

Hi! I'm Rachel 😊

- Software engineer
- Xamarin.Forms → .NET MAUI
- Eating, reading, exploring Boston
- Accessibility, mobile app development, tech for good

 @therachelkang

 @rachelkang

 <https://www.linkedin.com/in/rachel-j-kang>



Developing inclusive mobile apps

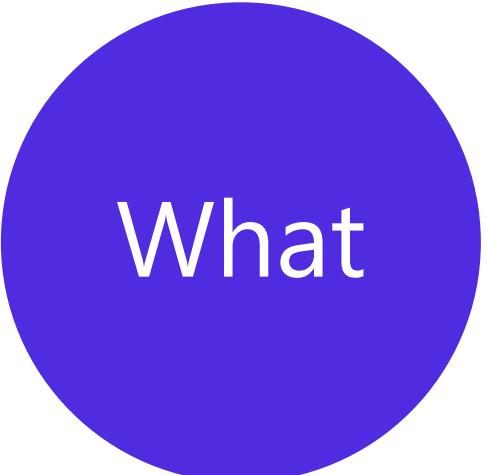
Rachel Kang (she/her)

Software Engineer | Xamarin.Forms → .NET MAUI

-  @therachelkang
-  @rachelkang
-  <https://www.linkedin.com/in/rachel-j-kang>



Developing inclusive mobile apps



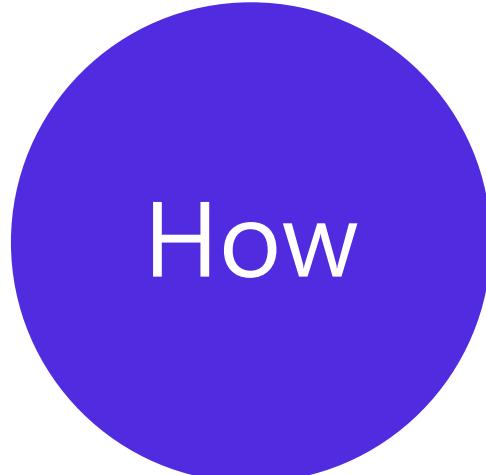
What

What does it mean?
What makes mobile apps
inclusive?
What does it mean to
develop mobile apps
inclusively?



Why

Why does this matter?
Why should you care?
Why is this relevant to you
as a mobile app developer
/ software engineer /
person in tech?



How

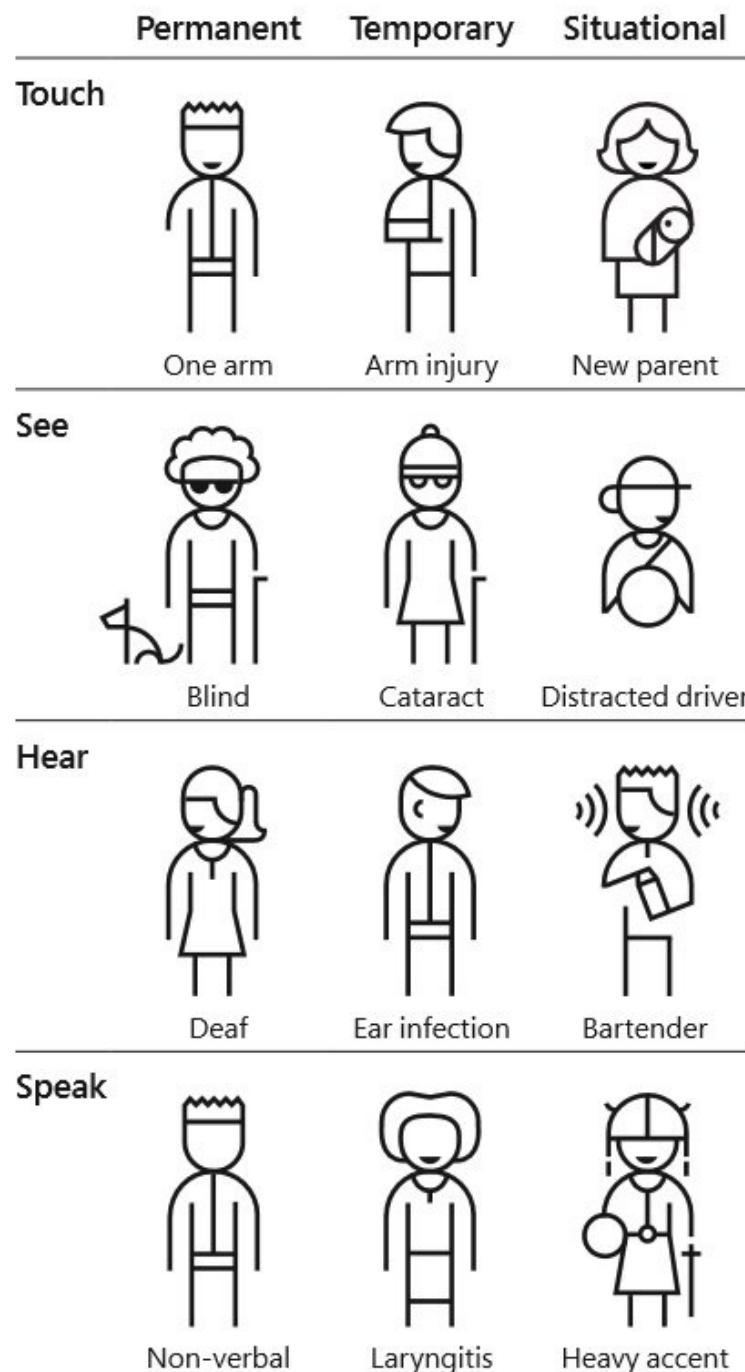
How can you achieve this?
How can you build
accessible and inclusive
mobile apps on Android?
iOS? Xamarin.Forms?

Developing **inclusive** mobile apps

- Inclusive vs. Accessible
 - **Accessible** → Inclusive of people of all **abilities**
 - **Inclusive** → Inclusive of people of all **abilities, identities, and experiences**
 - ability, age, gender, education, social grouping, income, ethnicity, nationality, sexual orientation, etc.
 - *If we use our own abilities as a baseline, we make things that are easy for some people to use, but difficult for everyone else.*
- *Great accessibility shouldn't be an obscure feature that only a small number of people use. At its best, accessibility should be a first-class citizen of the product or service you are creating and should benefit everyone.*

— Microsoft Design

— Rob Whitaker,
Developing Inclusive Mobile Apps: Building Accessible Apps for iOS and Android



The Persona Spectrum

We use the Persona Spectrum to understand related mismatches and motivations across a spectrum of permanent, temporary, and situational scenarios. It's a quick tool to help foster empathy and to show how a solution scales to a broader audience.

<https://www.microsoft.com/design>

Developing **inclusive** mobile apps

- Inclusive design principles
 - Provide comparable experience
 - Consider situation
 - Be consistent
 - Give control
 - Offer choice
 - Prioritize content
 - Add value

<https://inclusivedesignprinciples.org>

Developing inclusive mobile apps

- Inclusive design principles
 - Provide comparable experience --> i.e. content for alternatives
 - Consider situation --> i.e. color contrast
 - Be consistent --> i.e. consistent design patterns
 - Give control --> i.e. allow zoom
 - Offer choice --> i.e. accessible alternatives
 - Prioritize content --> i.e. prioritizing tasks
 - Add value --> i.e. make task completion easier

Developing inclusive mobile apps

- Web Content Accessibility Guidelines (WCAG)
 - Global a11y standard and legal benchmark for both web and mobile
 - 4 principles

Perceivable

Information and user interface components must be presentable to users in ways they can perceive.

- Text alternatives
- Time-based media
- Adaptable
- Distinguishable

Operable

User interface components and navigation must be operable.

- Keyboard accessible
- Time limits
- Seizures and physical reactions
- Navigable
- Input modalities

Understandable

Information and the operation of user interface must be understandable.

- Readable
- Predictable
- Input assistance

Robust

Content must be robust enough that it can be interpreted by ...assistive technologies.

- Compatible

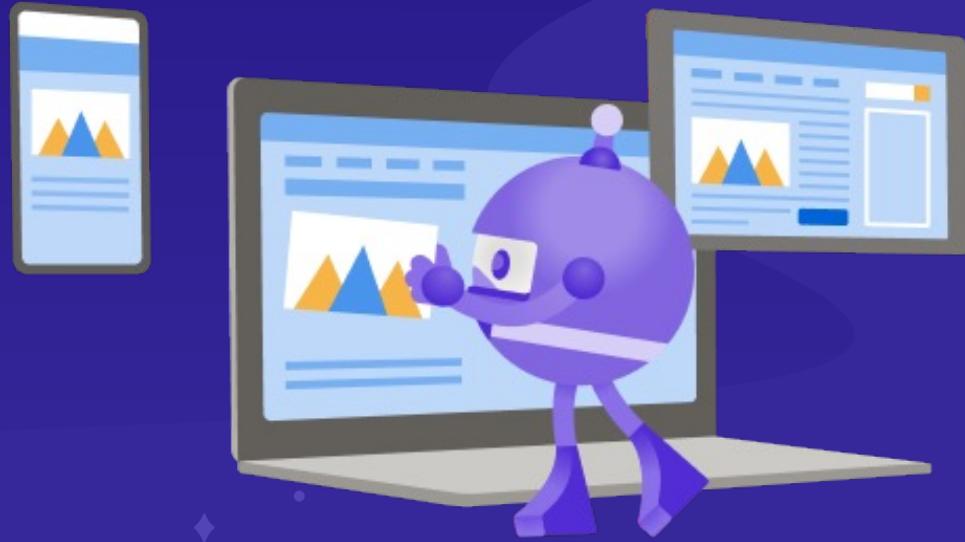
<https://www.w3.org/WAI/standards-guidelines/wcag/>

<https://www.w3.org/WAI/WCAG21/quickref/>

<https://www.w3.org/TR/WCAG21/>

Developing inclusive mobile apps

- **Empathy** is the name of the game
- Do your due diligence!
 - Accessibility settings and features
 - i.e. screen readers (TalkBack/VoiceOver), keyboard accessibility
 - Accessibility Model and APIs
 - <https://developer.android.com/guide/topics/ui/accessibility>
 - <https://developer.apple.com/accessibility/ios/>
 - <https://docs.microsoft.com/xamarin/xamarin-forms/app-fundamentals/accessibility/>
- Test and leverage available tools/resources



Let's explore together!

Android vs. iOS Accessibility APIs



android

```
<TextView  
    ...  
    android:importantForAccessibility="no"  
/>
```

```
textViewName.setImportantForAccessibility(  
    IMPORTANT_FOR_ACCESSIBILITY_NO)
```



```
labelName.isAccessibilityElement = false
```

Android vs. iOS Accessibility APIs



```
<Button  
    ...  
    android:contentDescription="Submit" />
```

```
buttonName.setContentDescription("Submit")
```



```
buttonName.accessibilityLabel = "Play"
```

Android vs. iOS Accessibility APIs



```
<TextView  
    ...  
    android:labelFor="@+id/thingToLabel" />  
  
<EditText  
    ...  
    android:id="@+id/thingToLabel" />"  
  
thingThatIsLabeling.setLabelFor(  
    thingToLabel.id)
```

Android vs. iOS Accessibility APIs



```
sliderName.accessibilityValue = "0"
```

Android vs. iOS Accessibility APIs



```
<EditText  
    ...  
    android:hint="Email address" />
```

```
editTextName.setHint("Email address")
```



```
buttonName.accessibilityHint =  
    "Sends your message."
```

Android vs. iOS Accessibility APIs



importantForAccessibility
contentDescription
labelFor
hint



isAccessibilityElement
accessibilityLabel
accessibilityValue
accessibilityHint

Android vs. iOS Accessibility APIs



importantForAccessibility
contentDescription
labelFor
hint



AutomationProperties.
IsInAccessibleTree
AutomationProperties.
Name
AutomationProperties.
HelpText
AutomationProperties.
LabeledBy



isAccessibilityElement
accessibilityLabel
accessibilityValue
accessibilityHint



Demo time!

Android vs. iOS Accessibility APIs



importantForAccessibility
contentDescription
labelFor
hint



isAccessibilityElement
accessibilityLabel
accessibilityValue
accessibilityHint

Android vs. iOS Accessibility APIs



importantForAccessibility
contentDescription
labelFor
hint
focusable
accessibilityTraversalAfter,
accessibilityTraversalBefore
nextFocusDown, nextFocusUp,
nextFocusLeft, nextFocusRight
nextFocusForward
isAccessibilityHeading
minWidth, minHeight
colorControlHighlight



isAccessibilityElement
accessibilityLabel
accessibilityValue
accessibilityHint
accessibilityElementsHidden
accessibilityViewIsModal
accessibilityElements
accessibilityTraits
accessibilityLanguage
accessibilityViewIsModal
accessibilityFrame,
accessibilityActivationPoint
accessibilityIncrement(),

Android vs. iOS Accessibility APIs



importantForAccessibility
contentDescription
labelFor
hint
focusable
accessibilityTraversalAfter,
accessibilityTraversalBefore
nextFocusDown, nextFocusUp,
nextFocusLeft, nextFocusRight
nextFocusForward
isAccessibilityHeading
minWidth, minHeight
colorControlHighlight



isAccessibilityElement
accessibilityLabel
accessibilityValue
accessibilityHint
accessibilityElementsHidden
accessibilityViewIsModal
accessibilityElements
accessibilityTraits
accessibilityLanguage
accessibilityViewIsModal
accessibilityFrame,
accessibilityActivationPoint
accessibilityIncrement(),

Accessibility in Xamarin.Forms

- Native support
- Automation properties
 - <https://docs.microsoft.com/xamarin/xamarin-forms/app-fundamentals/accessibility/automation-properties>
- Xamarin Community Toolkit
 - [Semantic Effects](#)
 - [Semantic Order View](#)
 - [Semantic Extensions](#)
- Learning path
 - <https://docs.microsoft.com/learn/modules/xamarin-forms-accessibility/>

Check out .NET MAUI!

The screenshot shows a Microsoft Docs page titled "Build your first app" for .NET MAUI. The page includes a sidebar with navigation links like ".NET MAUI", "What is .NET Multi-platform App UI?", "Supported platforms", "Get started", "Fundamentals", "User interface", and "Customize controls". The main content area features a "Choose your development environment" section with "Visual Studio 2022" and ".NET CLI" options. A note states: "In this tutorial, you'll learn how to create and run your first .NET Multi-platform App UI (.NET MAUI) app." Below this is a "Note" box: "Visual Studio for Mac support will arrive in a future release." The page also has a "Prerequisites" section with requirements for Visual Studio 2022, a configured .NET MAUI environment, and an Android emulator. At the bottom, there's a "Get started with Visual Studio 2022" link.

- Build your first app
 - <https://docs.microsoft.com/dotnet/maui/get-started/first-app?pivots=windows>
- Learn more about .NET MAUI
 - <https://docs.microsoft.com/dotnet/maui>

Accessibility in .NET MAUI

- Semantic properties
 - <https://docs.microsoft.com/dotnet/maui/fundamentals/accessibility>
- MAUI Community Toolkit
 - <https://github.com/CommunityToolkit/Maui>
- We're working on it!
 - <https://github.com/xamarin/Xamarin.Forms>
 - <https://github.com/dotnet/maui>
- Share your thoughts/ideas with us 😊
 - <https://github.com/dotnet/maui>
 - <http://aka.ms/discord-maui>

Thanks for joining!

-  @therachelkang
-  @rachelkang
-  <https://www.linkedin.com/in/rachel-j-kang>

