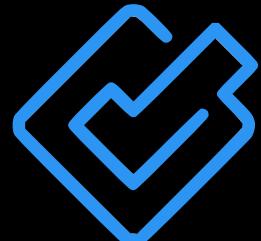


A Friend's Tale



A Friend's Tale

Unable to order food online

Unable to book a cab online

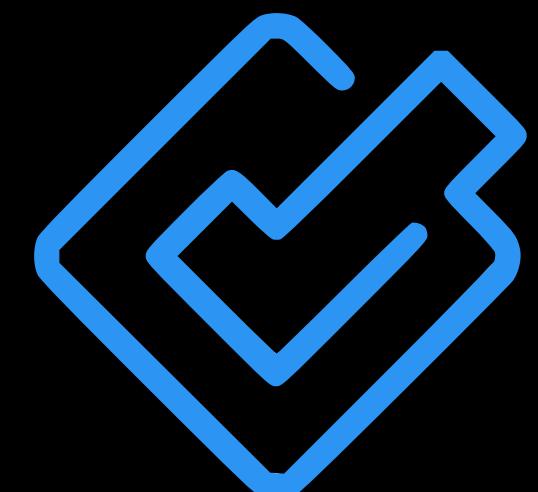
Struggles with simple tasks



Let me introduce
myself



Rizwan Ahmed



ohmyswift.com



Enhancing UX through Accessibility in iOS



Over 1 billion people live
with some form of
disability



Why bother?

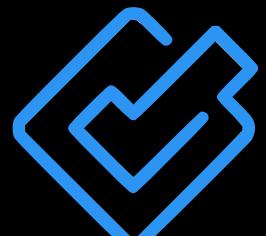
Several reasons



Why bother? Several reasons

It is the right thing to do

Everyone should get to experience
the joy of using a well-built software



Challenges

Vision The text is hard to read

Hearing No captions in the video, so can't understand the context

Physical and motor skills I can't progress because I can't touch a button

Literacy and learning I can't decipher the language. I don't know what it is conveyed



To make
accessible apps,
we need to have an
multi-dimensional
mindset



Let's get started



Dynamic Type



Dynamic Type

Accomplishes one simple goal -
Make sure everyone can
read text at a **size of their
preference**



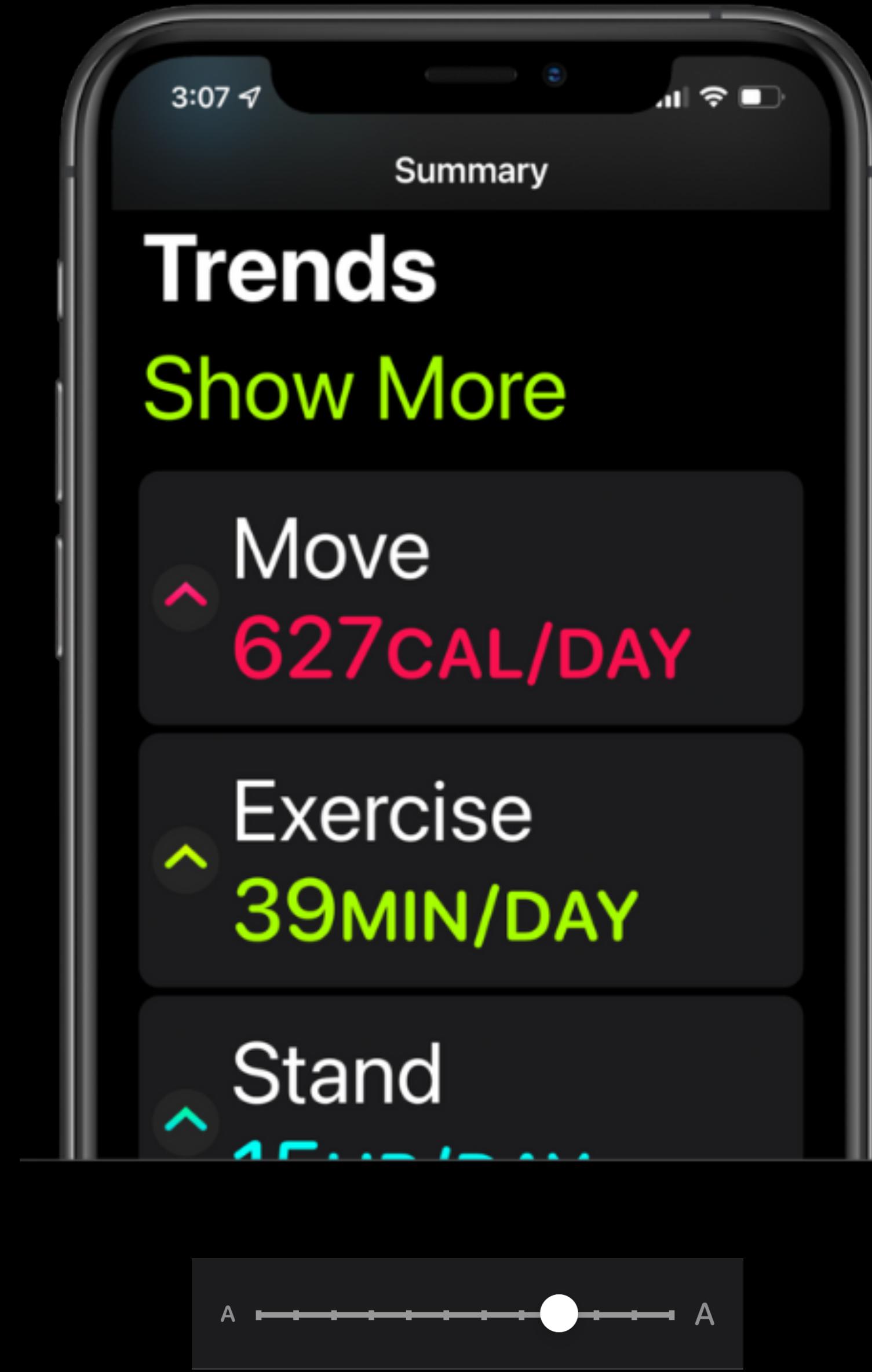
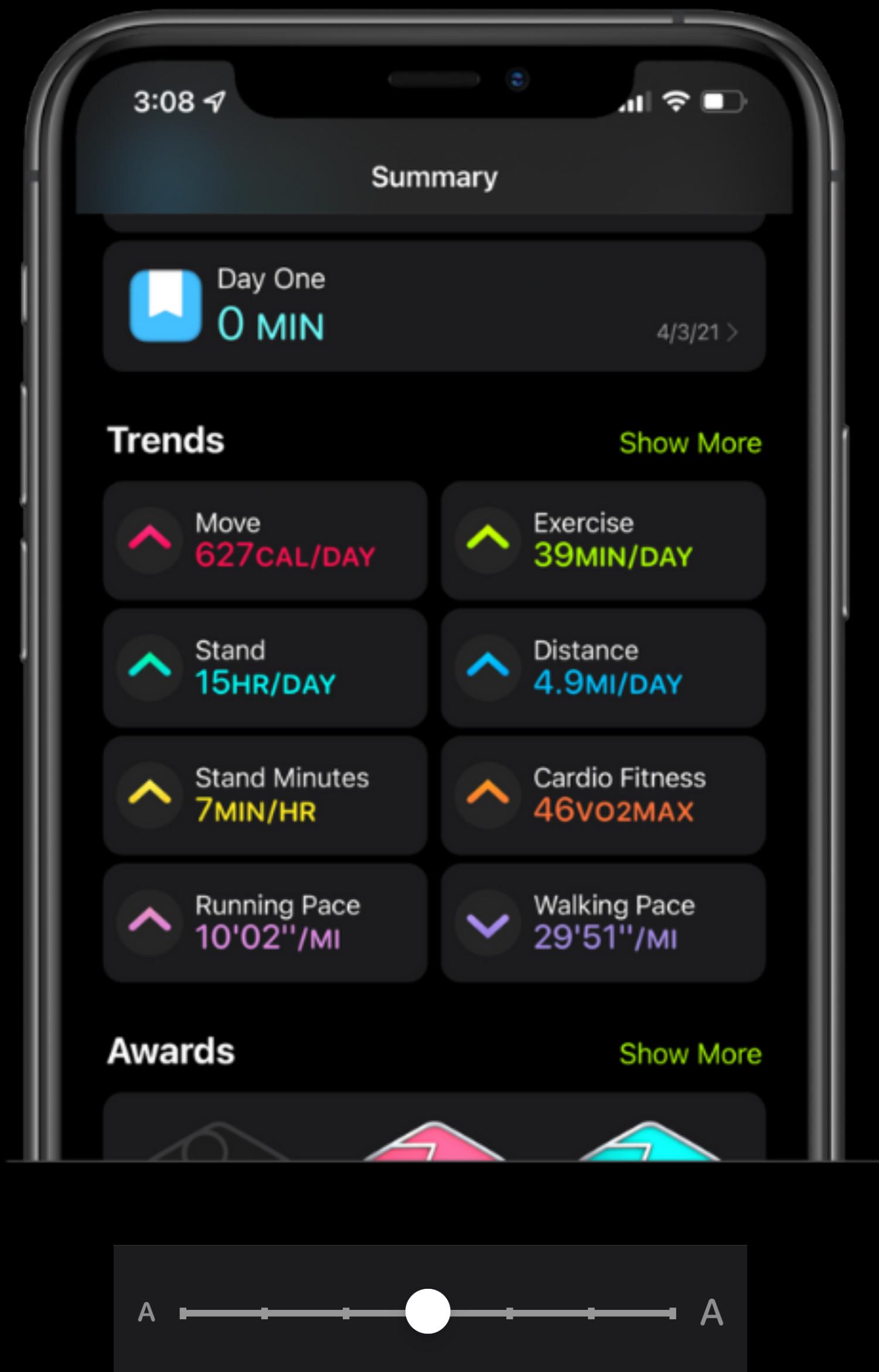
Dynamic Type

Set `adjustsFontForContentSizeCategory`
to `true`

Assign a font vended from a text style
using
`UIFont.preferredFont(forTextStyle:)`



Dynamic Type



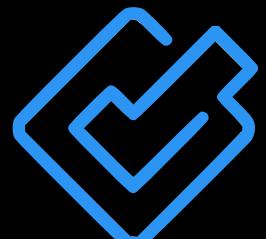
CUSTOM FONTS?



Dynamic Type



CUSTOM FONTS



Dynamic Type ❤️ Custom Fonts

```
let customFont = UIFont(name: String, size:  
CGFloat)
```

```
let metrics = UIFontMetrics(forTextStyle:  
UIFont.TextStyle)
```

```
let scaledFont = metrics.scaledFont(for:  
customFont)
```

```
label.font = scaledFont
```



Dynamic Type ❤️ Custom Fonts

The image displays two side-by-side screenshots of a mobile application interface, likely a task management app, demonstrating the integration of Dynamic Type and Custom Fonts.

Left Screenshot (Dynamic Type):

- Delayed** (Section Header)
- Webinar on Zoho Tasks (Task)
 - Open** button
 - User profile icon
 - Exclamation mark icon
 - Mon, Aug 28 (Due Date)
- Draft project proposal (Task)
 - Open** button
 - User profile icon
 - Exclamation mark icon
 - Wed, Aug 2 (Due Date)
- Expenses submission (Task)
 - Open** button
 - User profile icon
 - Exclamation mark icon
 - Jul 11 (Due Date)
- Review document for Legal team (Task)
 - Open** button
 - User profile icon
 - Exclamation mark icon
 - Jul 11 (Due Date)
- Business proposal review (Task)
 - Open** button
 - User profile icon
 - Exclamation mark icon
 - Jul 11 (Due Date)

Right Screenshot (Custom Fonts):

- Delayed** (Section Header)
- Webinar on Zoho Tasks (Task)
 - Open** button
 - User profile icon
 - Exclamation mark icon
 - Mon, Aug 28 (Due Date)
- Draft project proposal (Task)
 - Open** button
 - User profile icon
 - Exclamation mark icon
 - Wed, Aug 2 (Due Date)
- Expenses submission (Task)
 - Open** button
 - User profile icon
 - Exclamation mark icon
 - Jul 11 (Due Date)
- Review document for Legal team (Task)
 - Open** button
 - User profile icon
 - Exclamation mark icon
 - Jul 11 (Due Date)
- Business proposal review (Task)
 - Open** button
 - User profile icon
 - Exclamation mark icon
 - Jul 11 (Due Date)

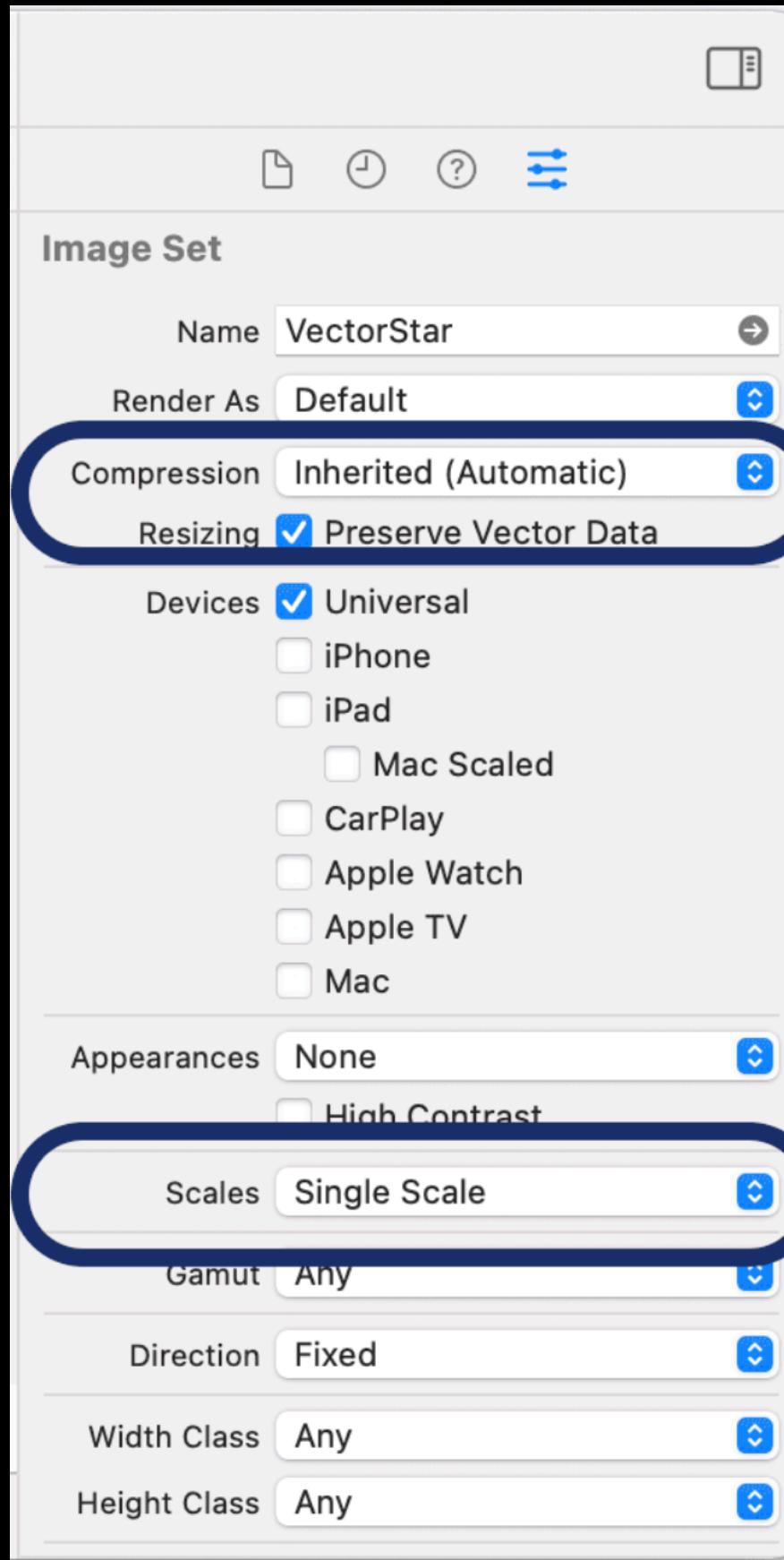
Bottom Icons:

- Blue checkmark icon (left)
- Red, green, and blue overlapping square icon (right) with the word "zoho" below it.

Scaling Images



Scaling Images



Set
`adjustsImageSizeForAccessibilityContentSizeCategory` to true

Use a vector image

Adjust your constraints
accordingly



Scaling Images

Delayed 32

Webinar on Zoho Tasks
Open ! Mon, Aug 28

Draft project proposal
Open ! Wed, Aug 2

Expenses submission
Open ! Jul 11

Review document for Legal team
Open ! Jul 11

Business proposal review
Open ! Jul 11

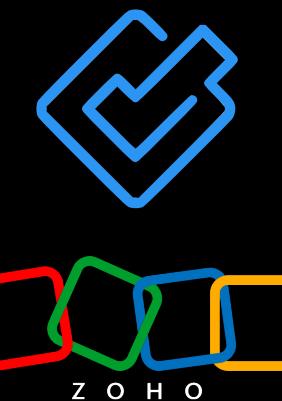
Monthly car wash 🚗

Delayed 32

Webinar on Zoho Tasks
Open ! Mon, Aug 28

Draft project proposal
Open ! Wed, Aug 2

Expenses submission
Open ! Jul 11

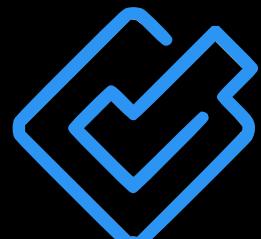


UIAccessibility



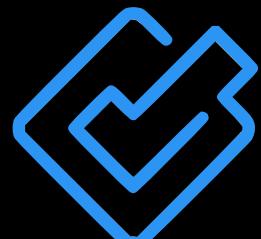
UIAccessibility

The UIAccessibility is an informal protocol that provides accessibility information about an app's user interface elements

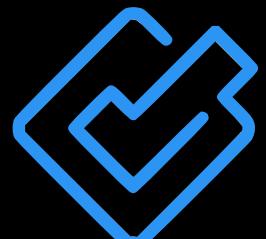


UIAccessibility

Most of the things are handled automatically and hence it is overlooked



Accessibility properties



Accessibility properties

- accessibilityLabel
- accessibilityValue
- accessibilityHint
- isAccessibilityElement
- accessibilityTraits
- accessibilityFrame



accessibilityLabel



A short, to-the-point string that identifies the element. It should be localized

...

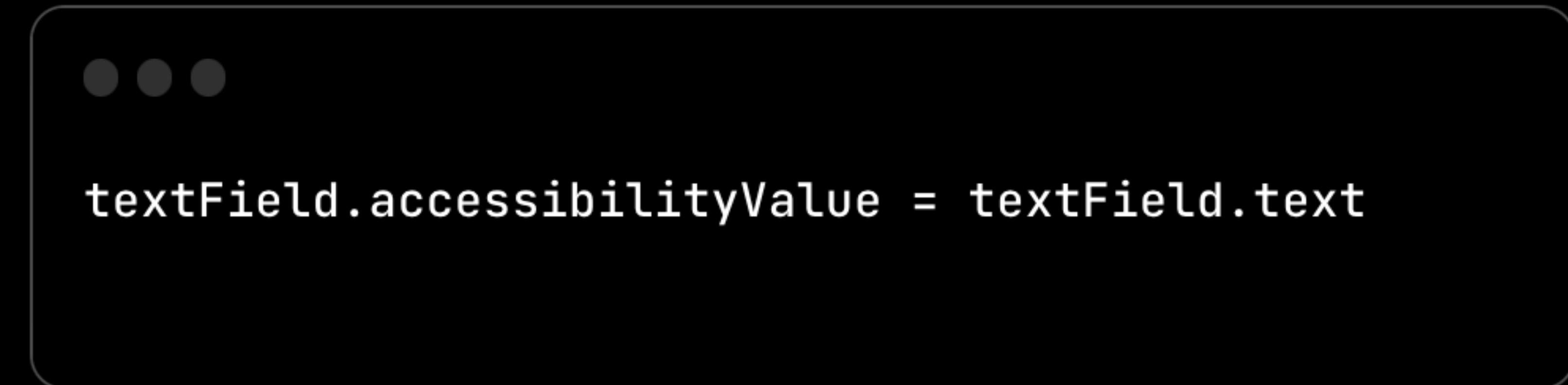
```
imageView.accessibilityLabel = "Abeer's profile picture"
```



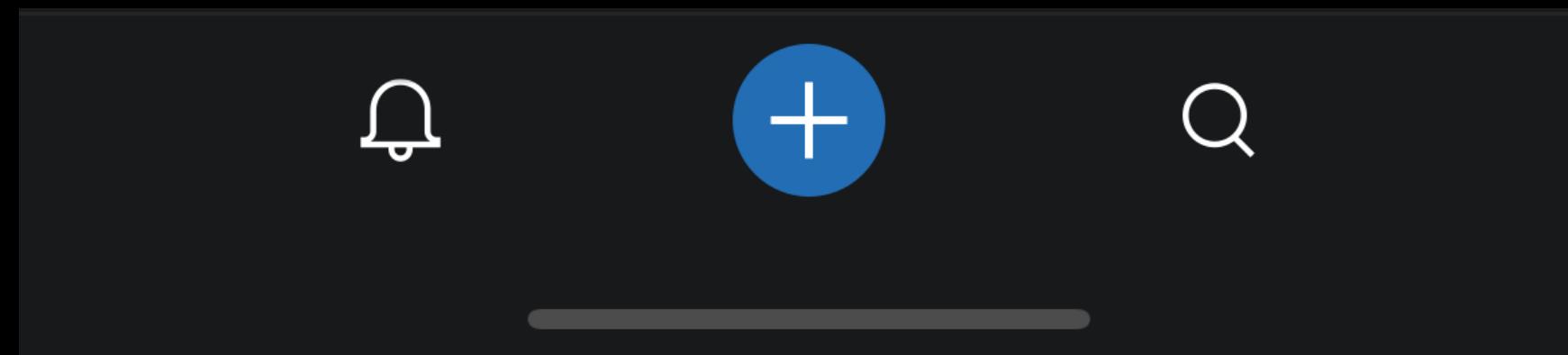
Z O H O

accessibilityValue

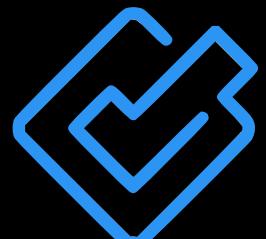
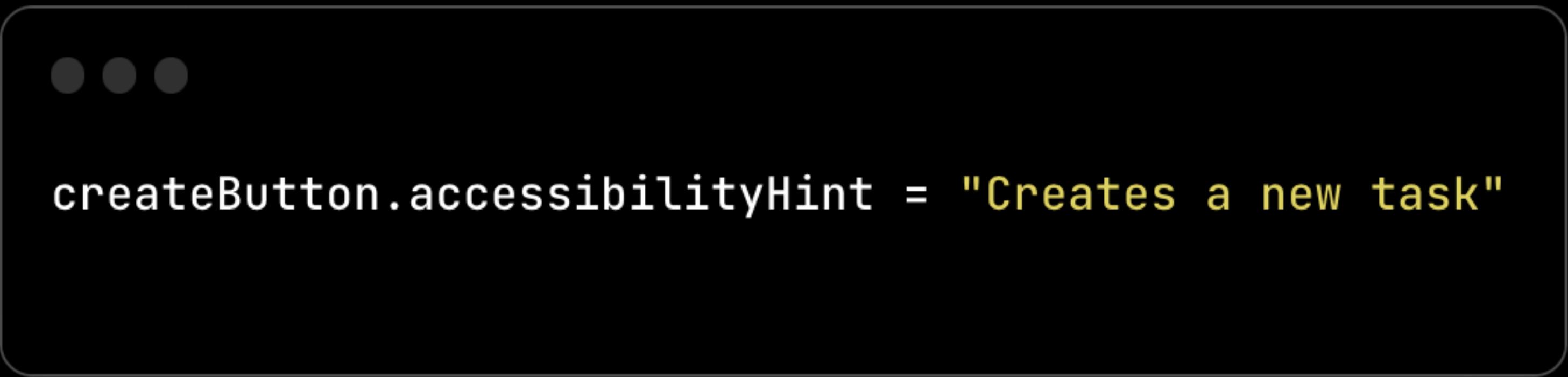
The element's value if it has one. It should be localized



accessibilityHint



Briefly describes what would happen if the user interacted with the element. It should be localized



ZOHO

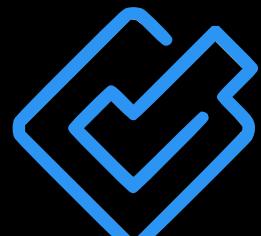
VoiceOver



VoiceOver

VoiceOver is the marquee element of accessibility on iOS

VoiceOver makes use of accessibility properties for functioning

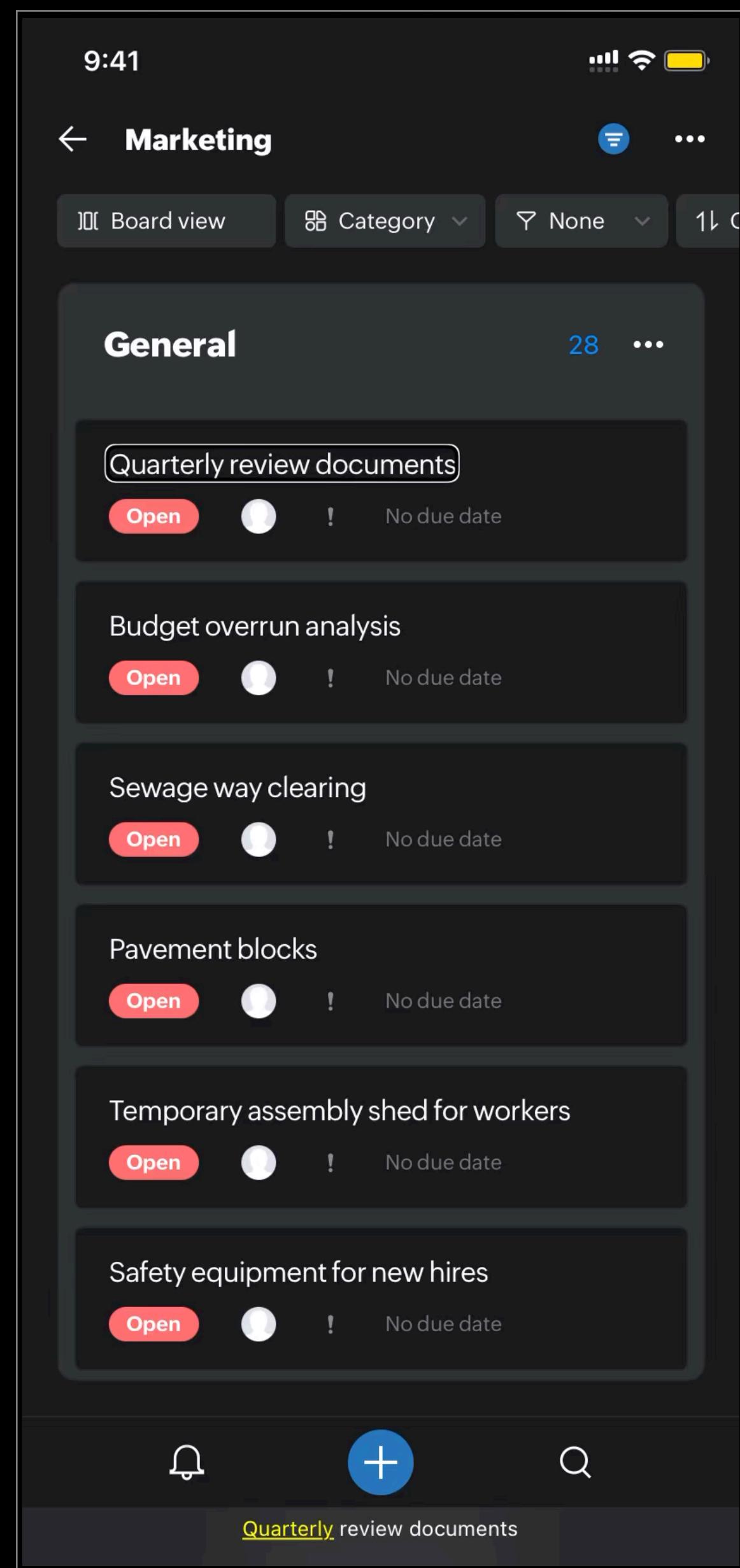


Z O H O

VoiceOver

Set `isAccessibilityElement = true`

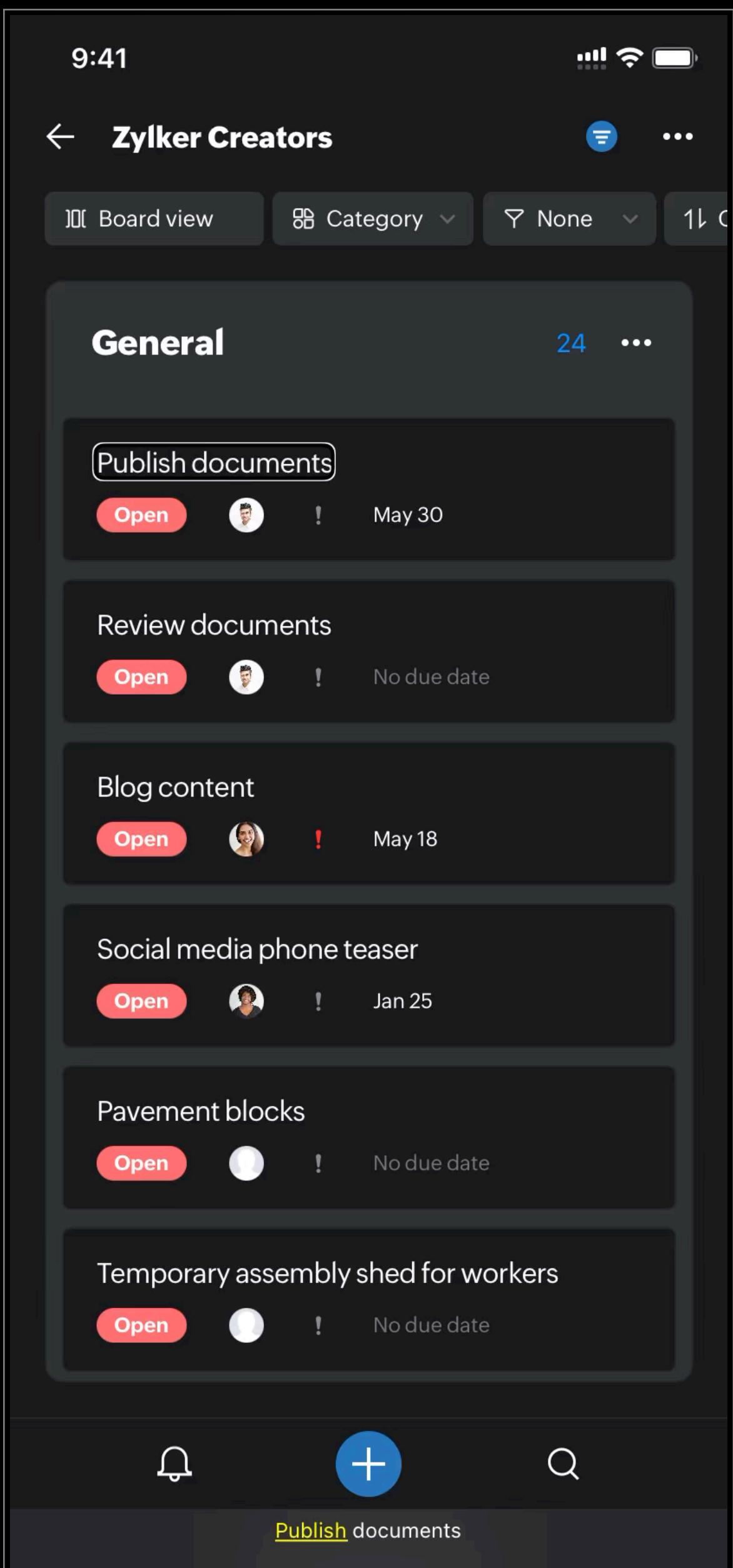
```
func addAccessibility() {  
    [titleLabel, dateLabel, statusButton, priorityButton, assigneeImageView].forEach {  
        $0.isAccessibilityElement = true  
    }  
}
```



Z O H O

VoiceOver

Add proper labels

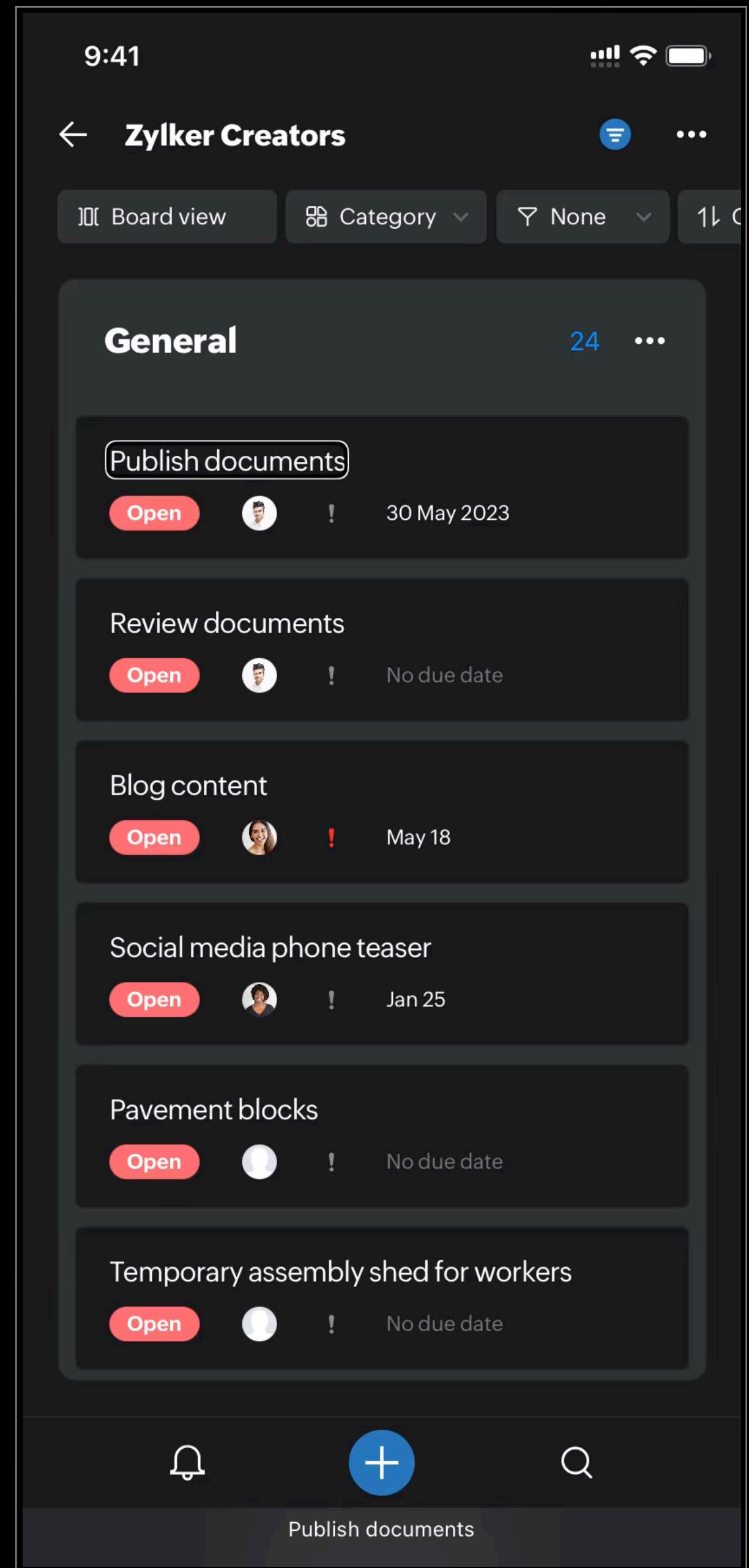


VoiceOver

Add hints

Adding proper accessibility hints

```
self.statusButton.accessibilityHint = "Tap to change the status."
```



VoiceOver



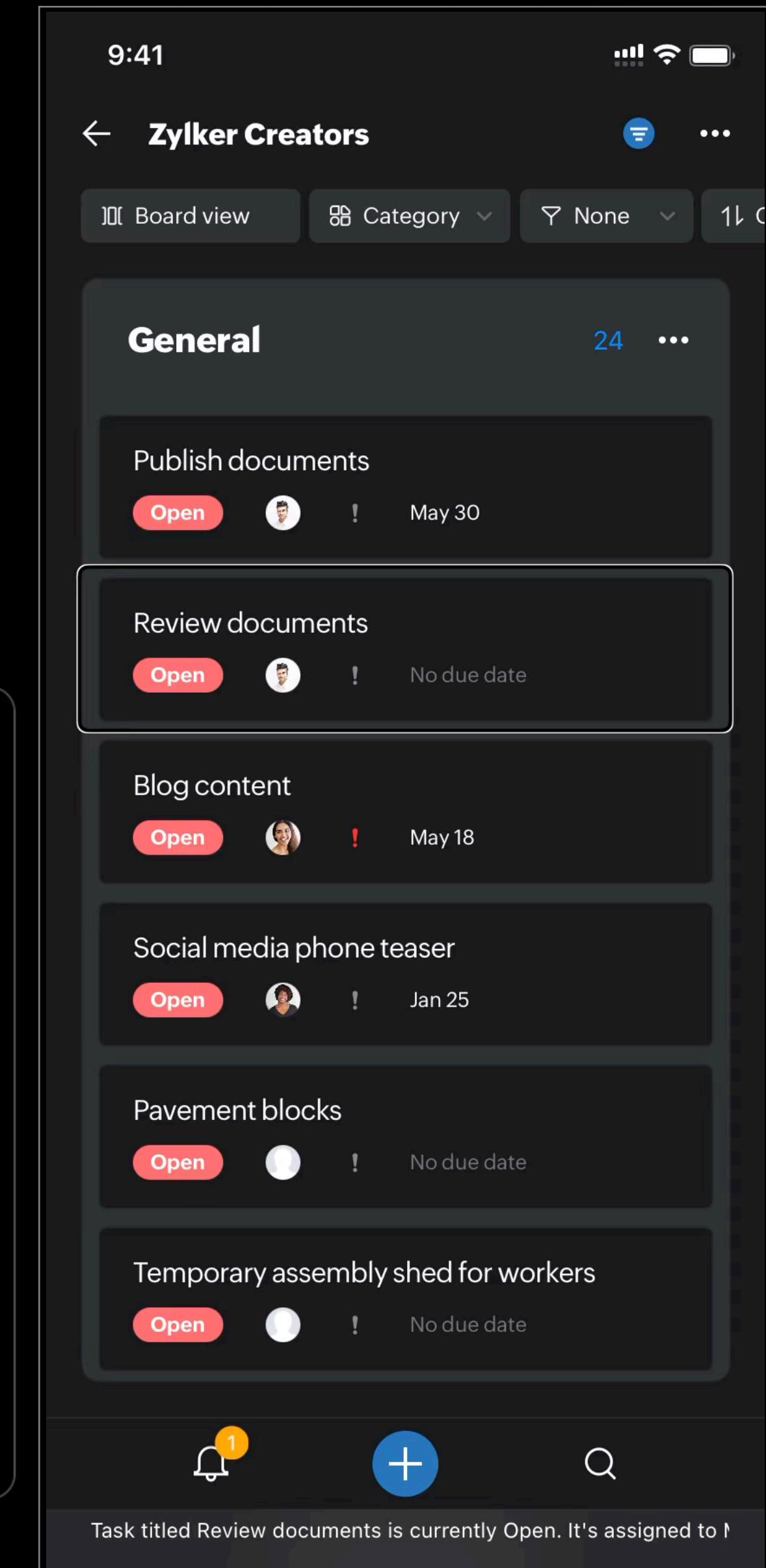
Flattening the hierarchy



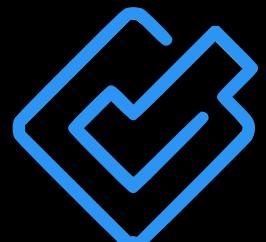
...

Flatten the hierarchy

```
[titleLabel, dateLabel, statusButton, priorityButton, assigneeImageView].forEach {  
    $0.isAccessibilityElement = false  
}  
  
let taskInfo = "Task titled \(data.taskTitle) is currently \(data.customStatusName)."  
let assigneeInfo = "It's assigned to \(data.assigneeName)."  
let priorityInfo = "It has \(data.taskPriority)"  
let dueDateInfo = (data.taskDueDate != nil) ? "It's due on \(data.taskDueDate!)." :  
"There's no set deadline."  
let accessibilityInfo = "\(taskInfo) \(assigneeInfo) \(priorityInfo) \(dueDateInfo)"  
self.accessibilityLabel = accessibilityInfo
```



Custom Actions

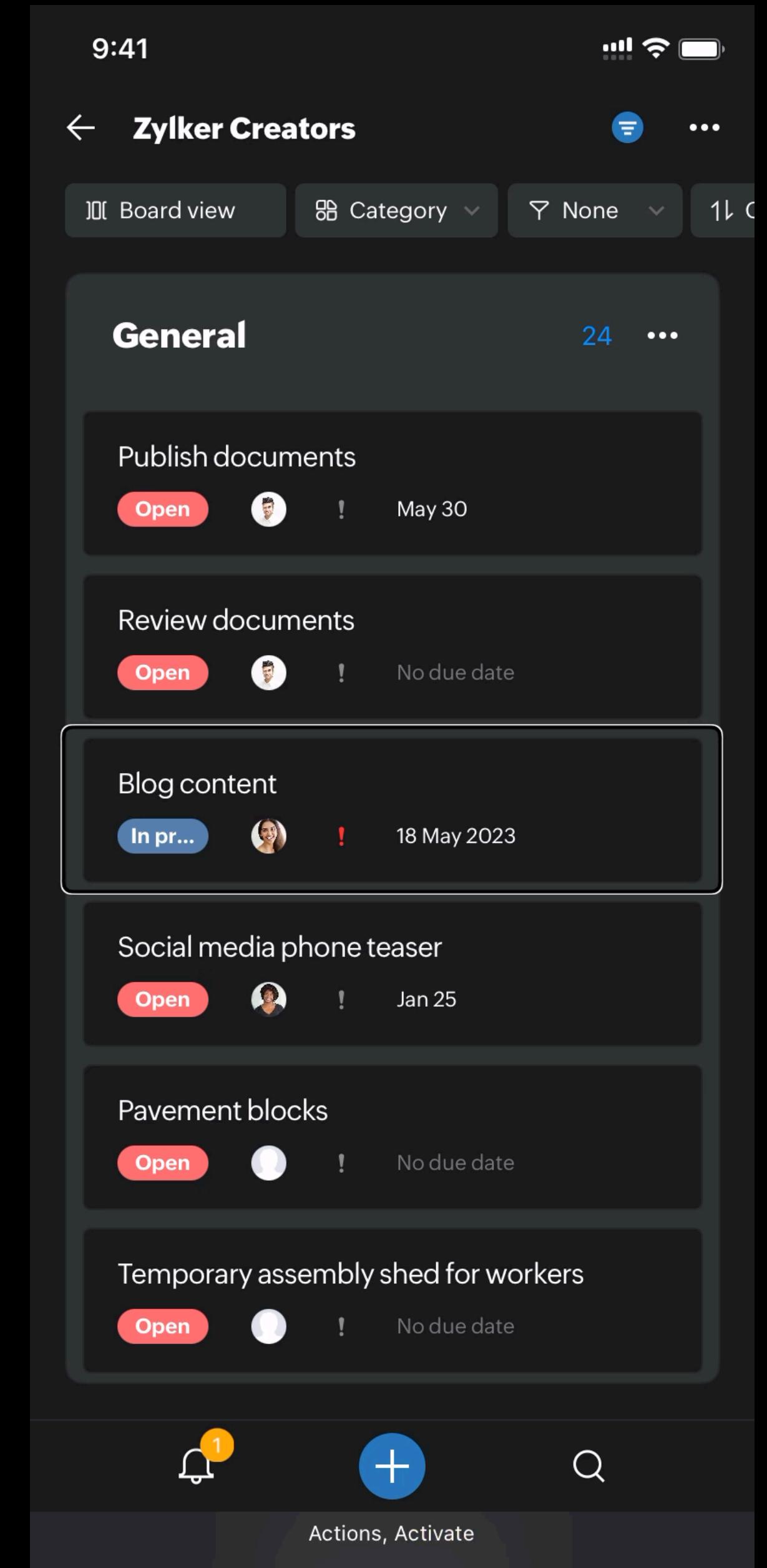
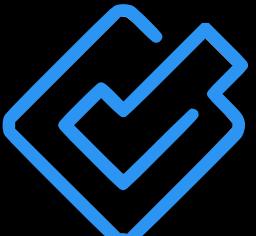


Custom actions

```
Custom Actions

let customAction = UIAccessibilityCustomAction(
    name: "Change status",
    target: self,
    selector: #selector(changeTaskStatus)
)

self.accessibilityCustomActions = [customAction]
```



Rotors



Rotors

Quick Navigation

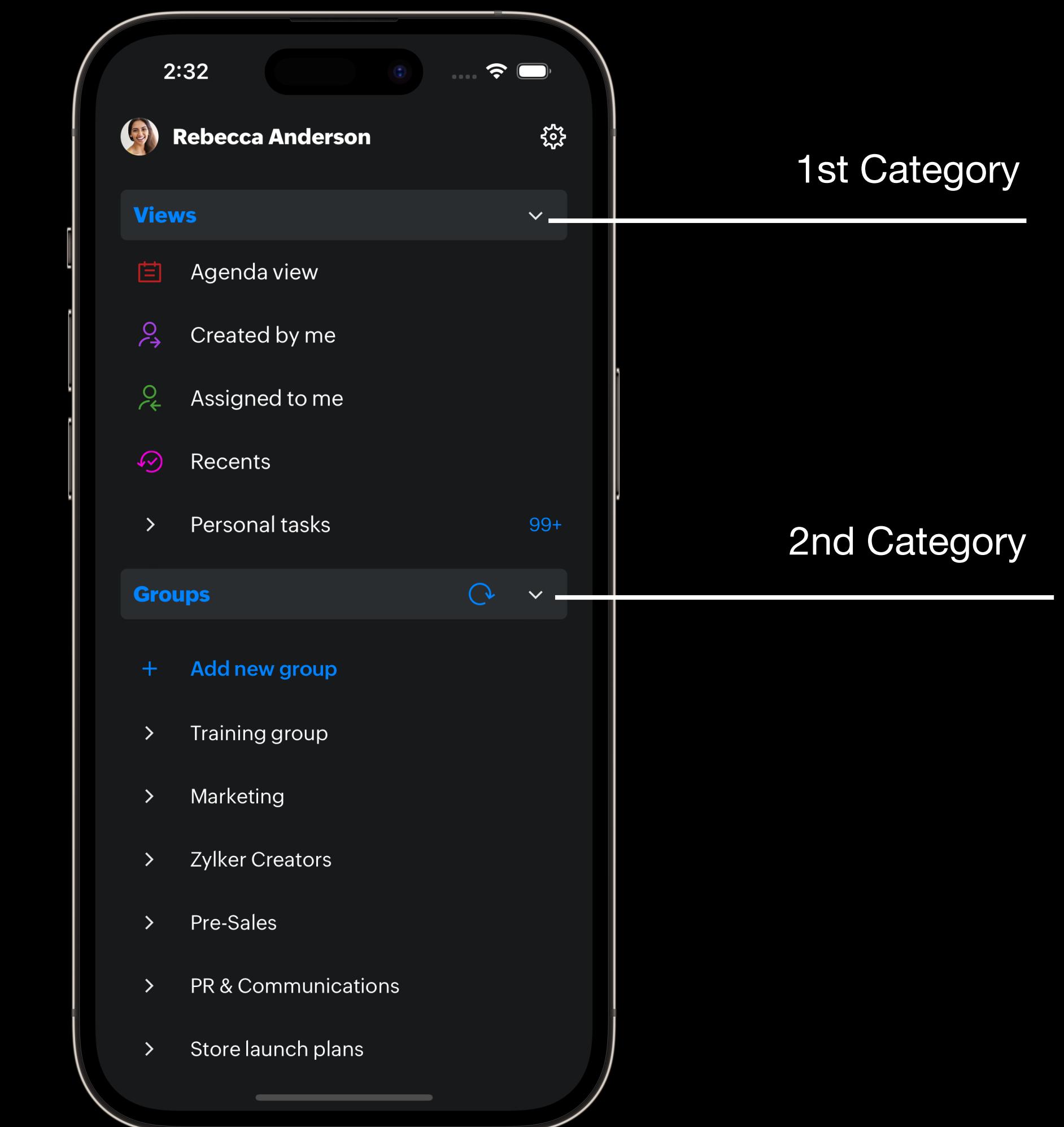
It can tell VoiceOver to only navigate by, or to, certain elements (such as headers).



Rotors

Makes use of
accessibilityTraits

myTappableLabel
.accessibilityTraits = [.header]



VoiceControl



Voice Control

Useful for people who
might have difficulty
touching or interacting with
the physical hardware of
their device

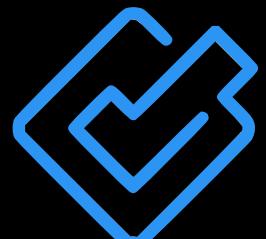


Voice Control

Names

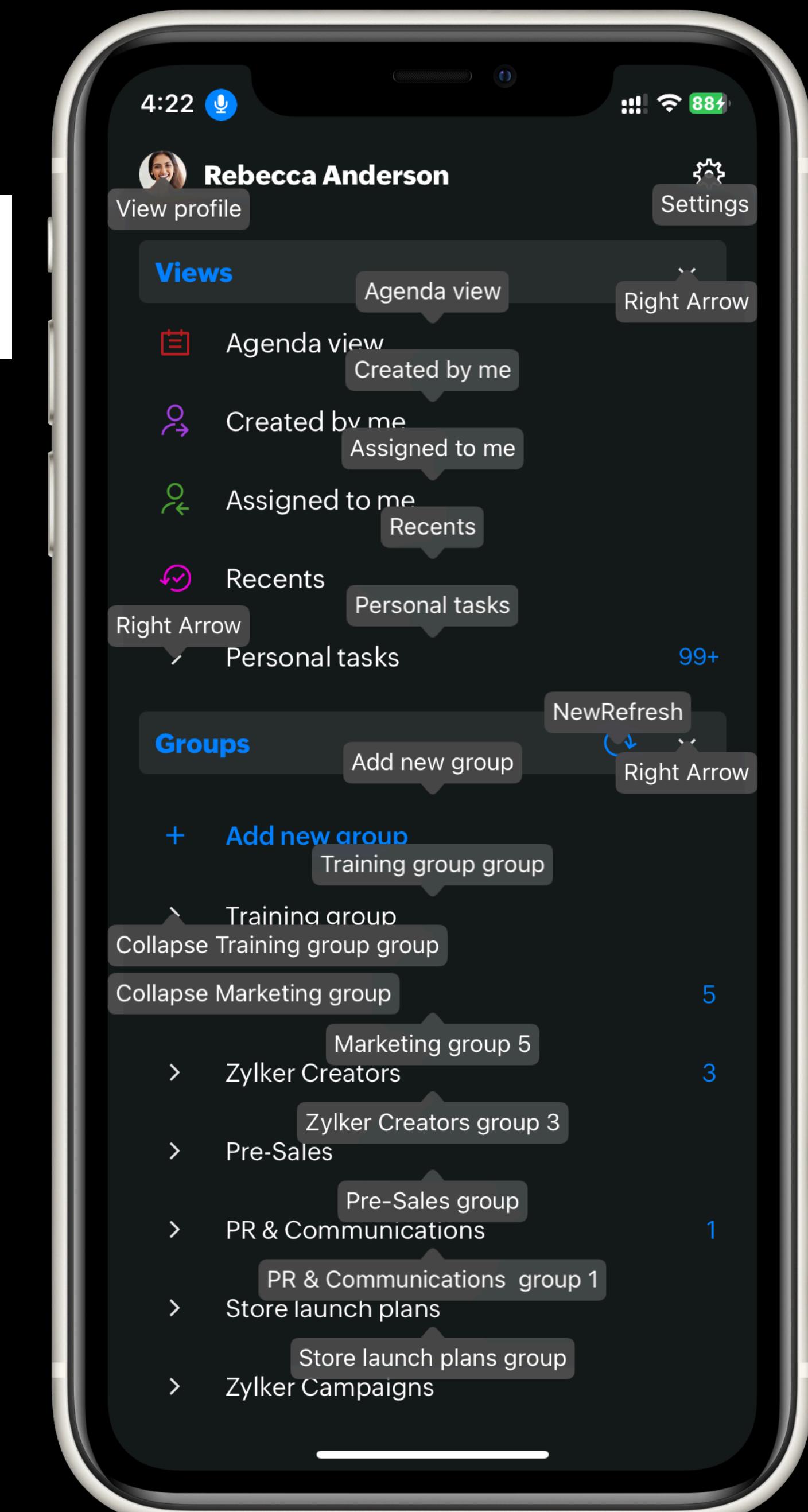
Grids

Numbers



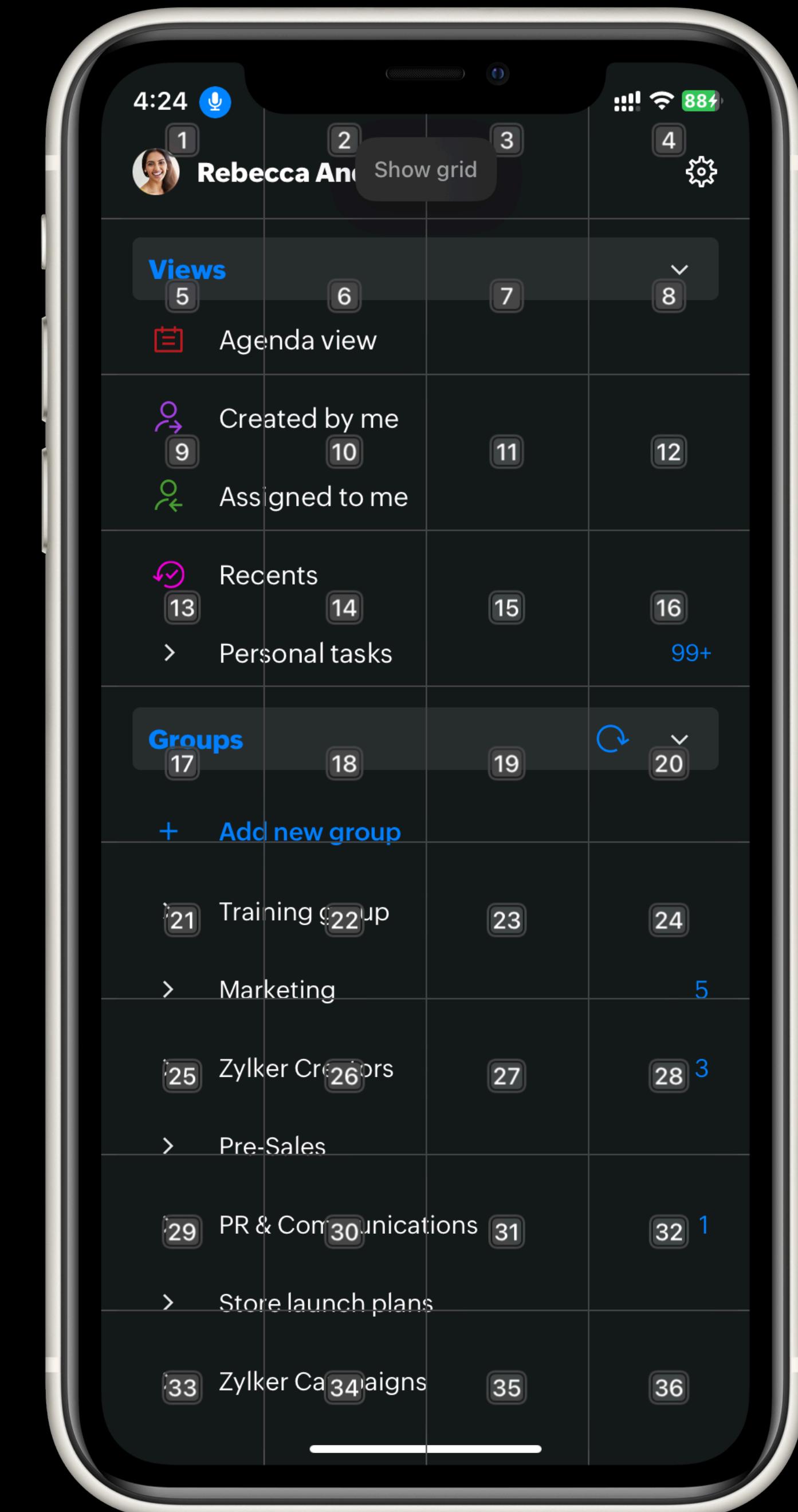
Voice Control

“Show names”



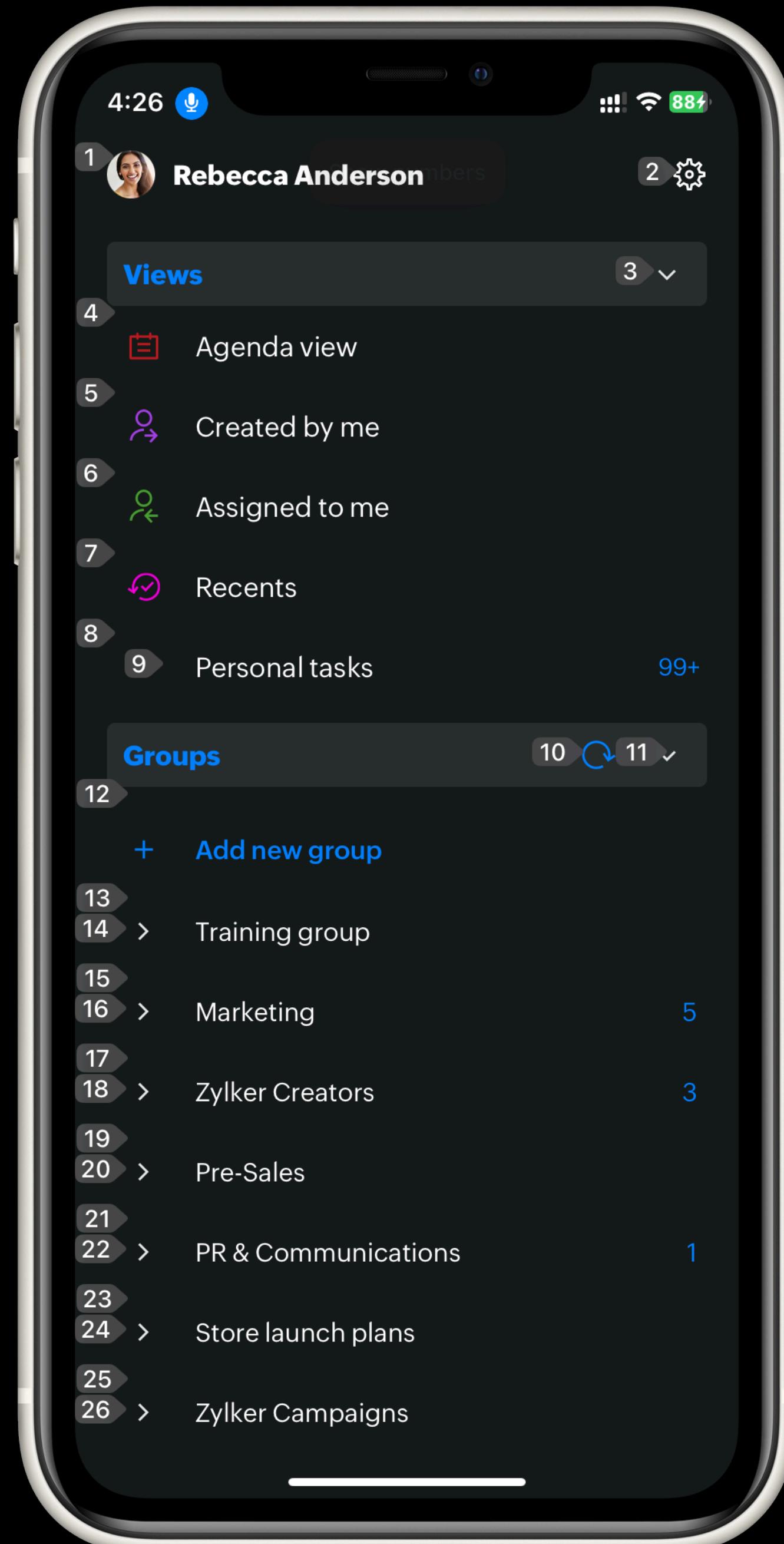
Voice Control

“Show grids”

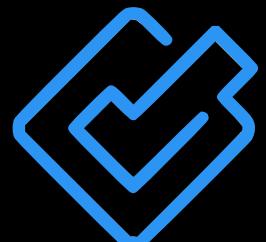


Voice Control

“Show numbers”

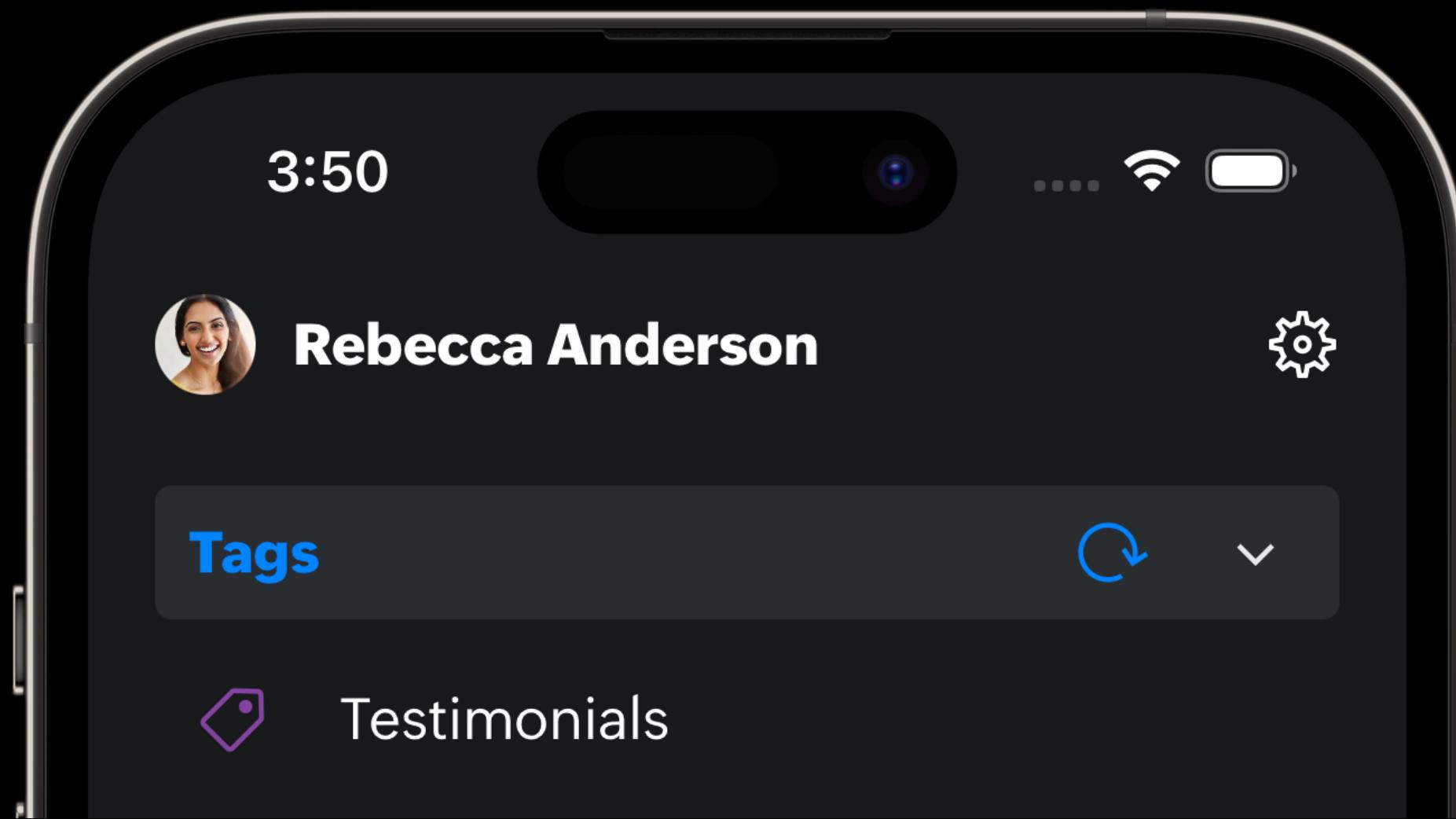


Use Descriptions for Colors

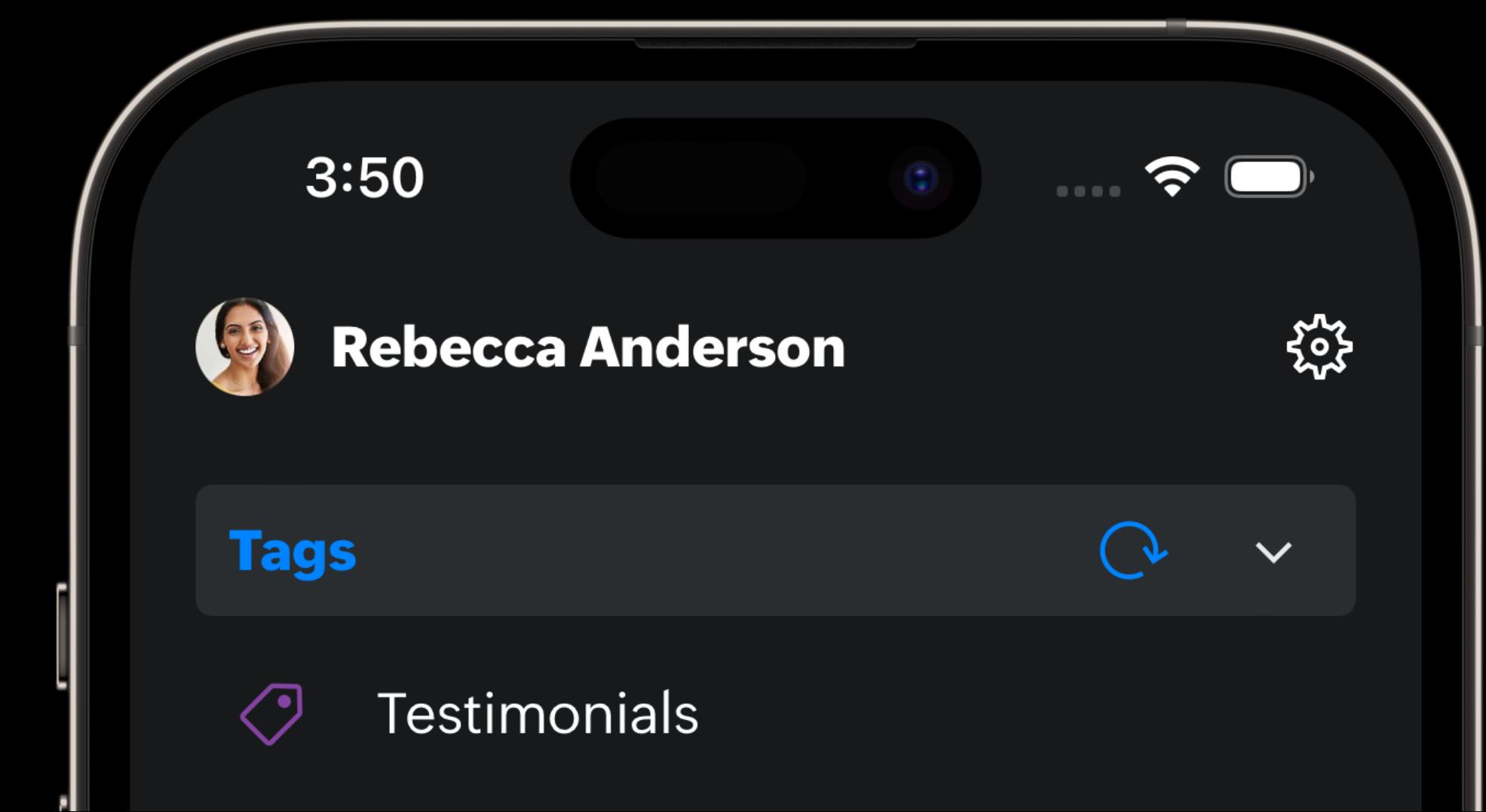


Z O H O

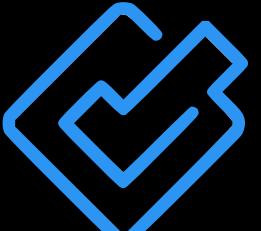
Use Descriptions for Colors



“Testimonial tag “



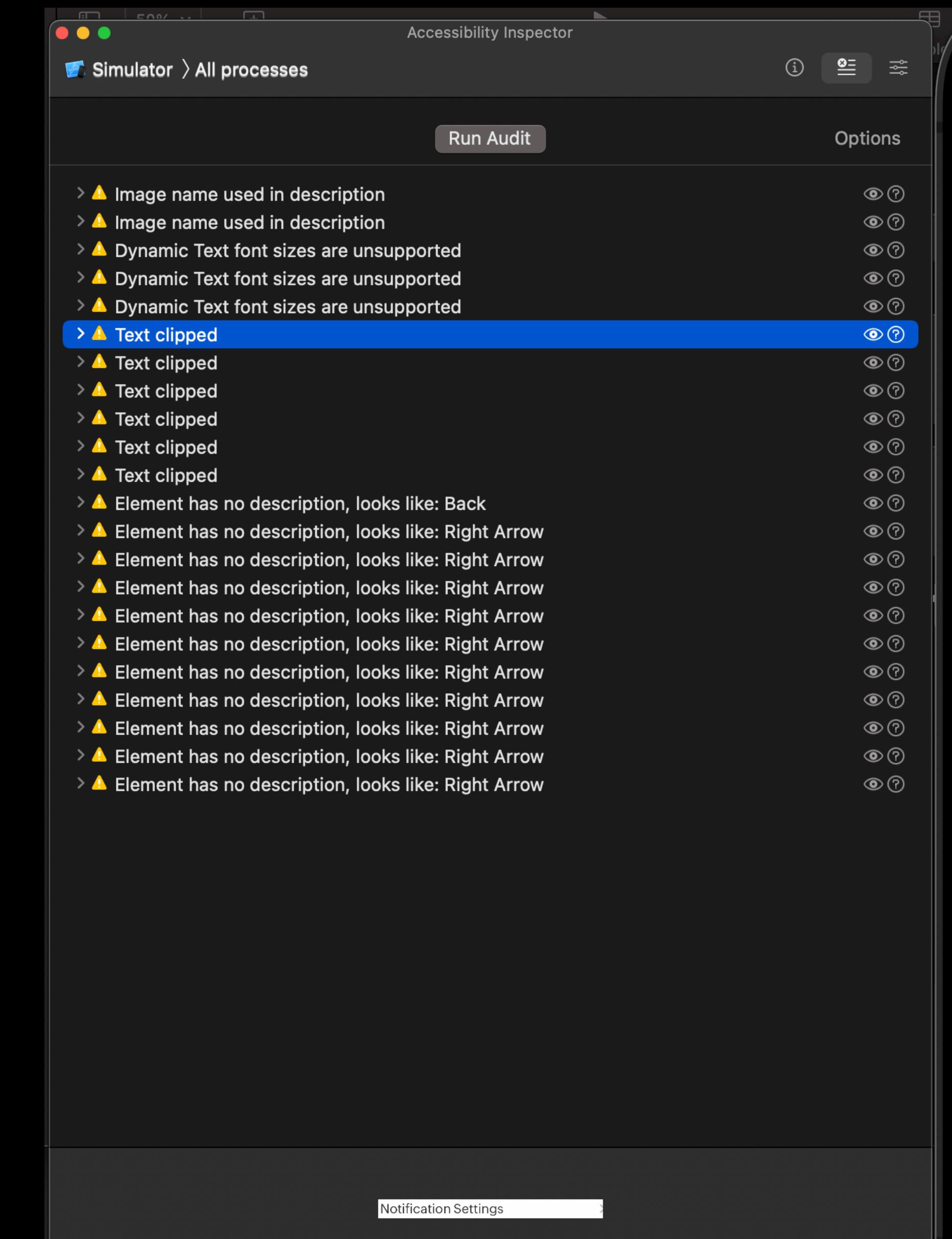
“Purple colored Testimonial tag “



Accessibility Inspector

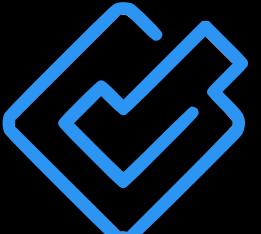


Accessibility Inspector



A recap

-
- Dynamic types
 - Accessibility properties
 - Voice over
 - Voice control
 - Rotors
 - Description for colors
 - Accessibility Inspector



Feedback Phase



**It is a repetitive
and
consistent process**



Further reading

- <https://ohmyswift.com/>
- [https://developer.apple.com/
accessibility/](https://developer.apple.com/accessibility/)



Stickers!



Reach out to me
@
bento.me/rizwan95

