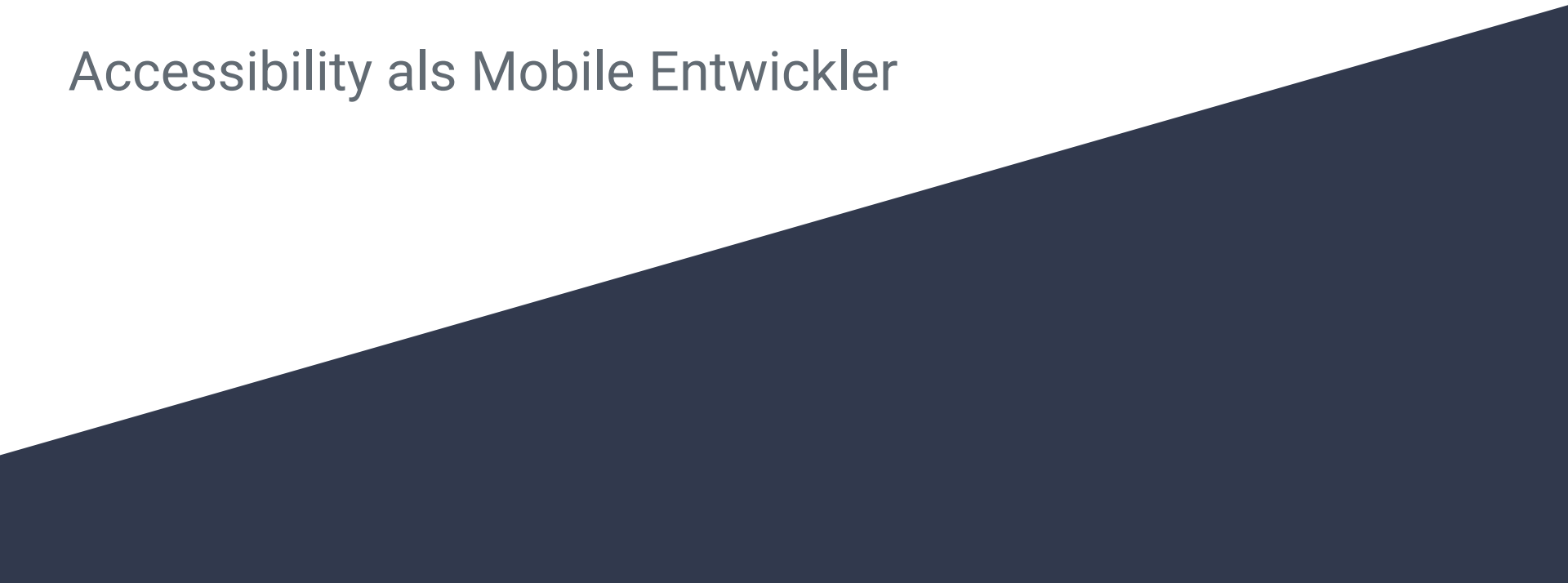


Apps für alle

Accessibility als Mobile Entwickler

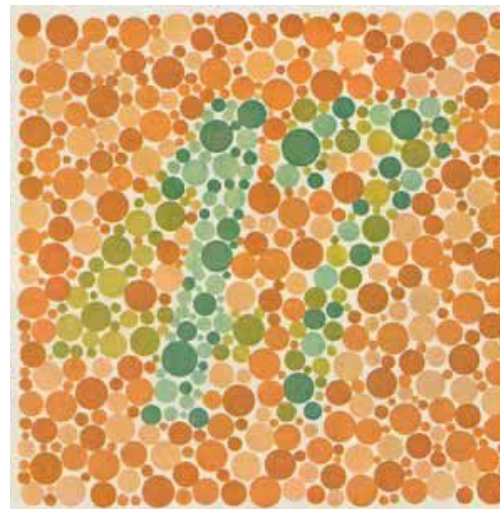
A dark blue diagonal gradient bar that starts from the bottom left corner and extends towards the top right corner, covering the lower half of the slide.



[Quelle: Wikimedia](#)



[Quelle: Wikimedia](#)



[Quelle: Wikimedia](#)



Quelle: Wikimedia, Jonn Leffmann



Quelle: Wikimedia



Quelle: cottonbro studio



Quelle: iam hogir



Quelle: Free Images



Quelle: Piotr Siedlecki

Accessibility Features kommen
uns allen zugute

Über mich



- 2018 Werkstudent App Entwicklung
- 2019 Thesis: “Entwicklung einer mobilen Anwendung zur Übersetzung der deutschen Lautsprache in die Deutsche Gebärdensprache”
- seit 2020 React Native Entwickler

Barrierefreiheitsstärkungsgesetz

Gesetz zur Umsetzung der
Richtlinie (EU) 2019/882
des Europäischen Parlaments
und des Rates über die Barrierefreiheits-
anforderungen für Produkte und
Dienstleistungen

Barrierefreiheitsstärkungsgesetz

- für digitale Produkte und Dienstleistungen
- Hersteller, Händler und Importeure von Produkten sowie Dienstleistungserbringer
 - Ausnahme:
 - < 10 Arbeitnehmer
 - + Jahresumsatz von max 2 Millionen Euro
- gültig ab 28.06.2025
- WCAG 2.2 Standard
- Erklärung zur Barrierefreiheit



**FOCUSING
ON ACCESSIBLE
DESIGNS
BECAUSE IT'S A
LEGAL REQUIREMENT.**



**FOCUSING ON
ACCESSIBLE DESIGN
BECAUSE GOOD
ACCESSIBILITY
BENEFITS EVERY USER.**

Definition laut Wikipedia

Barrierefreiheit bezeichnet die Gestaltung der Umwelt, die es allen Menschen ermöglicht, ohne Hindernisse mit ihrer Umgebung zu interagieren.

Wer macht Produkte barrierefrei?

- Management
- UX Designer
- Texter
- Developer
- QS Tester

Accessibility kann nicht nur von einer Person gedacht werden, sondern muss in allen Bereichen gedacht werden

Ansätze um Accessibility in App zu erhöhen

POUR

- Perceivable (Wahrnehmbarkeit)
 - z. B. richtiger Kontrast
- Operable (Bedienbarkeit)
 - z. B. Tastatursteuerung
- Understandable (Verständlichkeit)
 - z. B. Klare Fehlermeldungen
- Robust (Robustheit)
 - z. B. Nutzung von Screenreadern

Lesbarkeit von Texten (WCAG 1.4.3)

Text < 18pt oder Text < 14pt + fett

=> Kontrast der Farbe: 4.5:1

sonst

=> Kontrast der Farbe:

3:1

Text

Text

Große Bedienoberflächen (WCAG 2.5.5)

- Bereich min. 48 x 48dp (Android)
- muss nicht grafisch sichtbar sein

```
<ImageButton ...  
    android:paddingLeft="4dp"  
    android:minWidth="40dp"  
    android:paddingRight="4dp"  
  
    android:paddingTop="8dp"  
    android:minHeight="32dp"  
    android:paddingBottom="8dp" />
```



UI-Elemente beschreiben (WCAG 1.1)

- oft automatisch, wenn man richtiges Element verwendet
- keine Beschreibung des Typen der UI-Elemente
- Beschreibung nicht wiederholen

Bilder und Grafiken (WCAG 1.1)

- alternative Beschreibungen für alle Bilder, die Bedeutung vermitteln

```
contentDescription=stringResource(R.string.label_share)
```

- dekorative Elemente kennzeichnen

```
contentDescription=null
```

Label von Elementen (WCAG 3.3.2)

- Veränderbare Felder: hint Attribut

```
modifier = Modifier.semantics {contentDescription =  
    "Open the website"}
```

```
// Set hint for a TextField
```

```
TextField(  
    value = "",  
    onChange = { /* State update logic */ },  
    placeholder = { Text("Address") }  
)
```

Label von Elementen (WCAG 3.3.2)

- zusammengehörige Elemente gruppieren

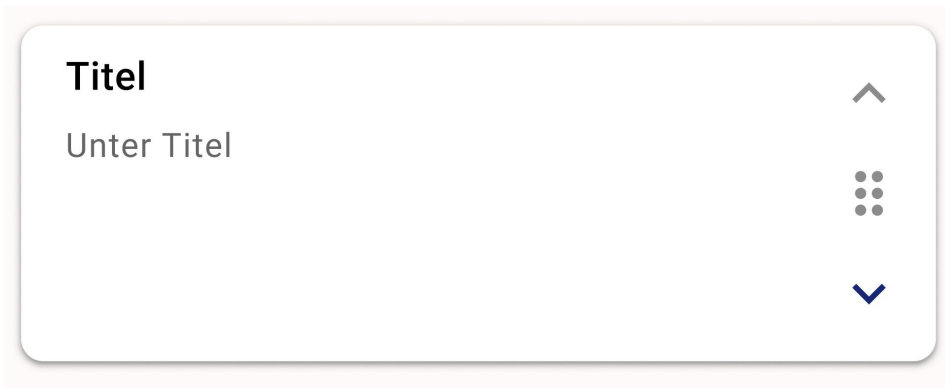
```
// Merge all semantics of box elements
Box(modifier = Modifier
    .semantics(mergeDescendants = true) { }
) {
    // Box content ...
}
```

Aktionen und Gesten (WCAG 2.5.1)

- Plattform-Gesten nicht überschreiben
- vereinfachte Gesten für gängige Interaktionen
- Gesten, sollten von Assistenzsystemen nutzbar sein

Aktionen und Gesten (WCAG 2.5.1)

- alternative Möglichkeiten um Aktion auszuführen
- Erweiterung von Aktionsbeschreibungen

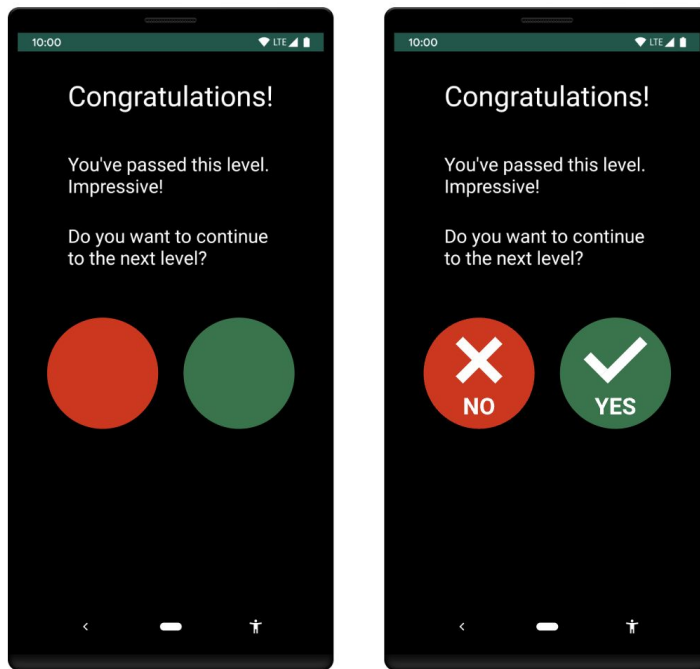


Standardelemente erweitern

- Viele Standardelemente haben Accessibility Informationen
- Sonderfälle können überschrieben werden
- was verändert wird, sollte neue Accessibility Info bekommen

Unterscheiden von Optionen (WCAG 1.4.1)

- mehrere optische Merkmale für Entscheidungsmöglichkeiten verbinden
- Emotionen oder ungewöhnliche Metaphern vermeiden



Medieninhalte (WCAG 2.2.2/ 1.2.1)

- Steuerelemente
- wichtige Informationen auch als Text anbieten



Testmöglichkeiten

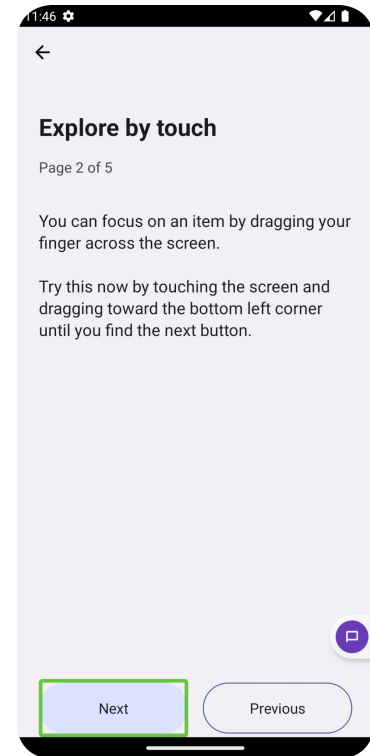
Arten zu testen

- Feedback von Menschen mit Behinderungen
- Manuell testen
- Analyse Tools
- Externe Testapps

Manuell testen

Talkback

- aus Play Store
- eigene Aktionen
 - links und rechts swipen um Item zu wechseln
 - Double Tab um Item zu aktivieren
 - 2 Finger zum Scrollen



Vergrößerte Darstellung

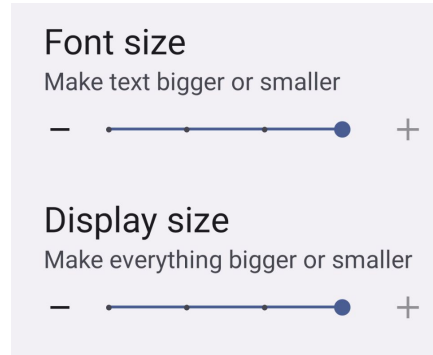
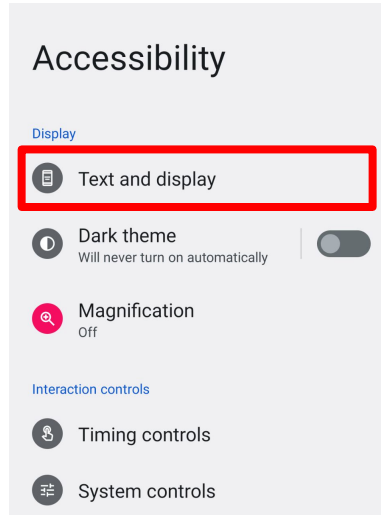


Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

HIER STEHT EIN SEHR LANGER TEXT

ERSTER BUTTON

ZWEITER BUTTON



Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum Lorem ipsum dolor sit amet, consetetur sadipscing elitr

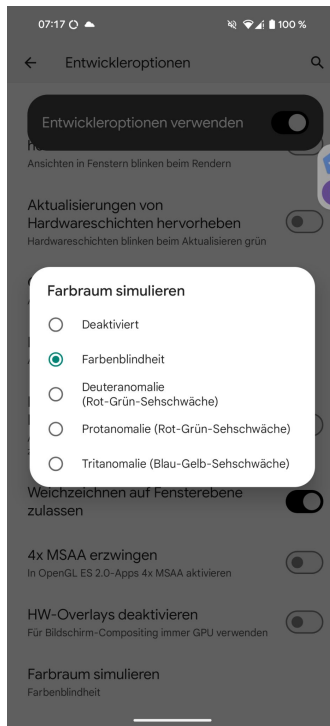
TER BUTTON

ZWEITER BUTTO



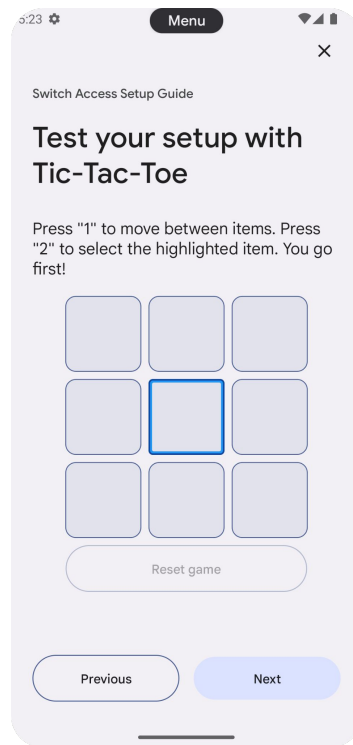
Farbsimulation

- Entwickler-einstellungen freischalten
- Entwickleroptionen
- Farbraum simulieren



Switch Access

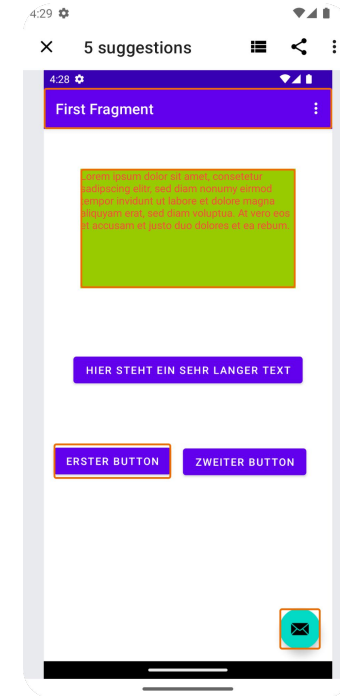
- aus Play Store
- Eingabemöglichkeiten:
 - Externer Schalter
 - Externe Tastatur
 - Tasten auf dem Gerät
 - Gesichtsbewegung
- mit zwei Tasten App bedienbar:
 - 1. Taste: Wechsel nächstes Element
 - 2. Taste: Auswahl Element



Analysertools

Accessibility Scanner

- aus Play Store oder in Android Studio
- Scannt Seite
- Verbesserungsvorschläge
 - Labels
 - Größe von Bedienfläche
 - Text und Bilder Kontrast
 - schwer Interpretierbare UI Elemente

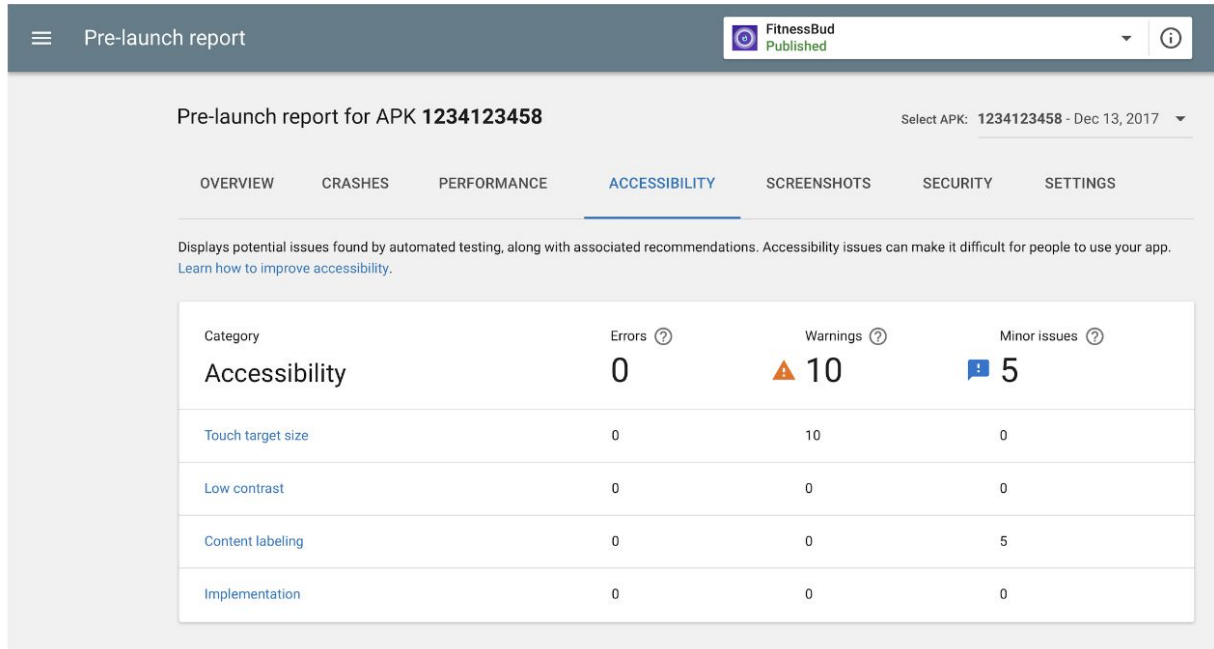


Quelle: Google Play Store

Pre-Launch Report in Google Play Console

- nach Upload der App verfügbar
- Verbesserungsvorschläge für:
 - Labels
 - Größe von Bedienfläche
 - Text und Bilder Kontrast
 - schwer Interpretierbare UI Elemente
- Schnappschüsse von Screens

Pre-Launch Report in Google Play Console

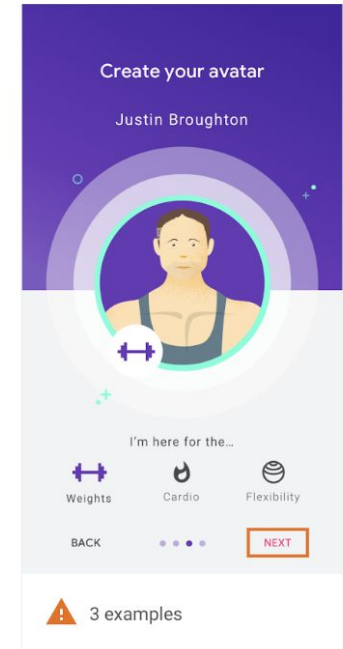


Pre-launch report for APK **1234123458** Select APK: 1234123458 - Dec 13, 2017

OVERVIEW CRASHES PERFORMANCE **ACCESSIBILITY** SCREENSHOTS SECURITY SETTINGS

Displays potential issues found by automated testing, along with associated recommendations. Accessibility issues can make it difficult for people to use your app. [Learn how to improve accessibility.](#)

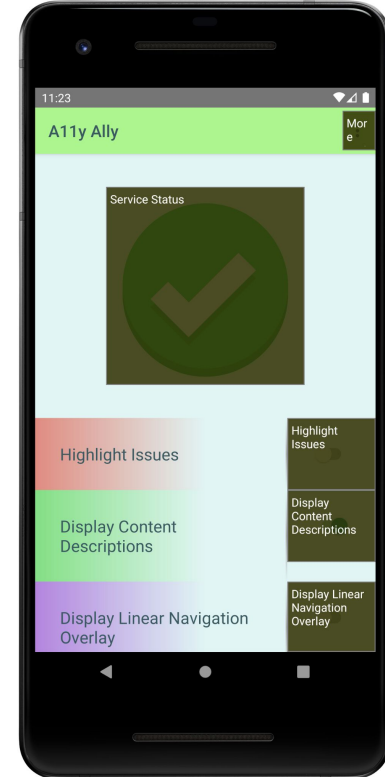
Category	Errors ?	Warnings ?	Minor issues ?
Accessibility	0	10	5
Touch target size	0	10	0
Low contrast	0	0	0
Content labeling	0	0	5
Implementation	0	0	0



Externe Testapps

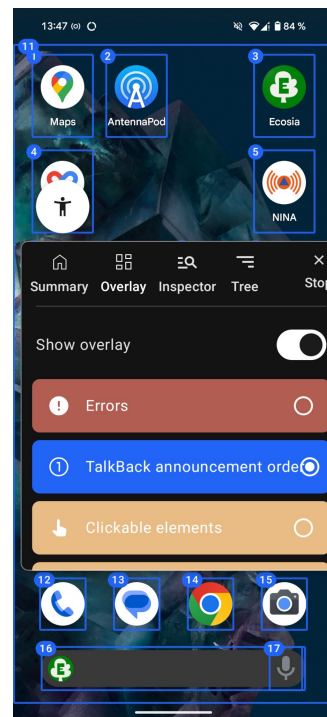
A11y von quittle

- Github Repo
- hebt mittels AR-Overlay die Probleme hervor
- Labels für Screen Reader
- Berichte über Barrierefreiheitsprobleme
- noch sehr rudimentär
- nur für Android



AccessibilityTester von Glitch

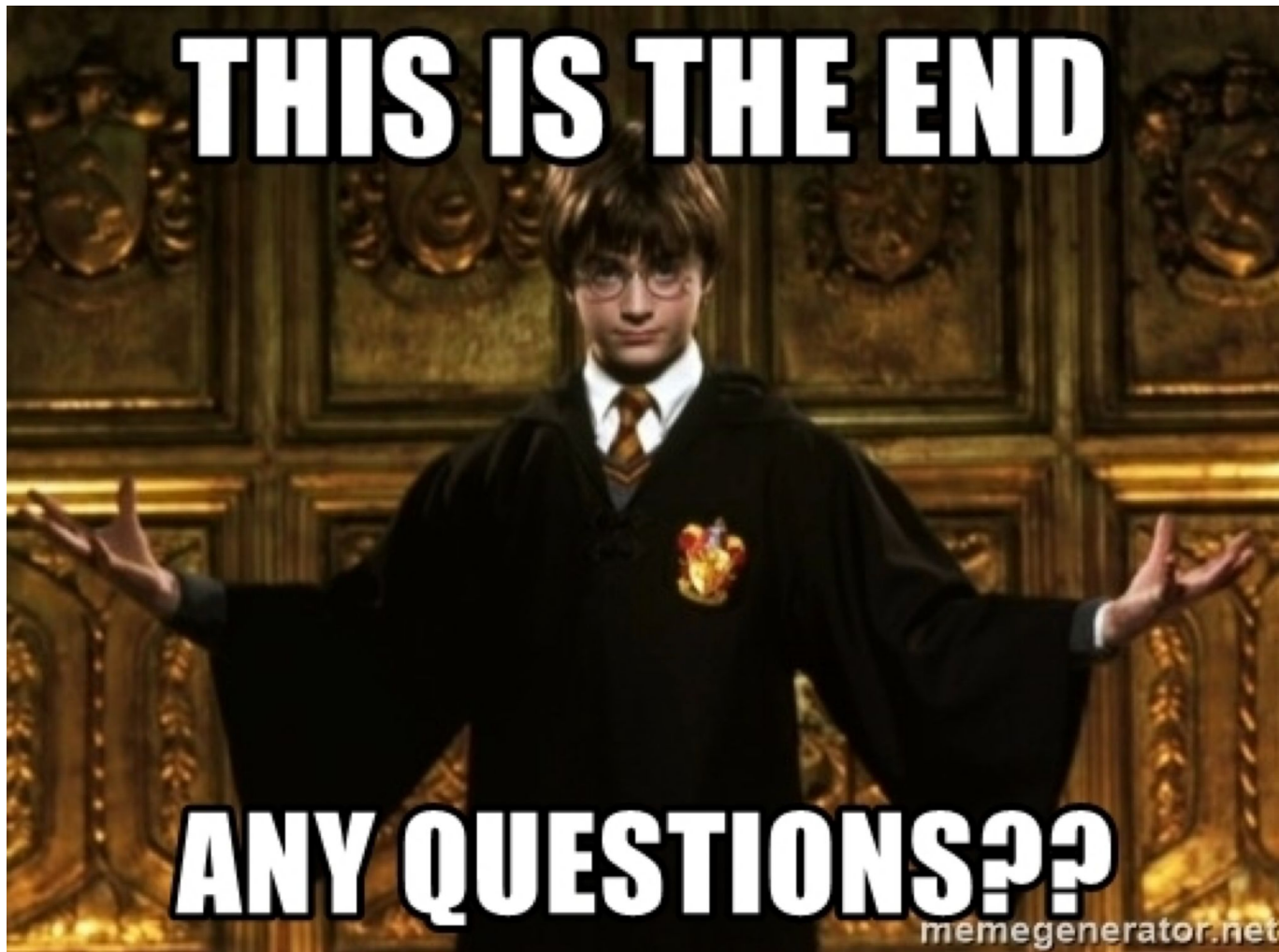
- aus Play Store
- Overlay für verschiedene Elemente
- zeigt Fehler an
- zeigt Vorlesereihenfolge an
- Möglichkeit Elemente genauer anzuschauen
- noch im Early Access
- nur für Android



Viele Möglichkeiten um ein
erstes Gefühl zu bekommen, ob
eine App accessible ist

Learnings

- Accessibility kommt uns allen zugute
- Accessibility kann nicht nur von einer Person gedacht werden, sondern muss in allen Bereichen gedacht werden
- Viele Möglichkeiten um ein erstes Gefühl zu bekommen, ob eine App accessible ist



Giulia Maier



Mastodon:
Lycentzia@kif.rocks

LinkedIn: Giulia Maier



Ressourcen (Android)

- Google Developer Guide:
<https://developer.android.com/guide/topics/ui/accessibility>
- Material UI Guide:
<https://m3.material.io/foundations/accessible-design/overview>
- Codelab:
<https://developer.android.com/codelabs/starting-android-accessibility#0>