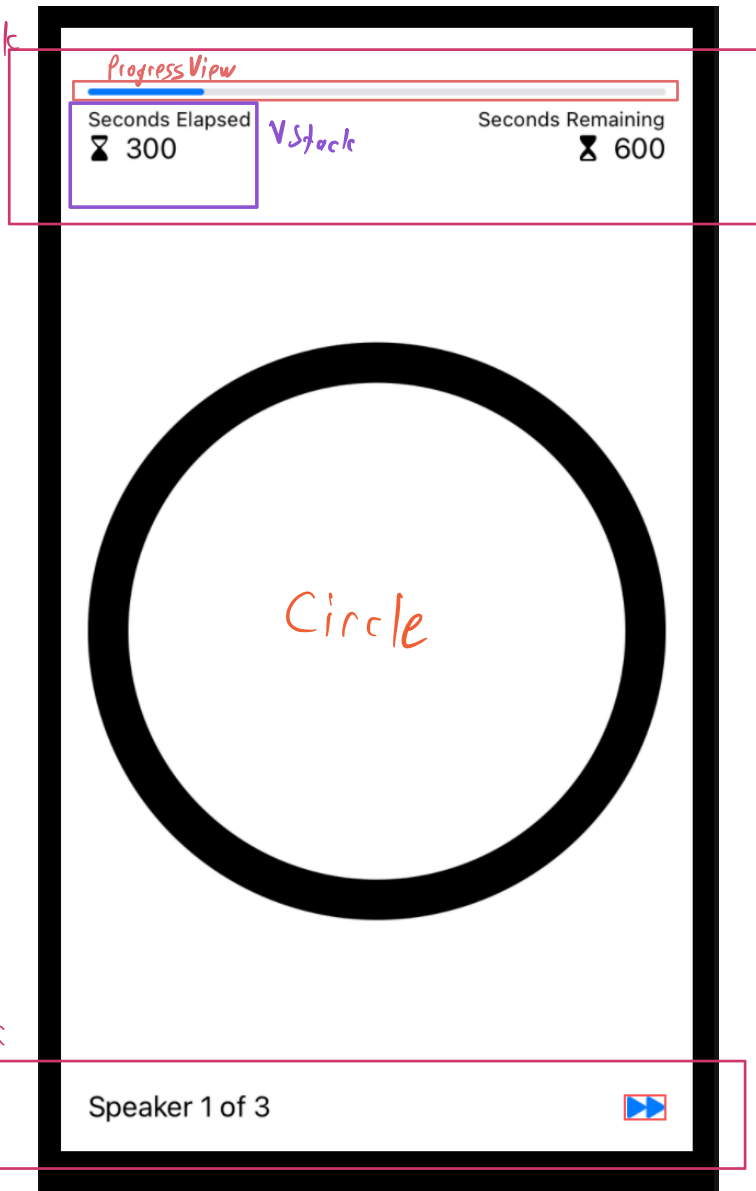
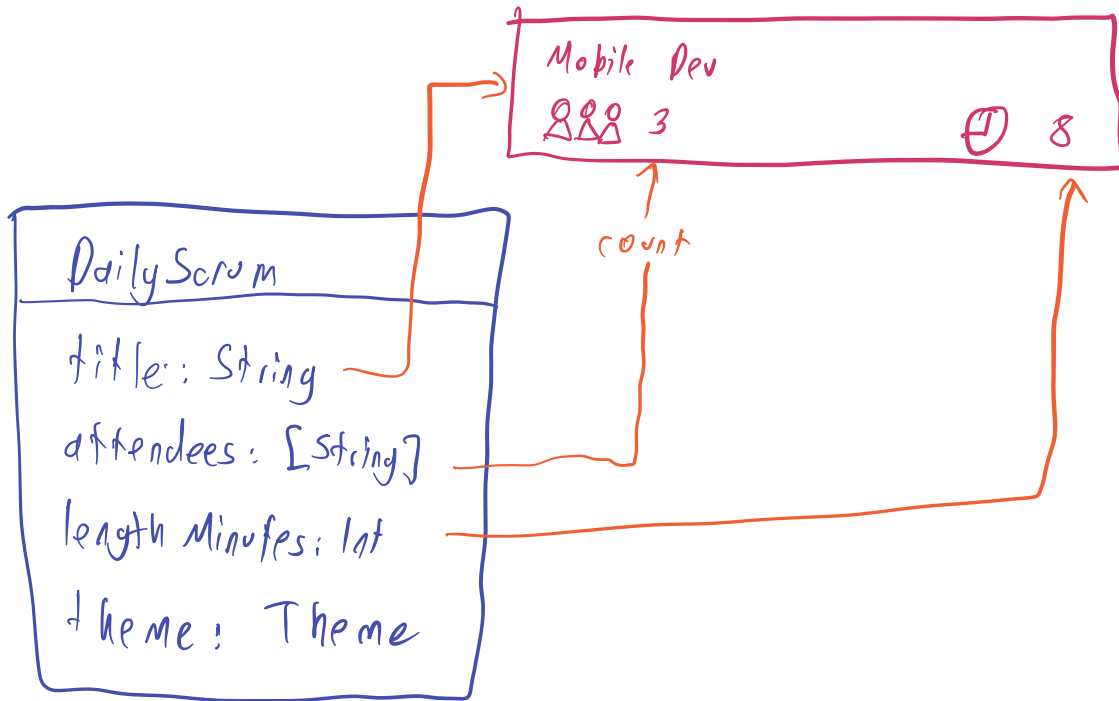


HStack



HStack

Card View

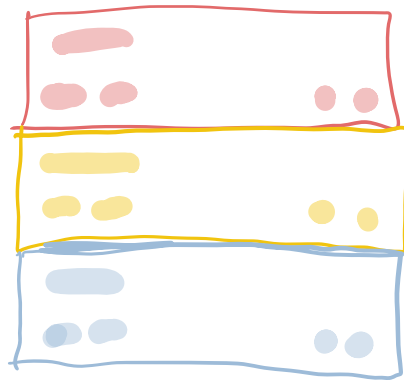


Created `TrailingIconLabelStyle`
to style our label to our liking

```
6 //
7
8 import SwiftUI
9
10 struct TrailingIconLabelStyle: LabelStyle {
11     func makeBody(configuration: Configuration) -> some View {
12         HStack {
13             configuration.title
14             configuration.icon
15         }
16     }
17 }
18
19 extension LabelStyle where Self == TrailingIconLabelStyle {
20     static var trailingIcon: Self { Self() }
21 }
22
```

Scrum View

Making a List
of CardViews

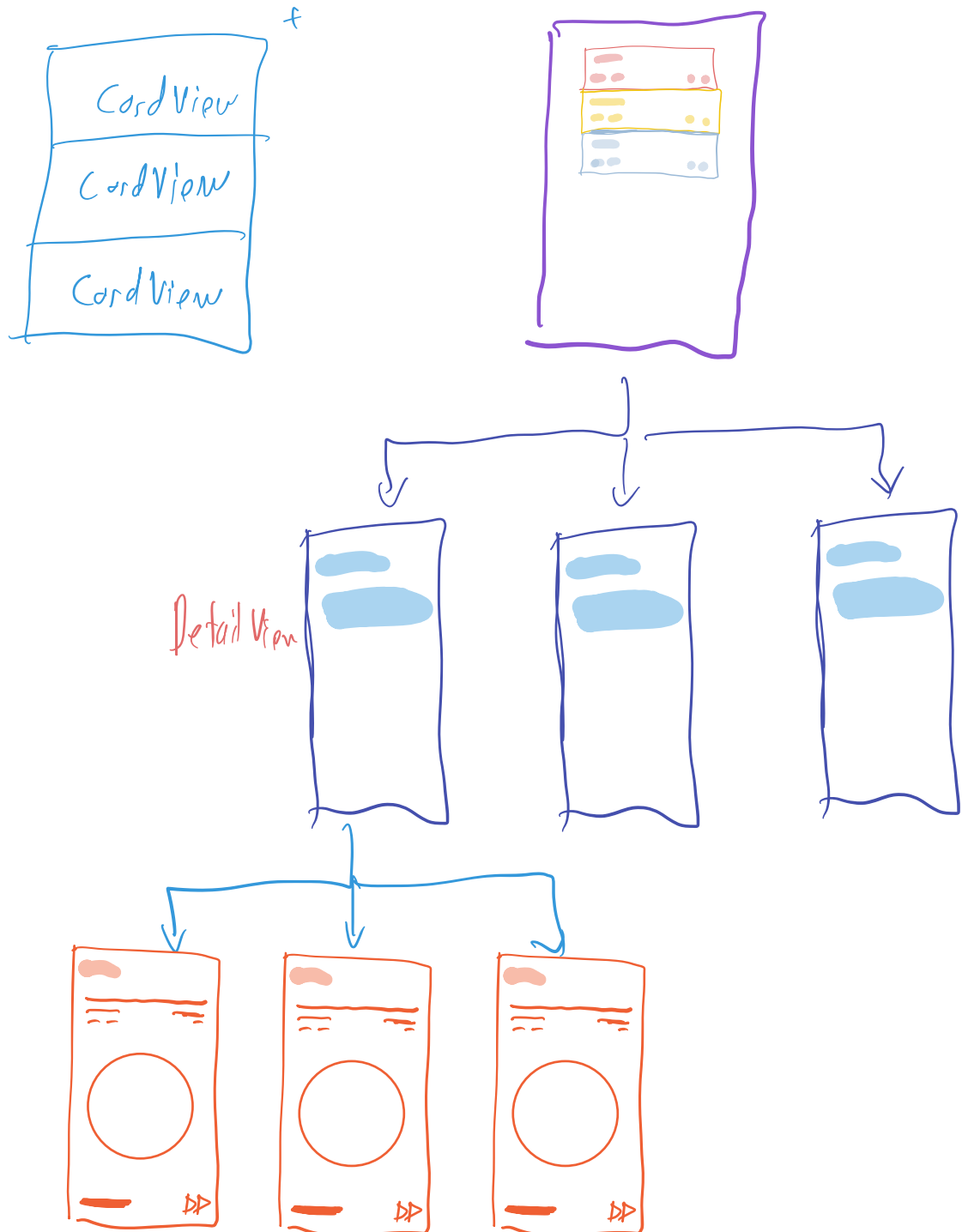


They conform to
our Daily Scrum list for Each

Identifiable library is a must for lists
since Swift generate unique ID's for each

Navigation

Navigation View



Detail View

Design → Navigation Title

Meeting Info (header)

Section

- Start Meeting Label with N. Link >
- Meeting Length Label with Spacer & minutes
- Theme Label with Spacer Poppy

padding(5)
cornerRadius(5)

List

forEach (attendees)

Section

- Sam Label with attendee.name
- Kevin
- Saul
- Ignacio
- Gus

Names have
now ID in a
new struct Attendee
so that they can
be added to
Navigation

Detail Edit View

.sheet

.cancelAction

Binded to \$isEditing

.confirmationAction

Navigation
Title

Design

Section

MEETING INFO

Title

TextField



5 minutes

Slider and Label
with Space

Attendees

Section

Anna

Jane

New Attendee

ForEach(Attendees)

TextField

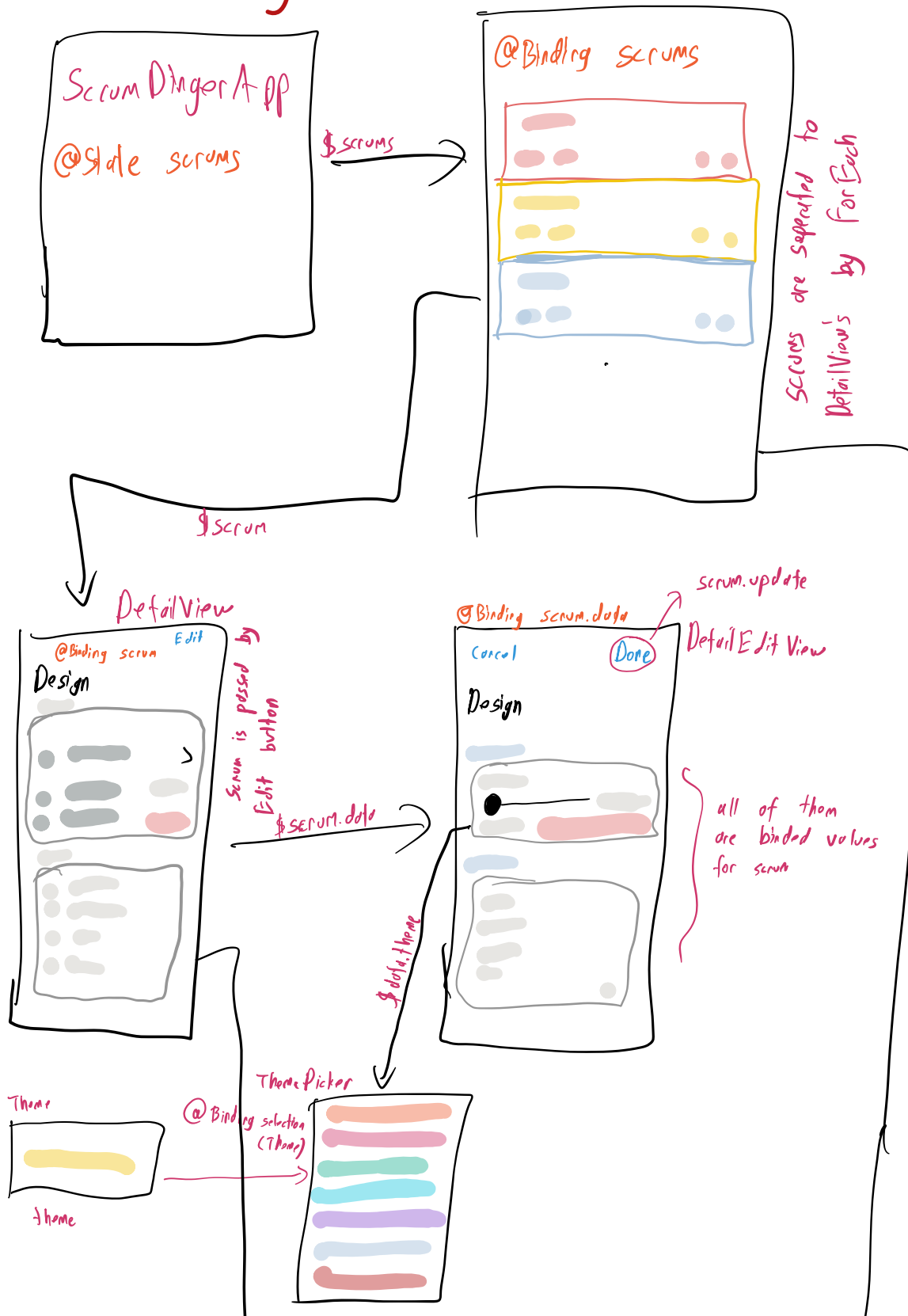


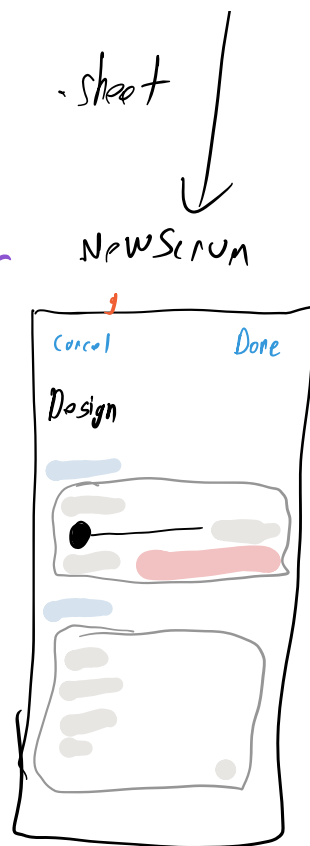
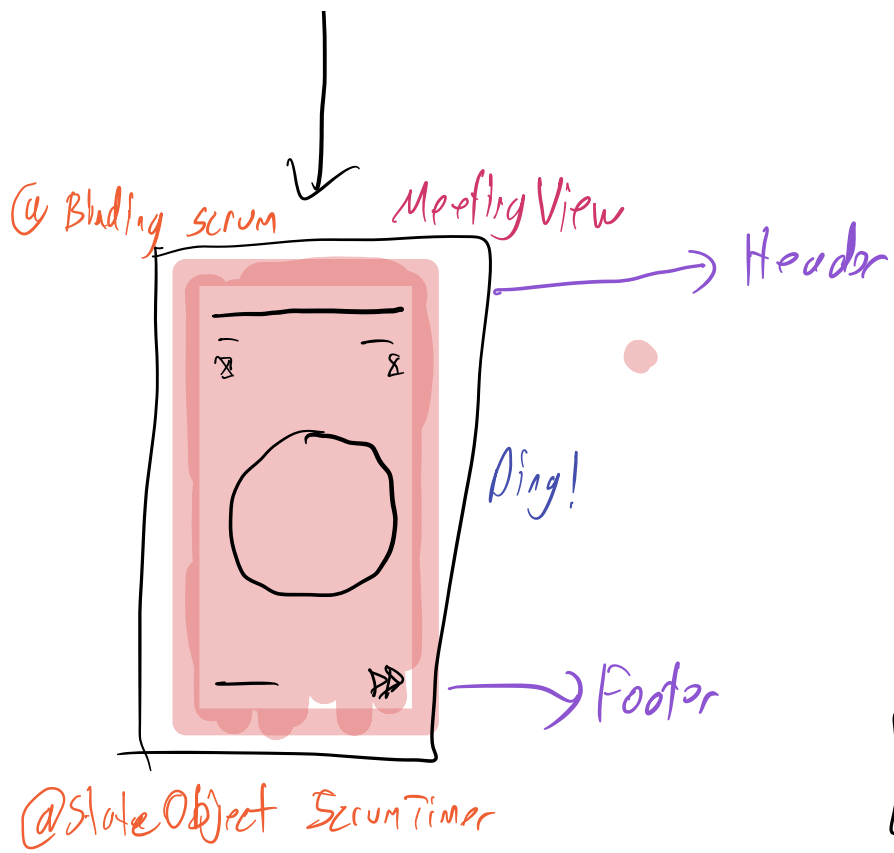
Button
with Animation

@State NewAttendee

binded to TextField

Data Passing





Persisting Data

Screen Store

Loading

Get data from `screens.data`

Dispatch Queue, main, async Make data process in
the background

load either with existing data
or new array

Saving

encode screens and write to `outFile`

complete it by passing count

onAppear

load screens to `store.screens`

Meeting Timer View