

# App Design and User Experience

Lecture 11

**\*Please sit with your project team!\***

CIS 1951

# Last time, in CIS 1951...

## UIKit & UIKit Integration with SwiftUI

- UIKit basics: MVC
- User interaction in UIKit: event and input management
- Integrating UIKit in SwiftUI: using `UIViewRepresentable`
- Combining UIKit & SwiftUI: navigation and data sharing strategies
- **Questions? Comments? Feedback?**

# CIS 1951 as a whole

Lectures 1-6: The Basics

Lectures 7-10: Technologies

Lectures 11-13: Beyond Development

# The App Design Process

# How do we get an app from scratch?

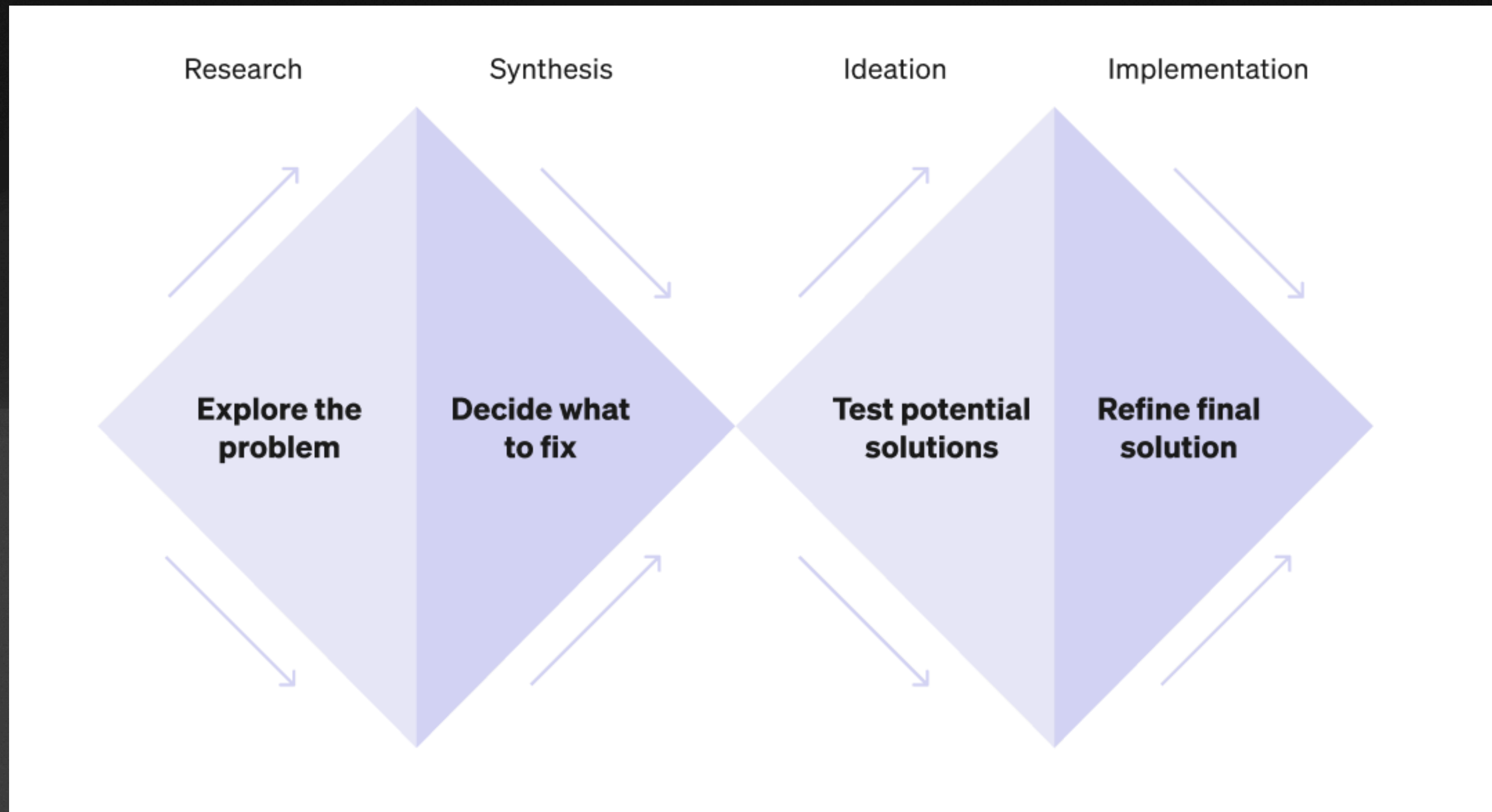
What do we need to know?

- **Problem:** What problem do we want to solve?
- **Solution/Features:** How do we solve it?
- **UI/UX:** How will people access/use our solution?
- **Implementation:** How do we build our solution?

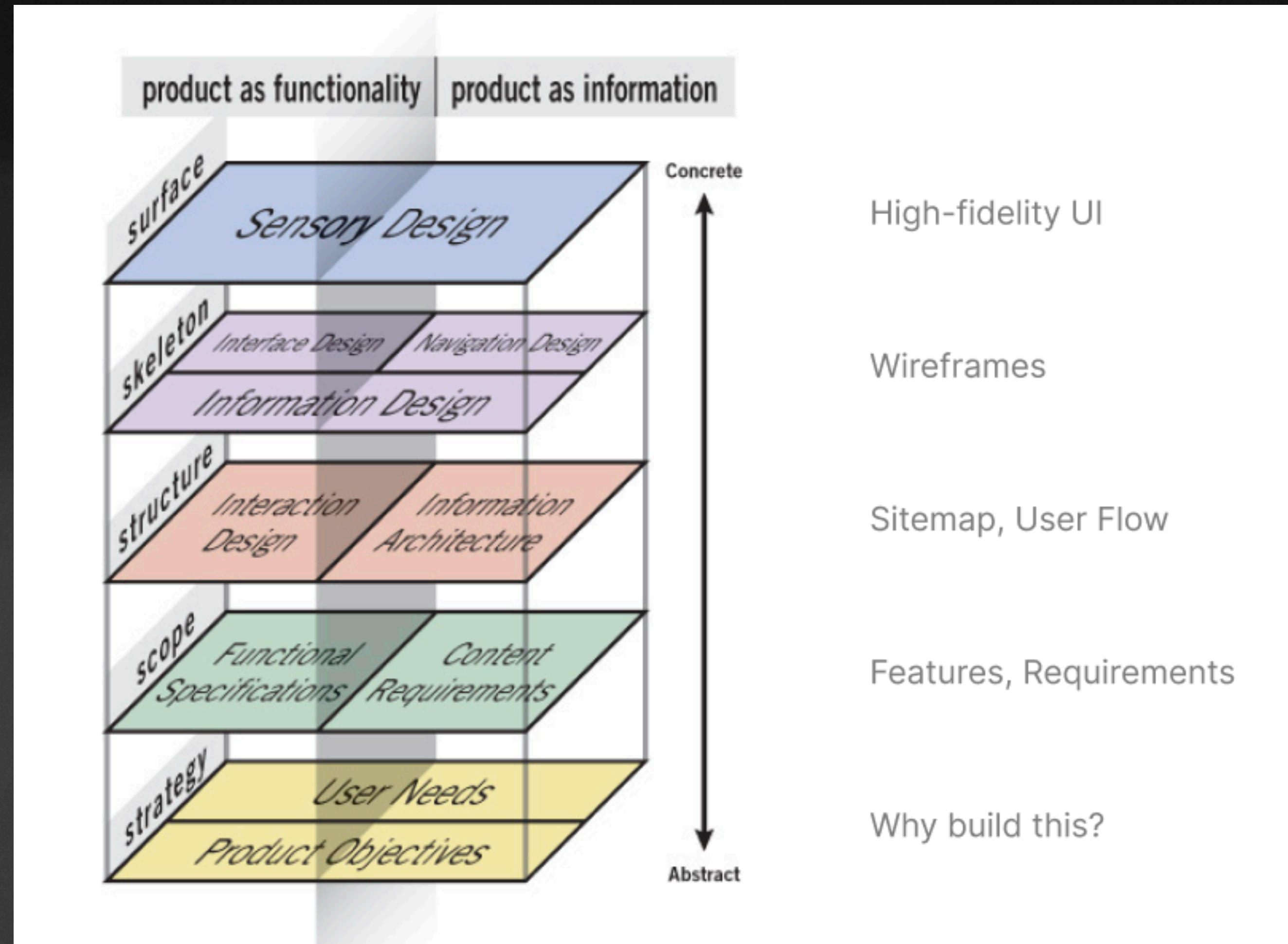
# What is UI/UX?

- **UI** = User interface, aka what the user **sees** on our app
- **UX** = User experience, aka how the user **interacts** with our app

# Design Thinking

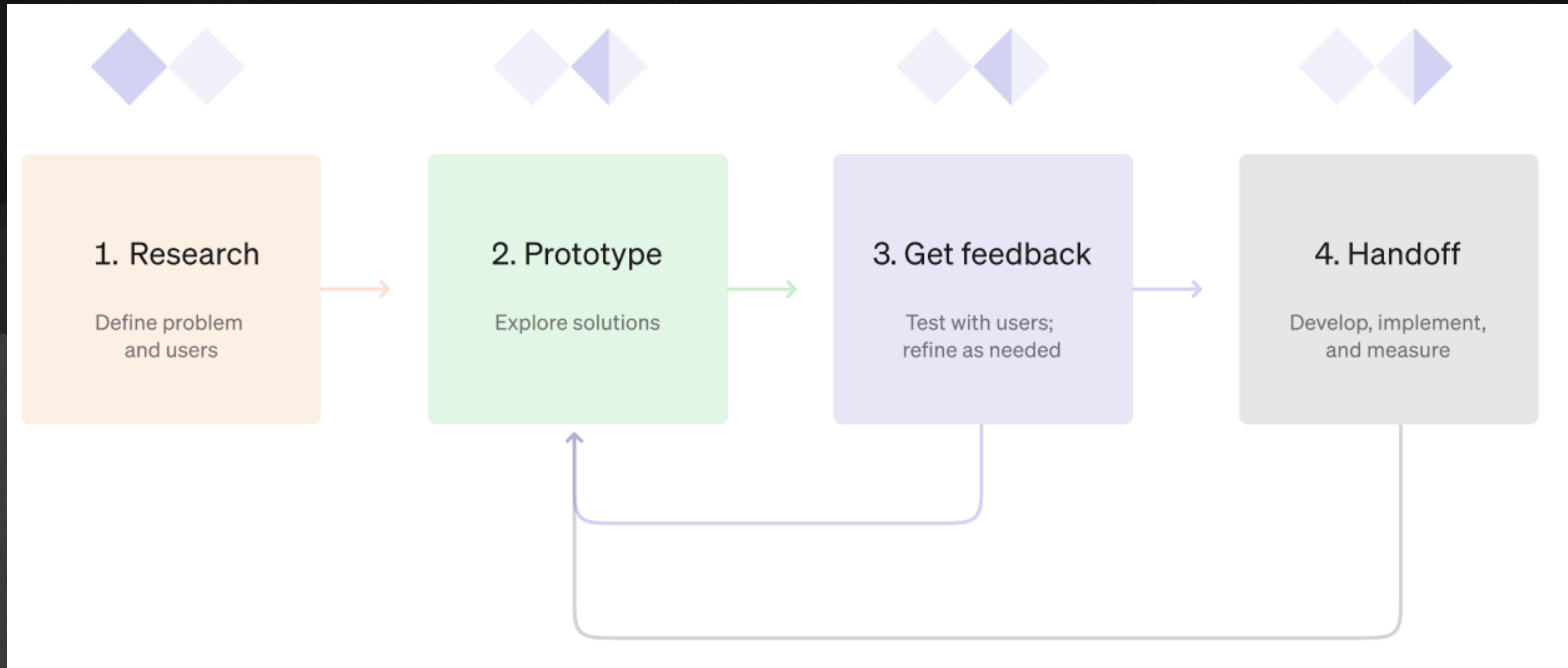


# The Planes of UX





# The UX Process



# The App Design Process

## Steps

- User Stories
- Low-fi Sketch
- User Flow
- View Hierarchy Diagram
- High-fi Sketch

# User Stories

# User Stories

## Definition

**“Brief, informal explanations of software features written from the perspective of the end user”**

# User Stories

## Structure

“As a [persona], I [want to], [so that].”

# User Stories

## Examples

- Consider our HW4 weather app:
  - “As a frequent traveler, I want to quickly check the weather forecast for multiple cities I plan to visit, so that I can pack appropriately and make informed travel arrangements.”
  - “As a gardening enthusiast, I want to monitor the weather conditions of my local area and save historical weather data, so that I can plan my gardening activities based on past weather trends and upcoming forecasts.”

# User Stories

Your Turn!

**Generate 5 user stories for your project app.**

# User Stories

Your Turn!

**Pick 1 to share!**

**Say your app idea, then your selected user story.**



# Low-fi Sketch

# Low-fi Sketch

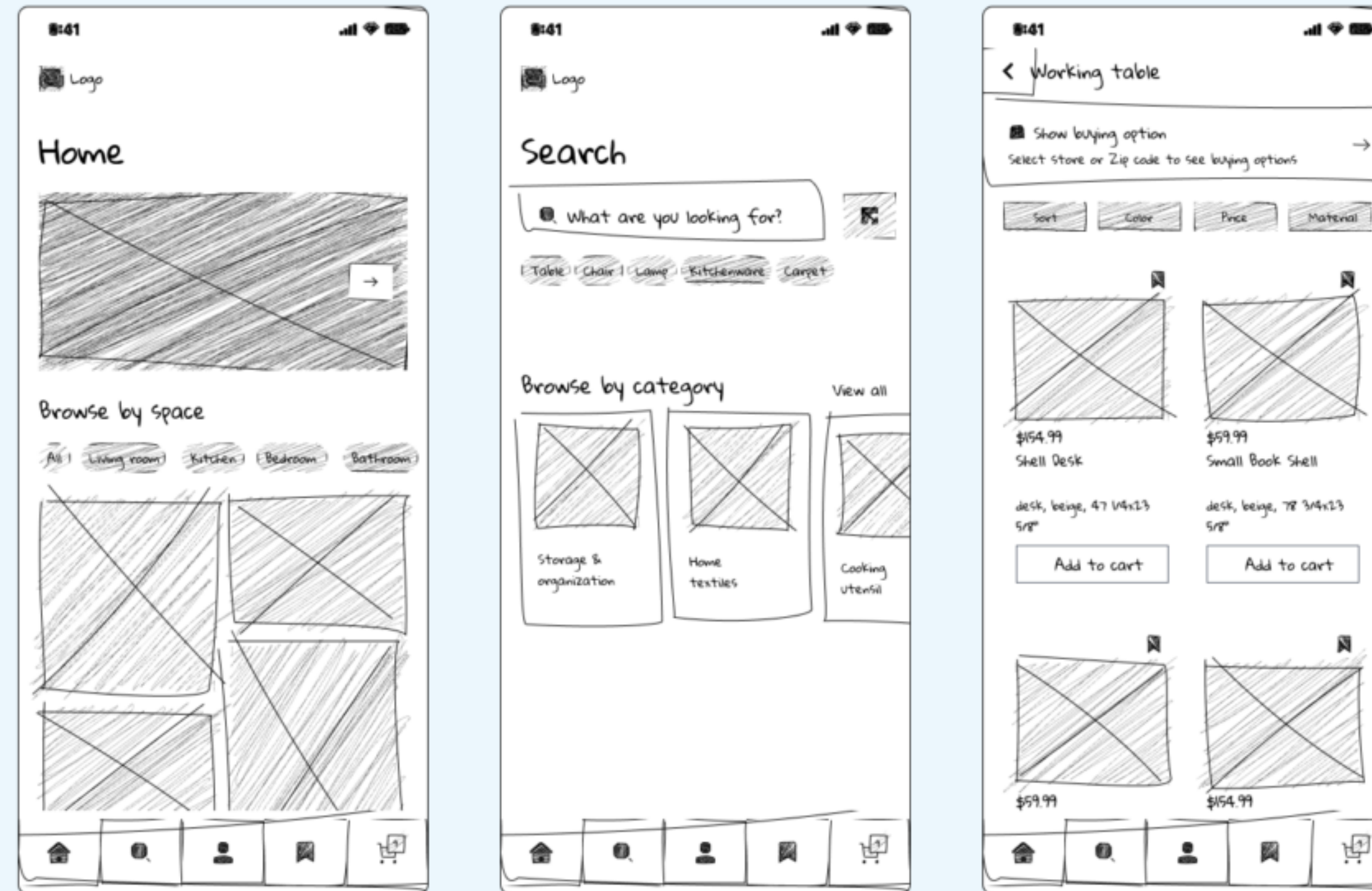
## Definition

**A rough drawing or skeleton of how your app will look and work.**

\*Sometimes also called a “wireframe”

# Low-fi Sketch

## Example



# Low-fi Sketch

Your Turn!

**Use pencil and paper, sketch a wireframe of your app.**

Which screens do you plan to have?

What's on each screen?

# Low-fi Sketch

Your Turn!

**Pass your sketch to your neighbor.**

**Write down what you think your neighbor's app does based on the sketch you see.**

What features can you see?

# User Flow

# User Flow

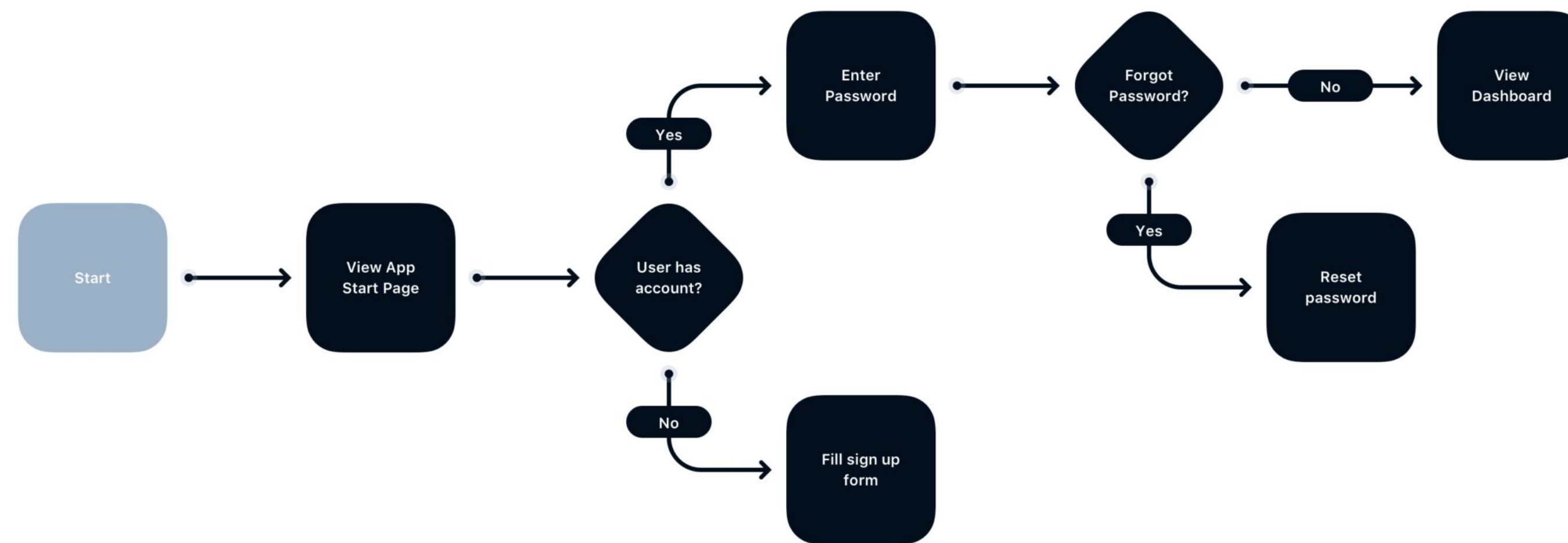
## Definition

**A visualized path that the user follows through an app to complete single or multiple tasks.**

# User Flow

## Example

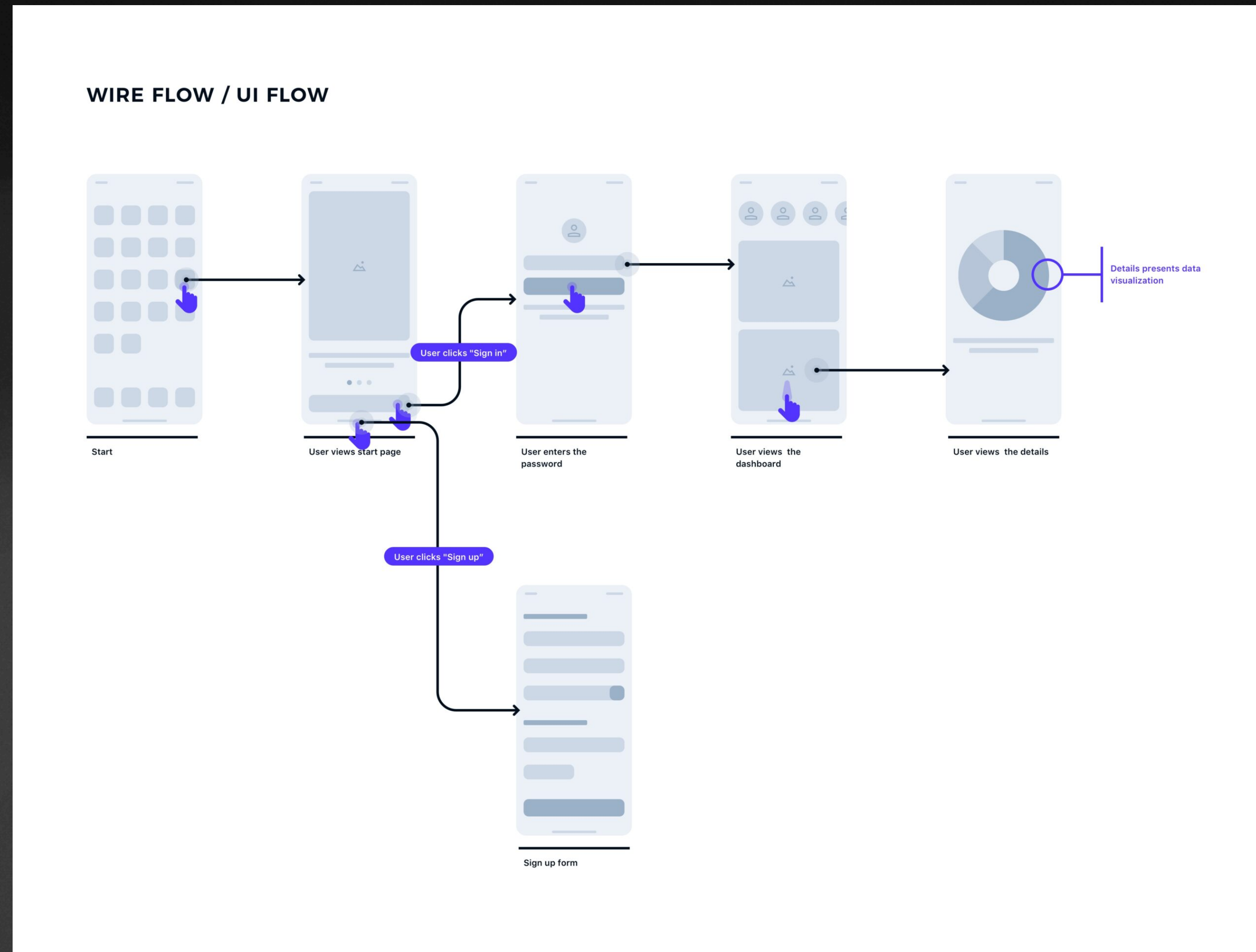
### TASK FLOW





# User Flow

## Example



# User Flow

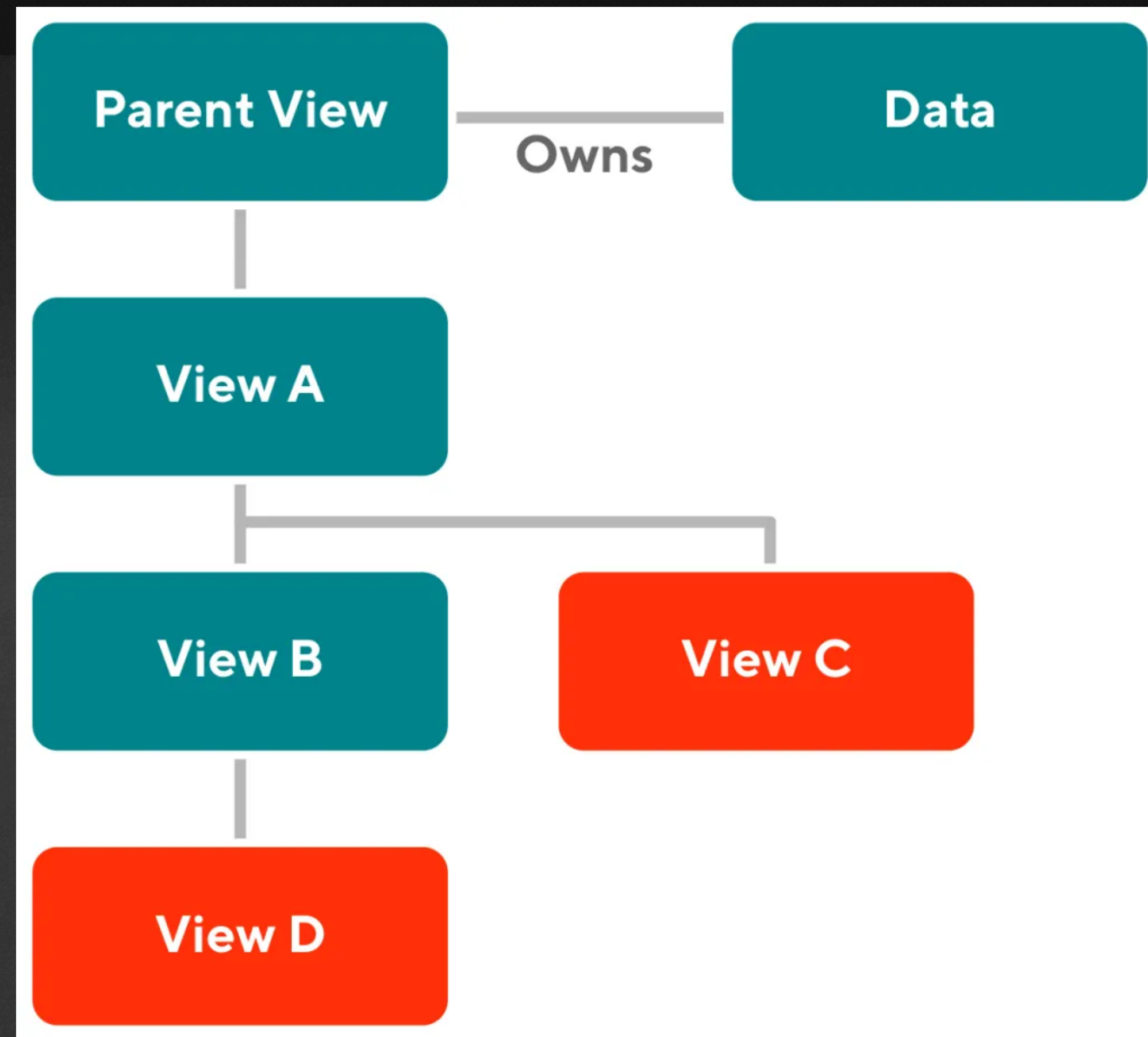
Your Turn!

**On your low-fi sketch, draw at least 3 user interactions using arrows across components.**

# View Hierarchy Diagram

# View Hierarchy Diagram

Review from Week 4!



# High-fi Sketch

# High-fi Sketch

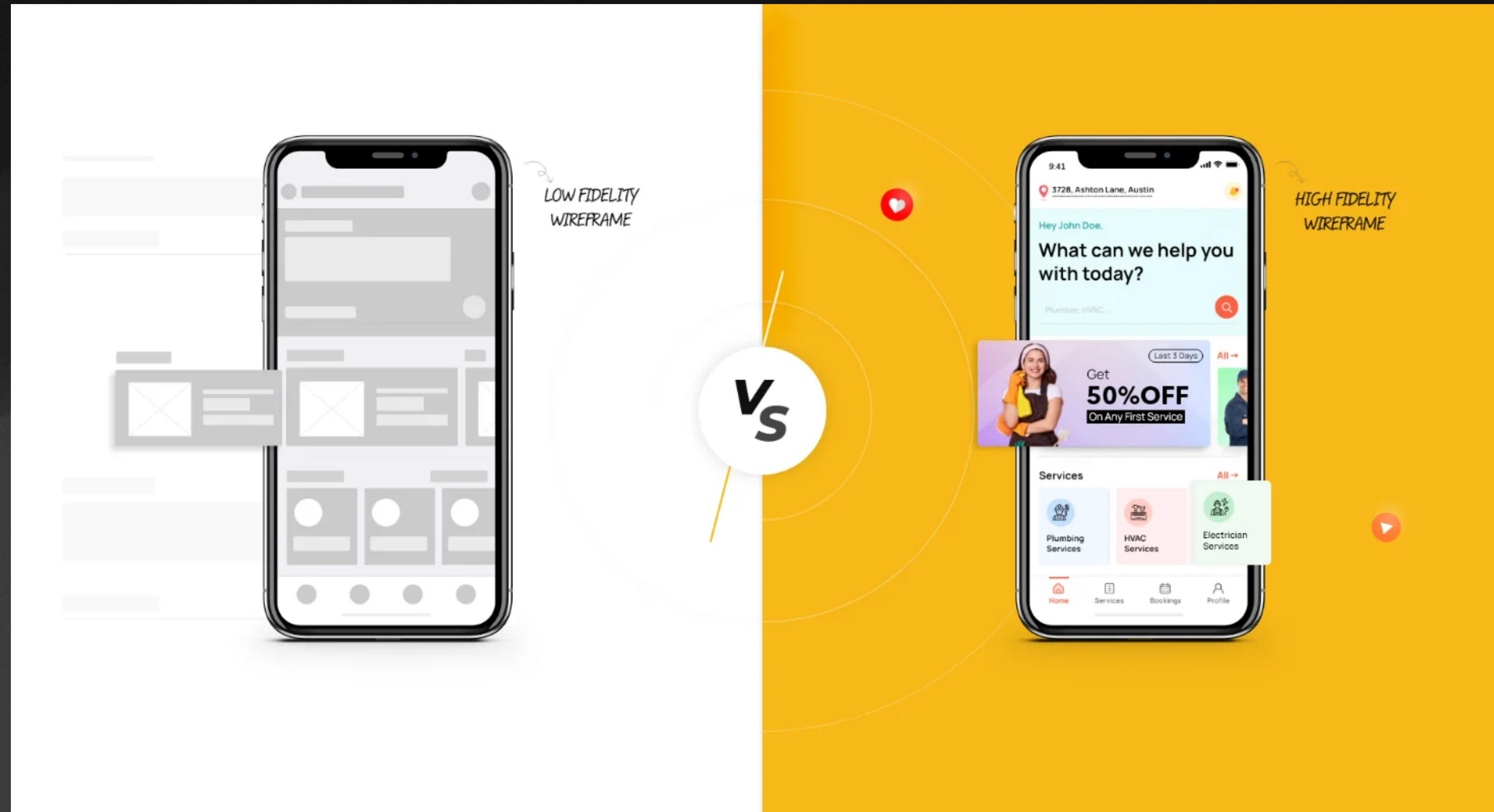
## Definition

**A realistic, interactive prototype that closely resembles the final design of a project.**

\*Sometimes also called a “high-fi wireframe” or “prototype”

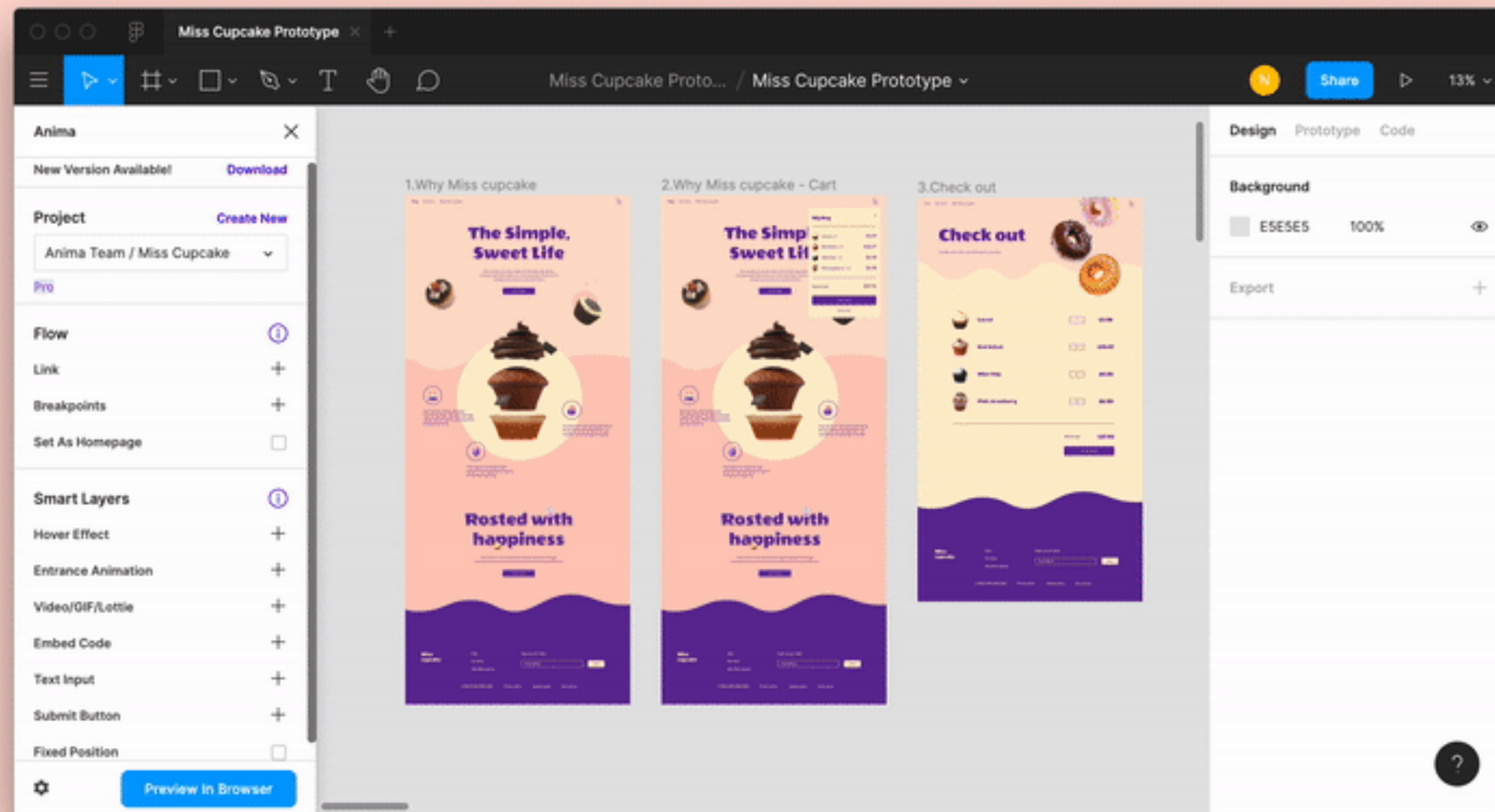
# High-fi Sketch

## Example



# High-fi Sketch

Tool: Figma



- Draw
- Prototype interactions
- Simulate
- Access community assets/tools



# UI/UX Design: Starter Tips

# **1. Visibility of system status**

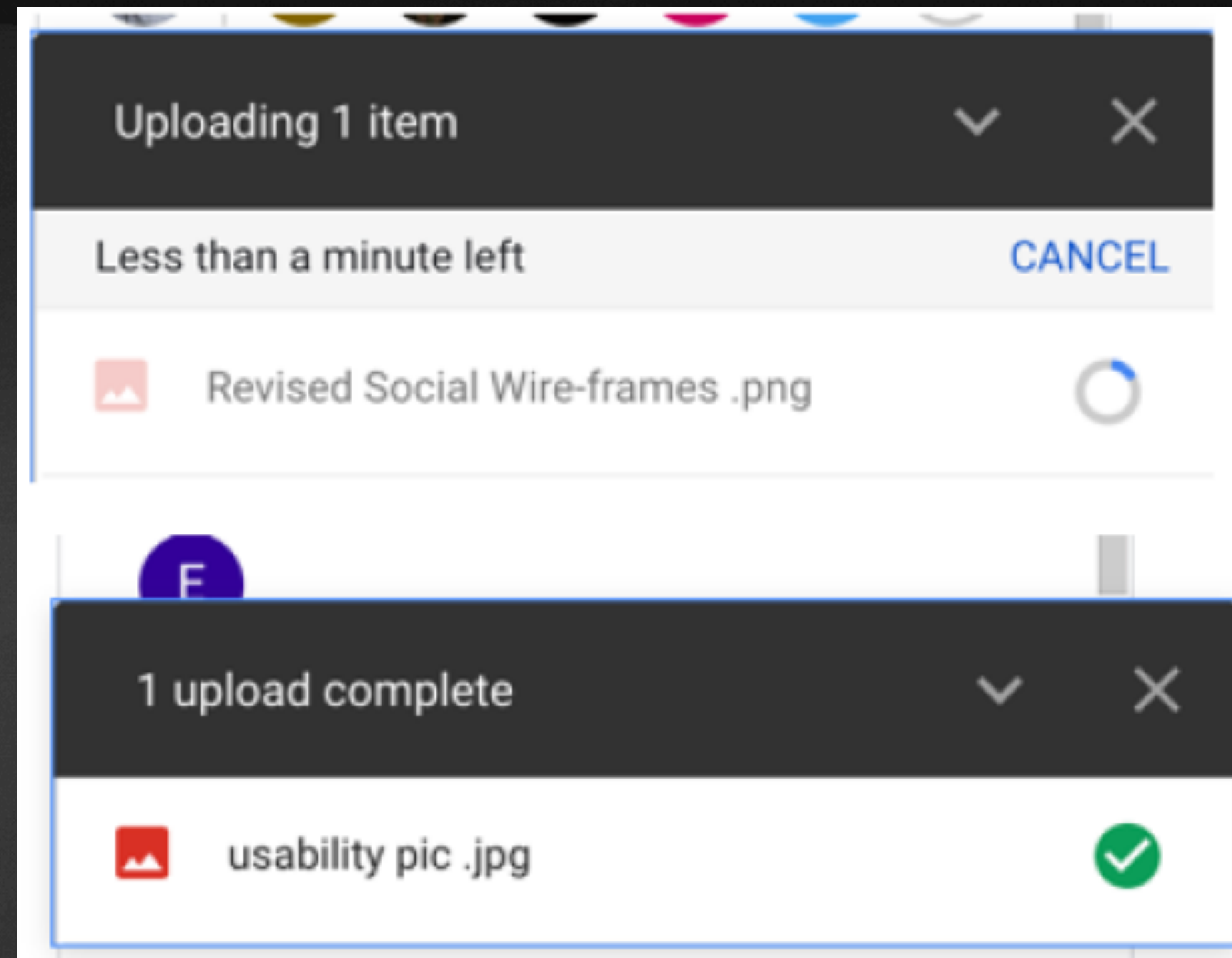
# **Visibility of system status**

**Keep users informed about what's going on.**

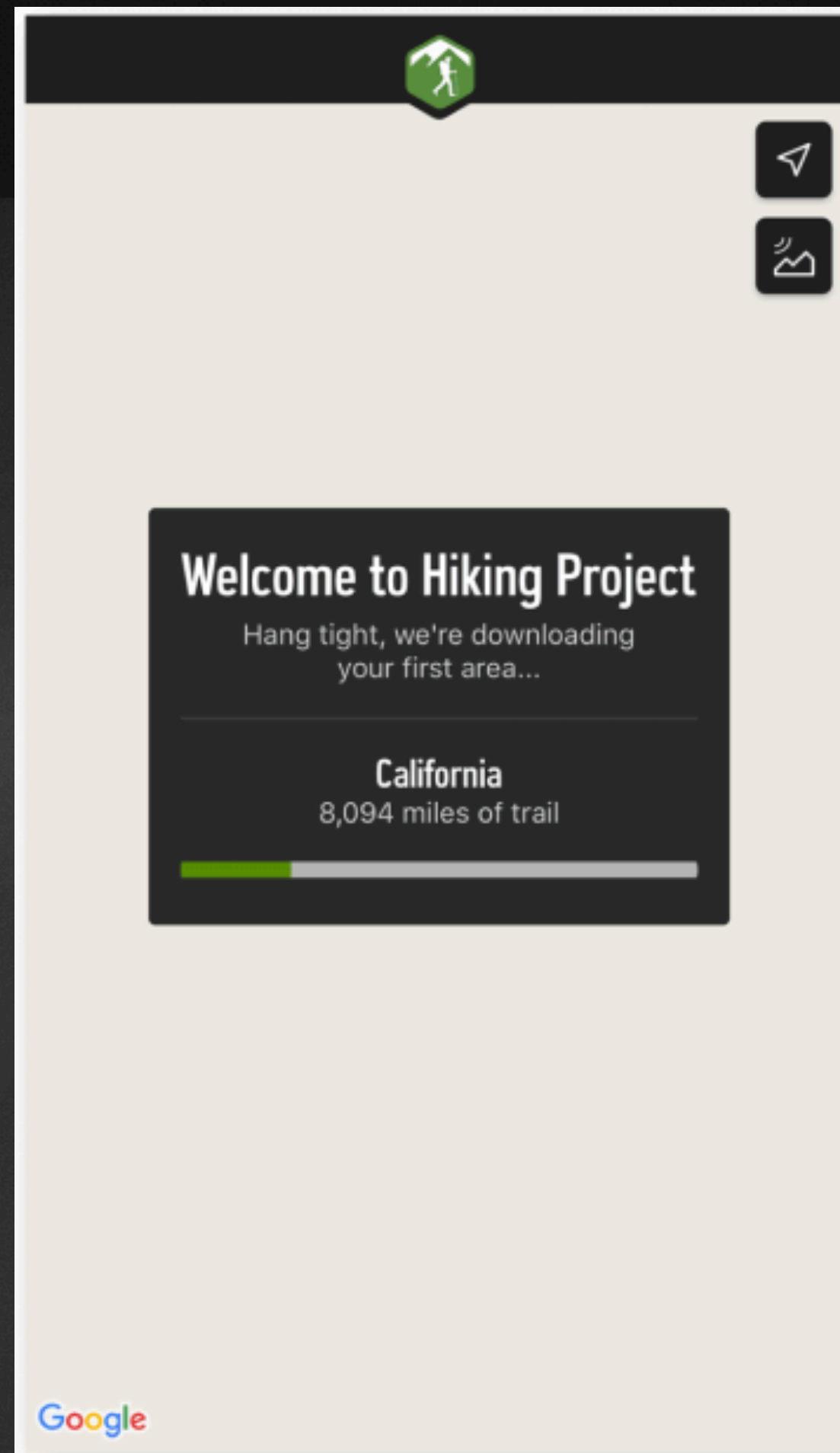
# Visibility of system status



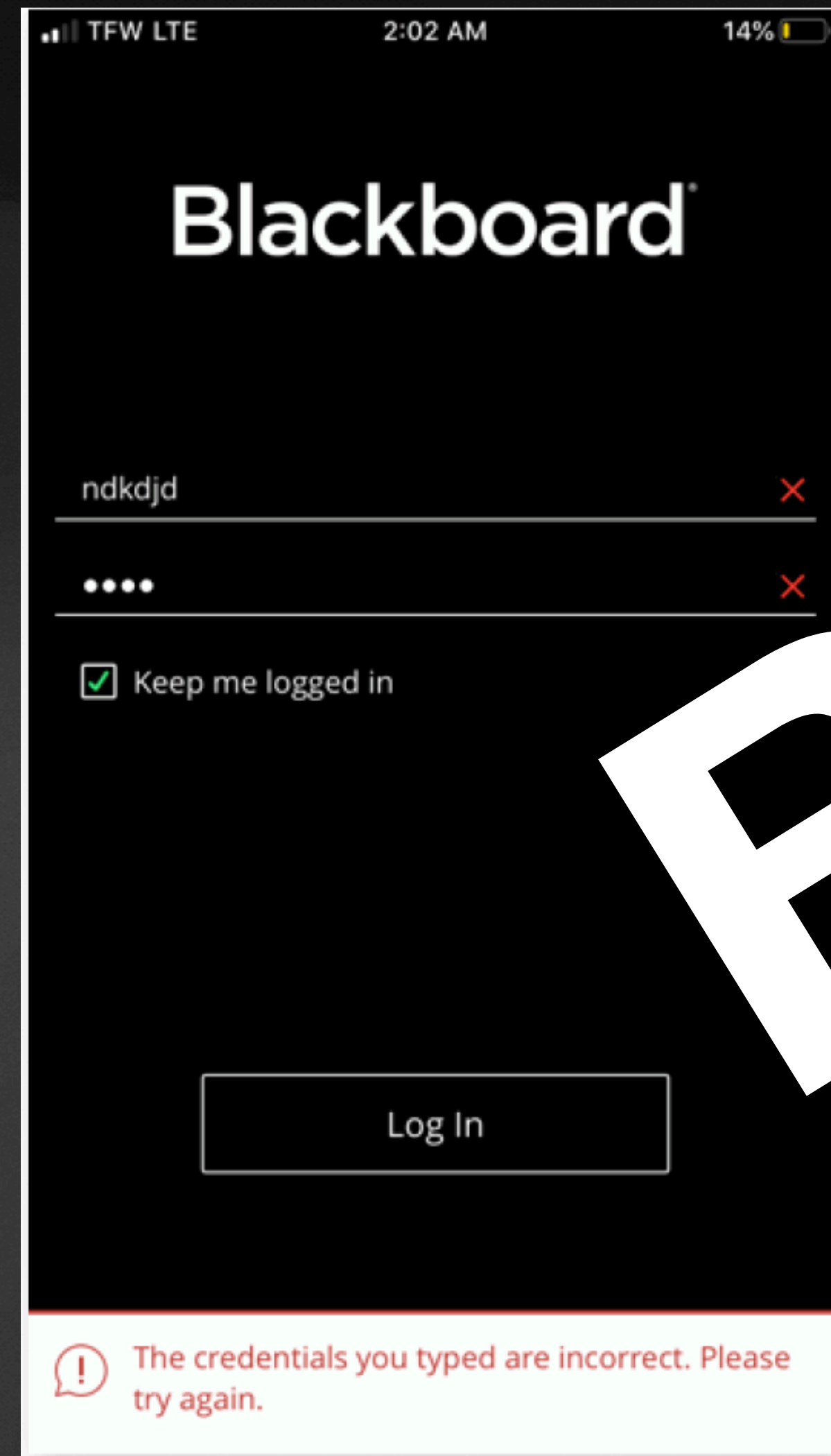
# Visibility of system status



# Visibility of system status



# Visibility of system status



**BAD**

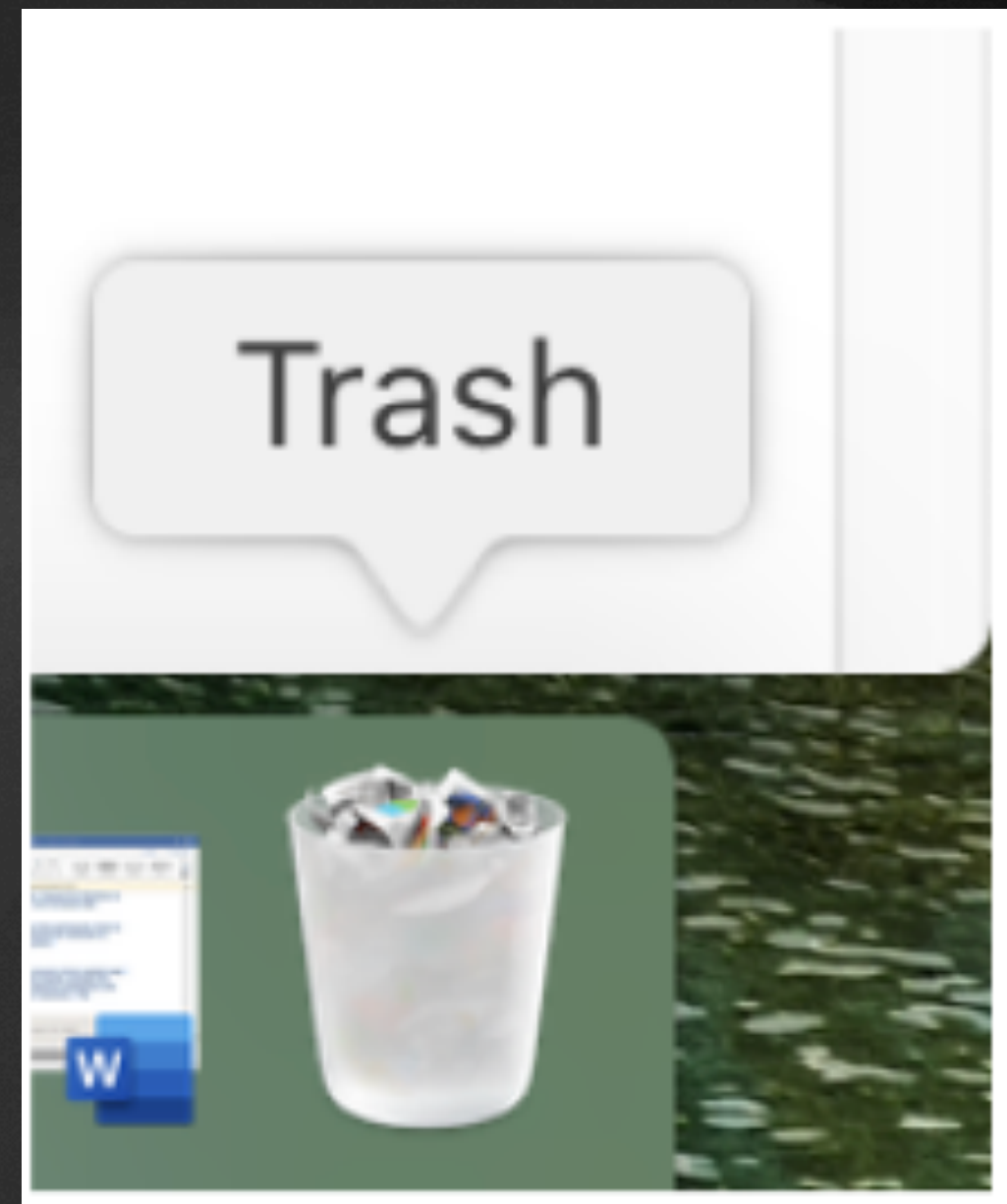
## **2. Match between system and the real world**



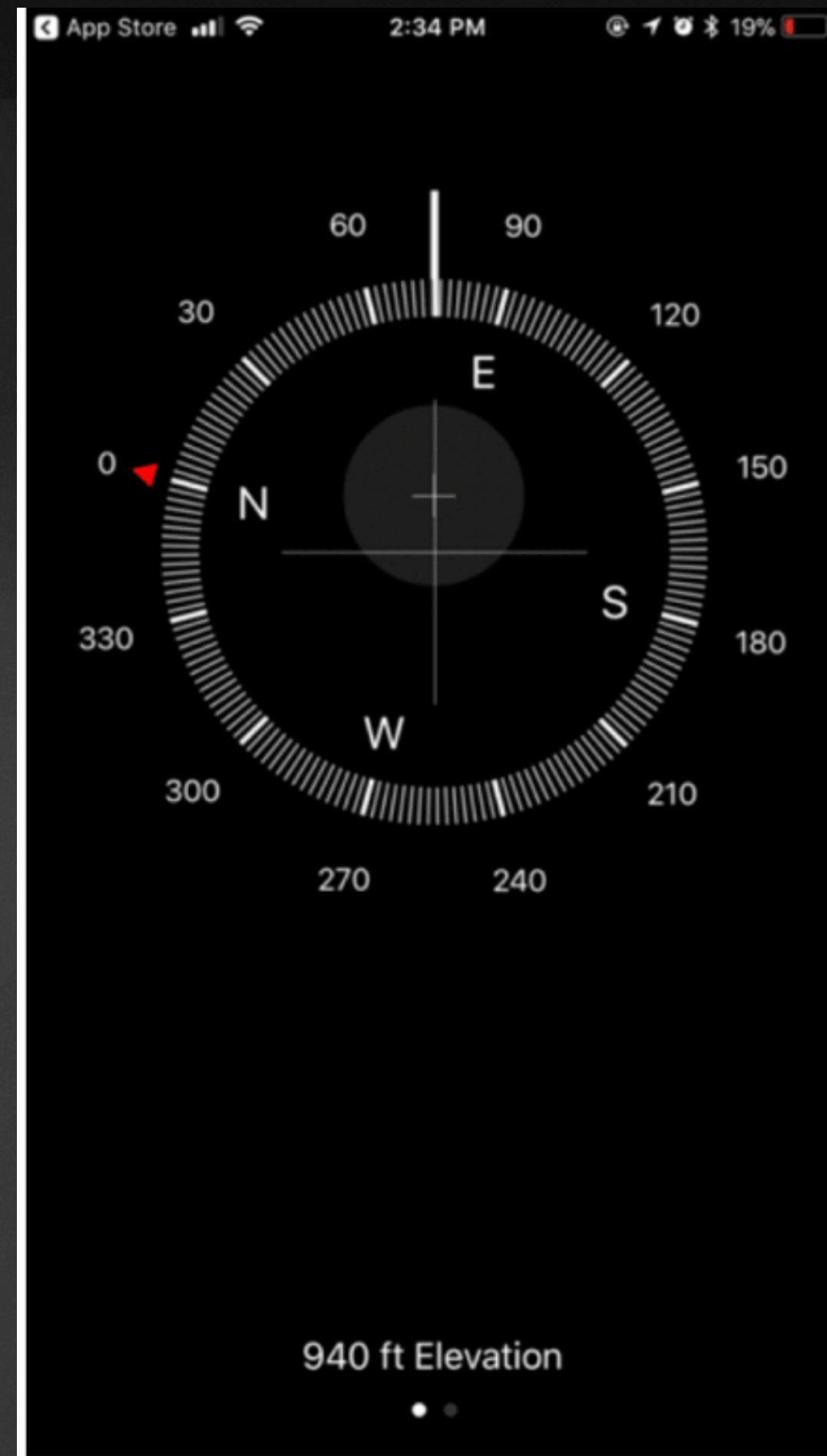
**Match between system and the real world**

**Speak the users' language**

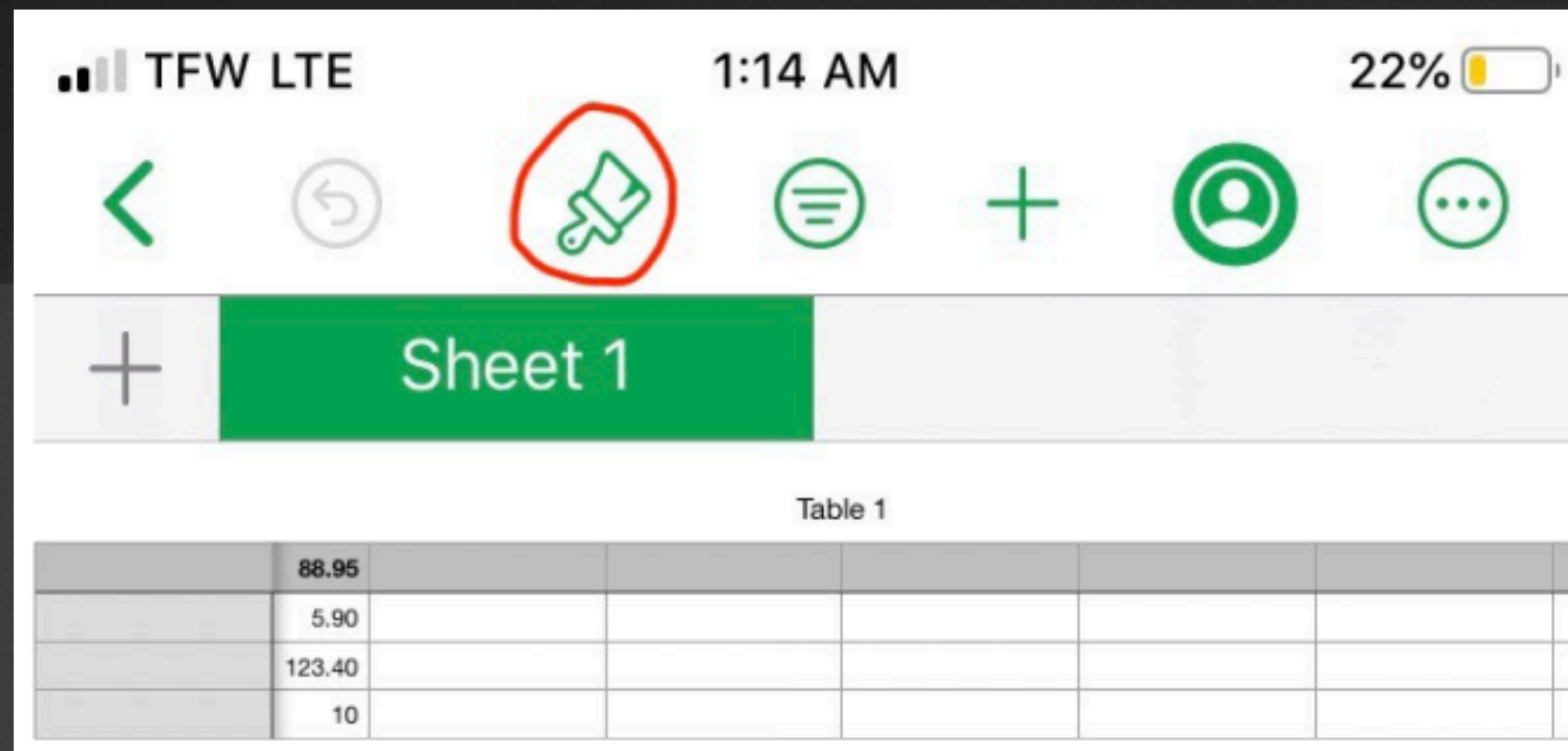
# Match between system and the real world



# Match between system and the real world



# Match between system and the real world



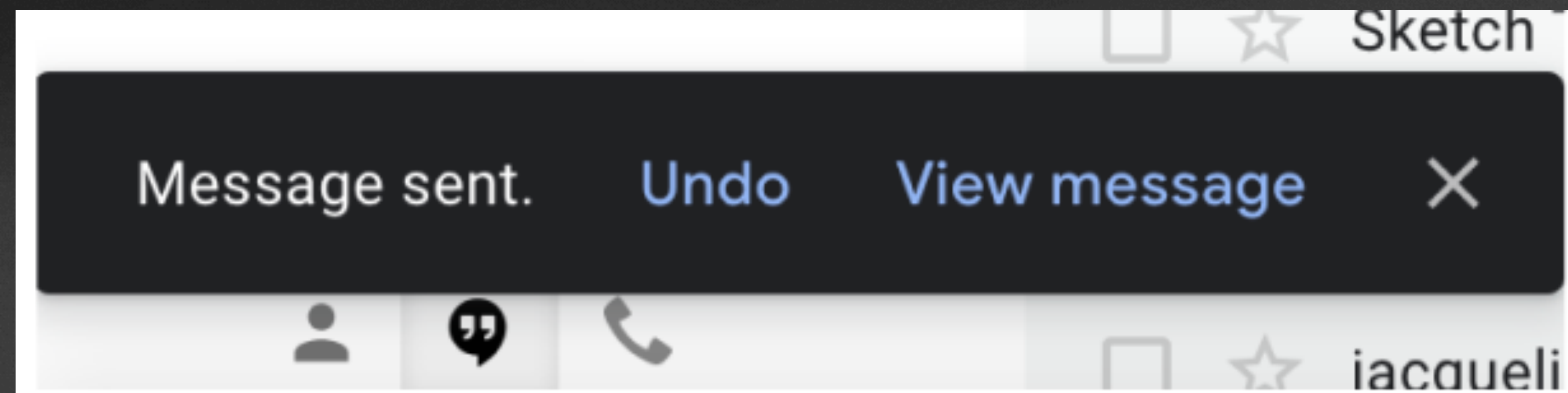
**BAD**

# **3. User control and freedom**

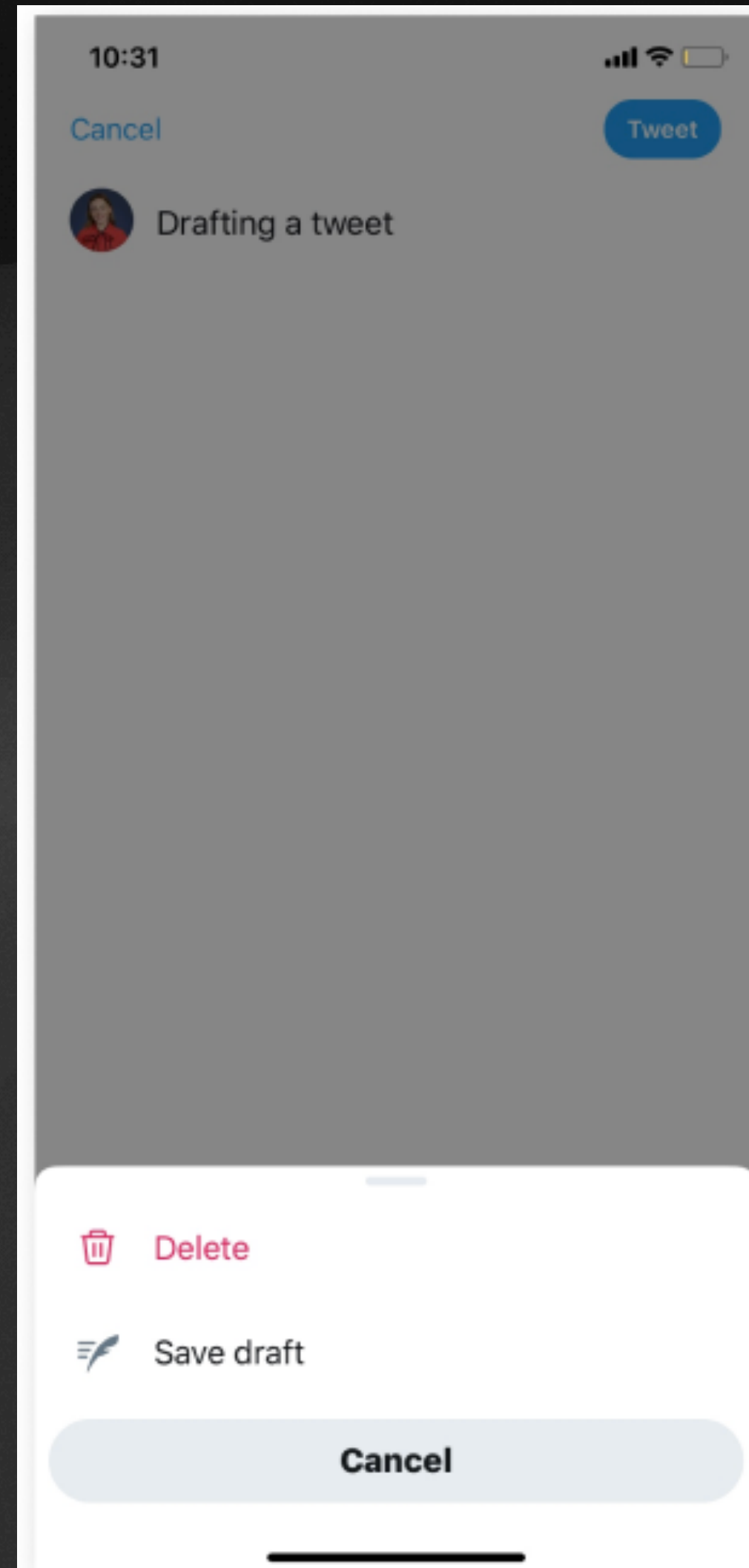
**User control and freedom**

**Make it easy to leave a flow or undo an action**

# User control and freedom

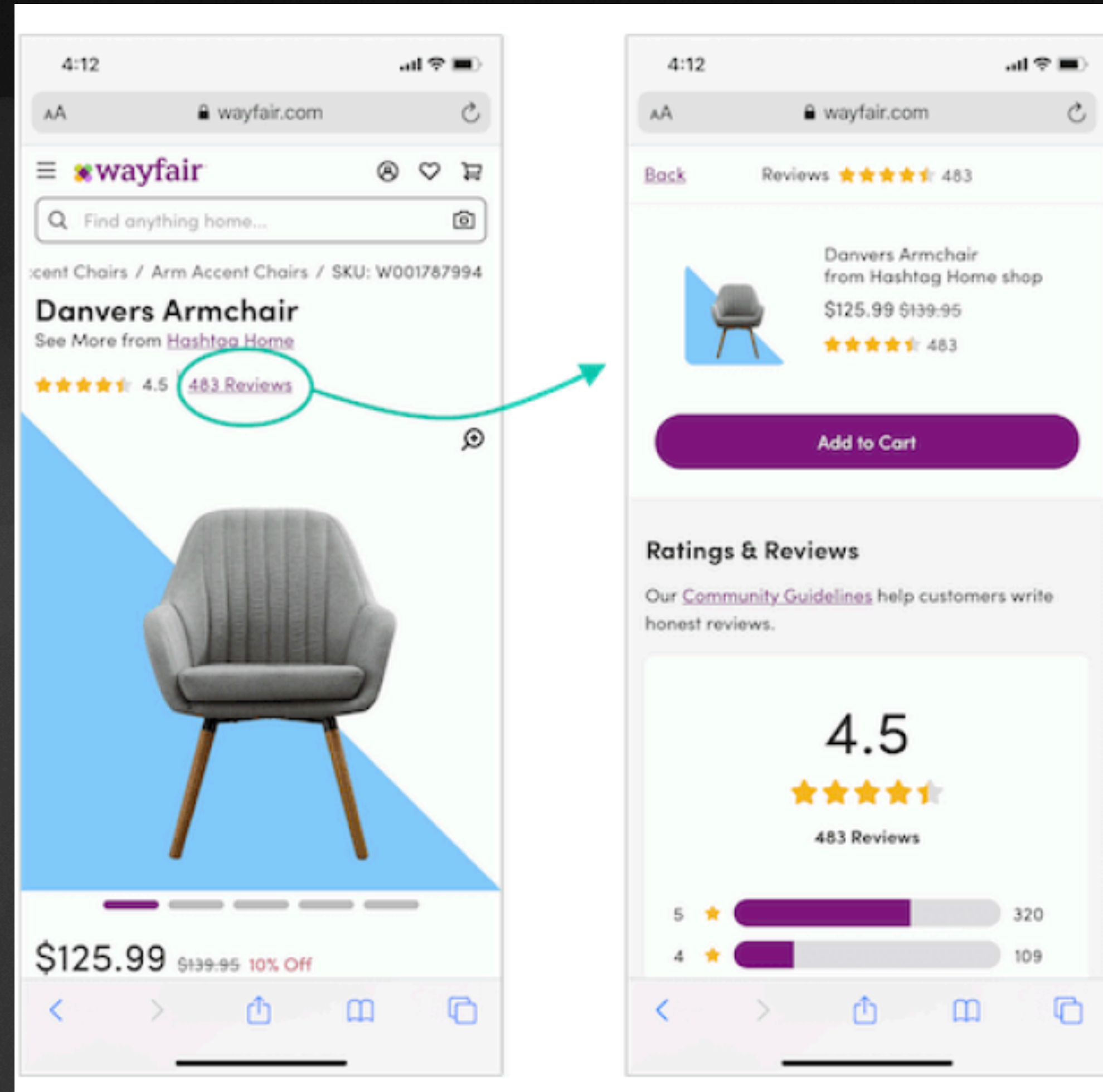


# User control and freedom





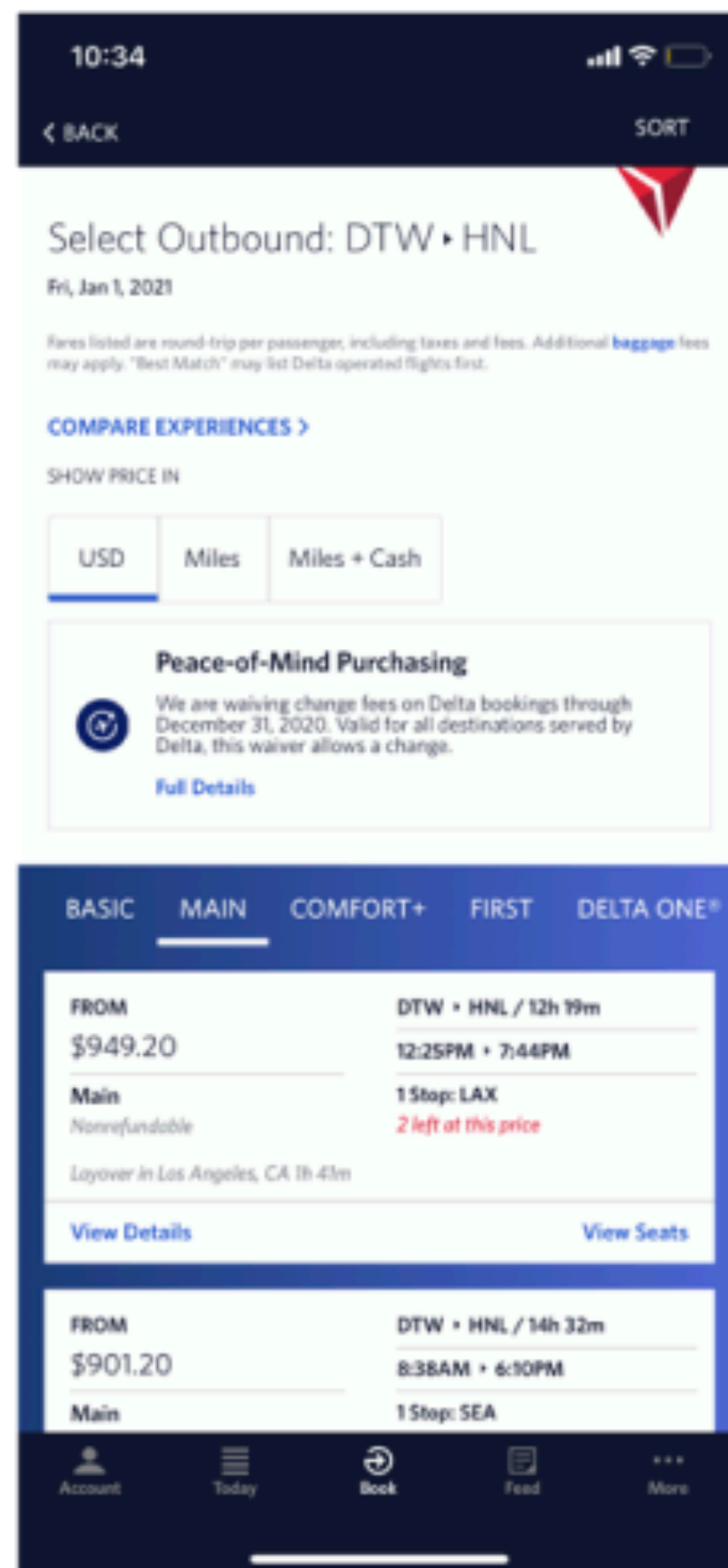
# User control and freedom



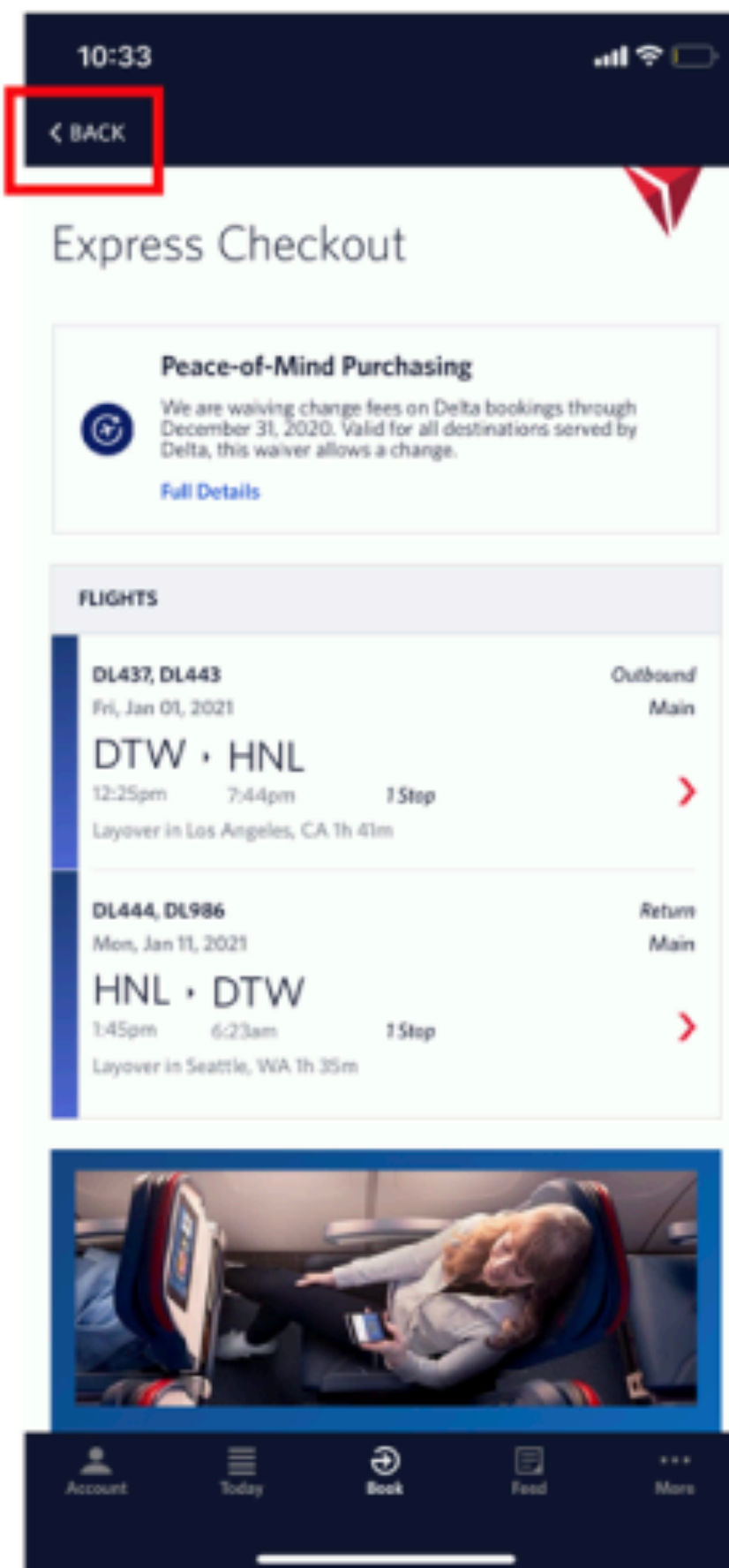
# User control and freedom



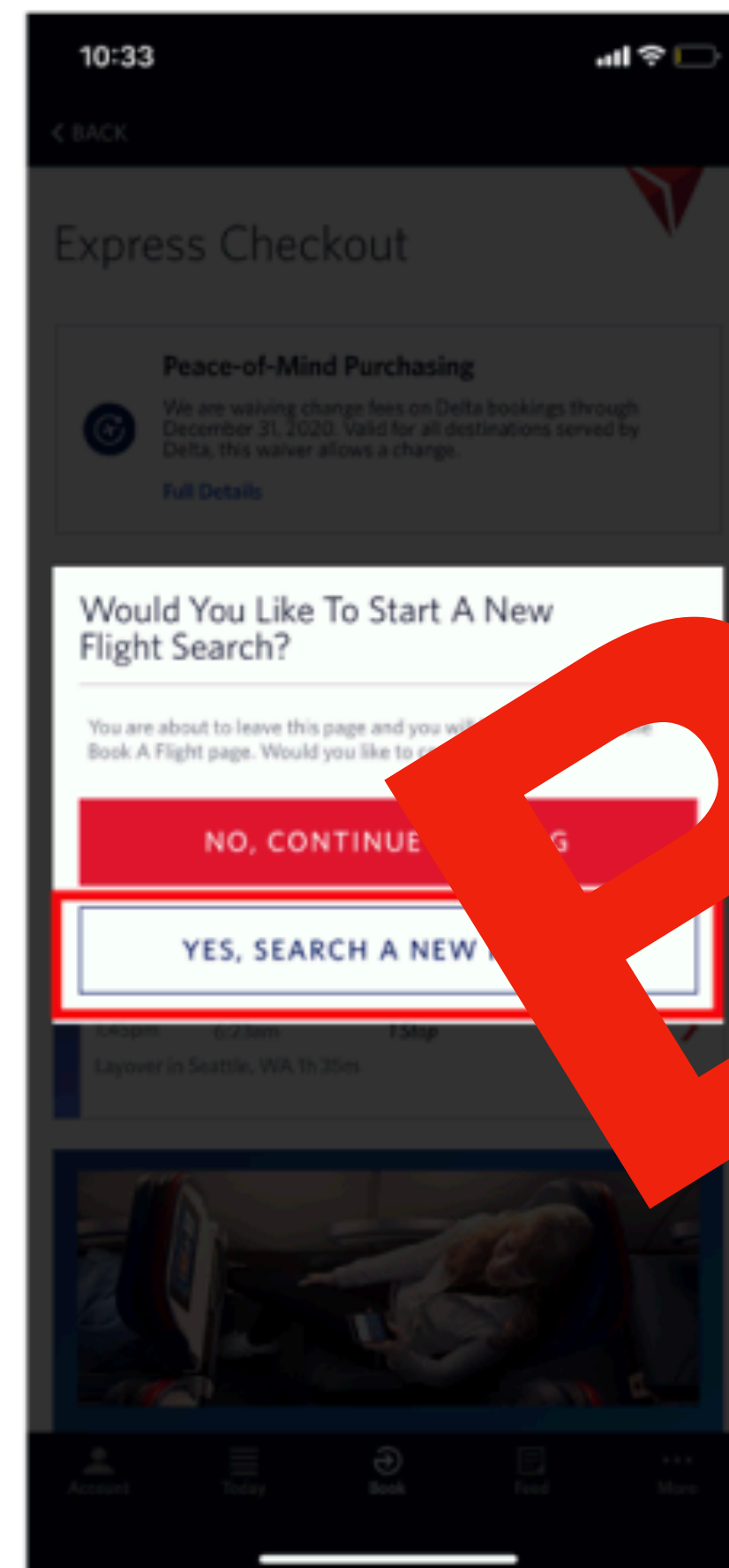
1



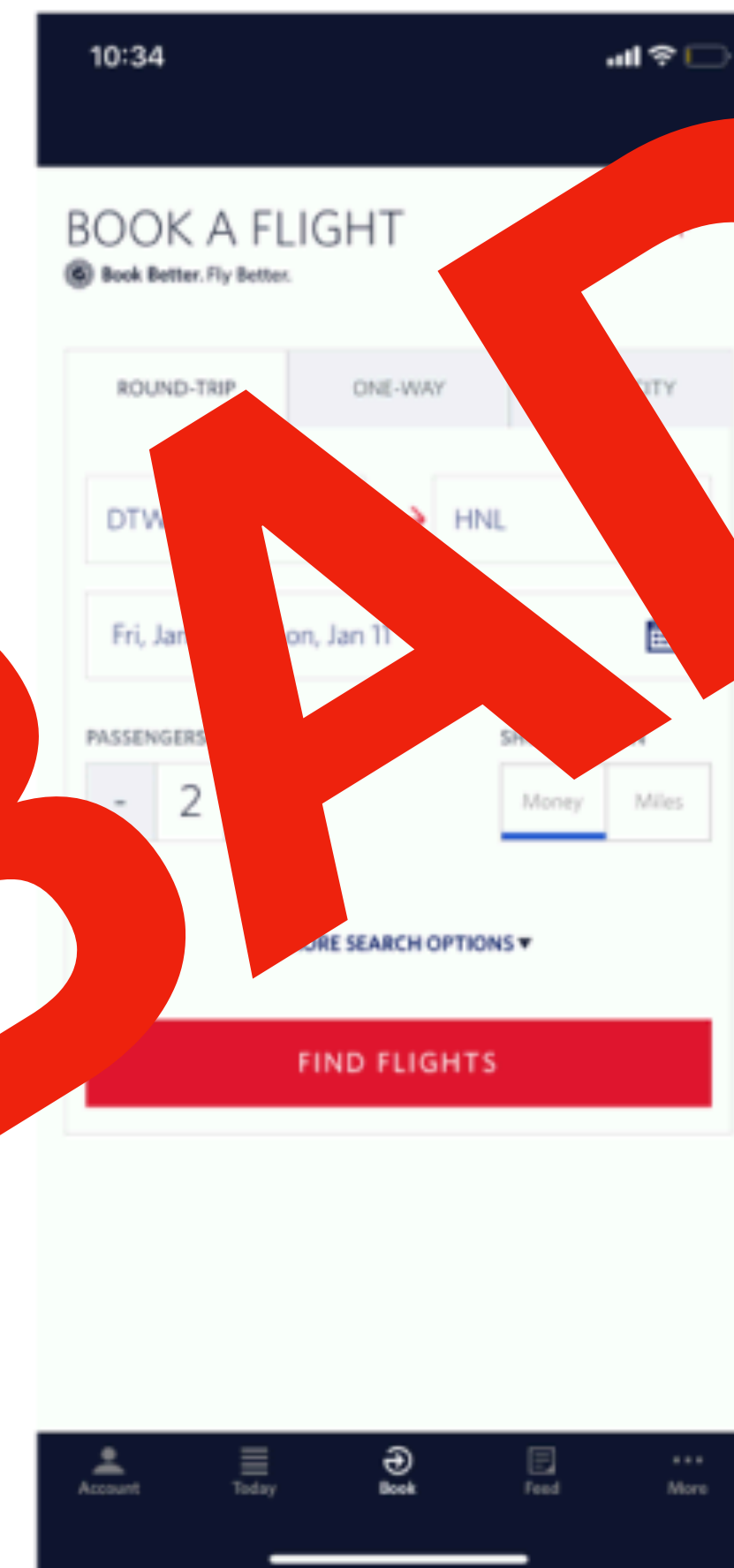
2



3



4



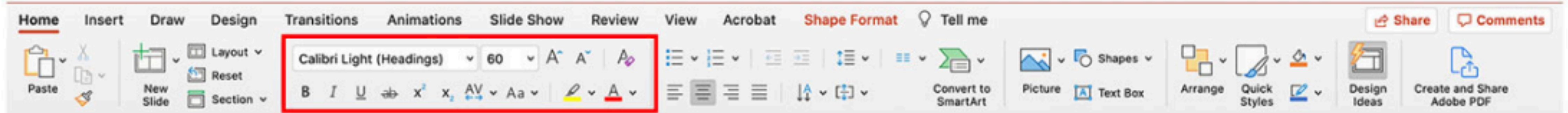
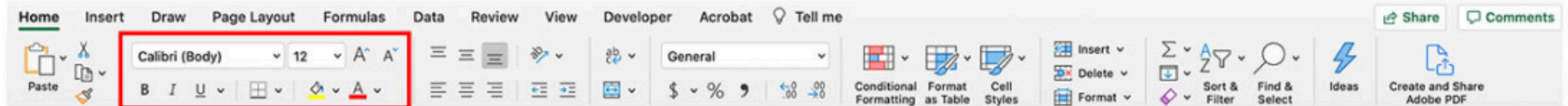
## **4. Consistency and standards**

# Consistency and standards

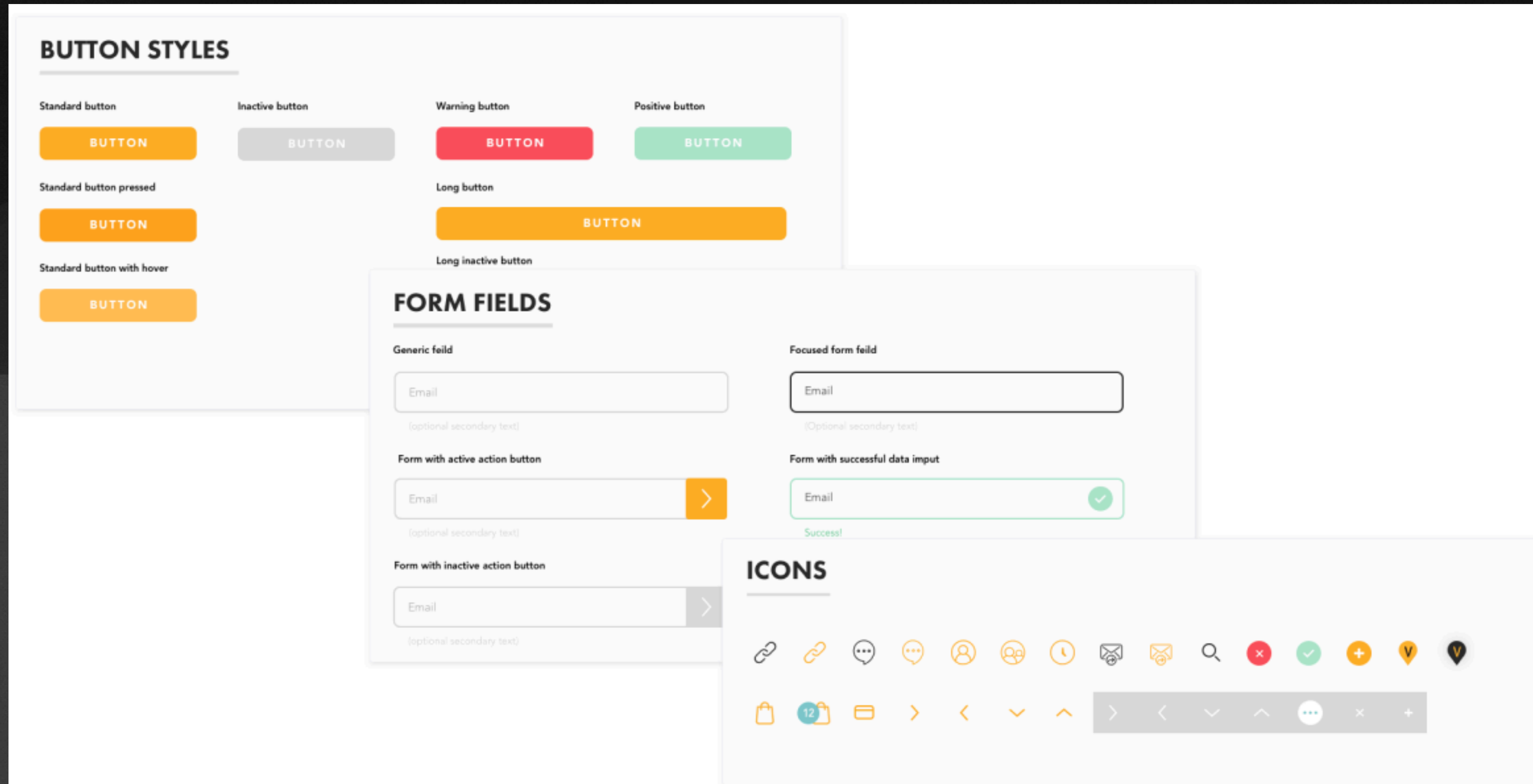
## **Jakob's Law:**

Users spend the majority of their time using products other than yours. They will expect your product to behave like all of those other products.

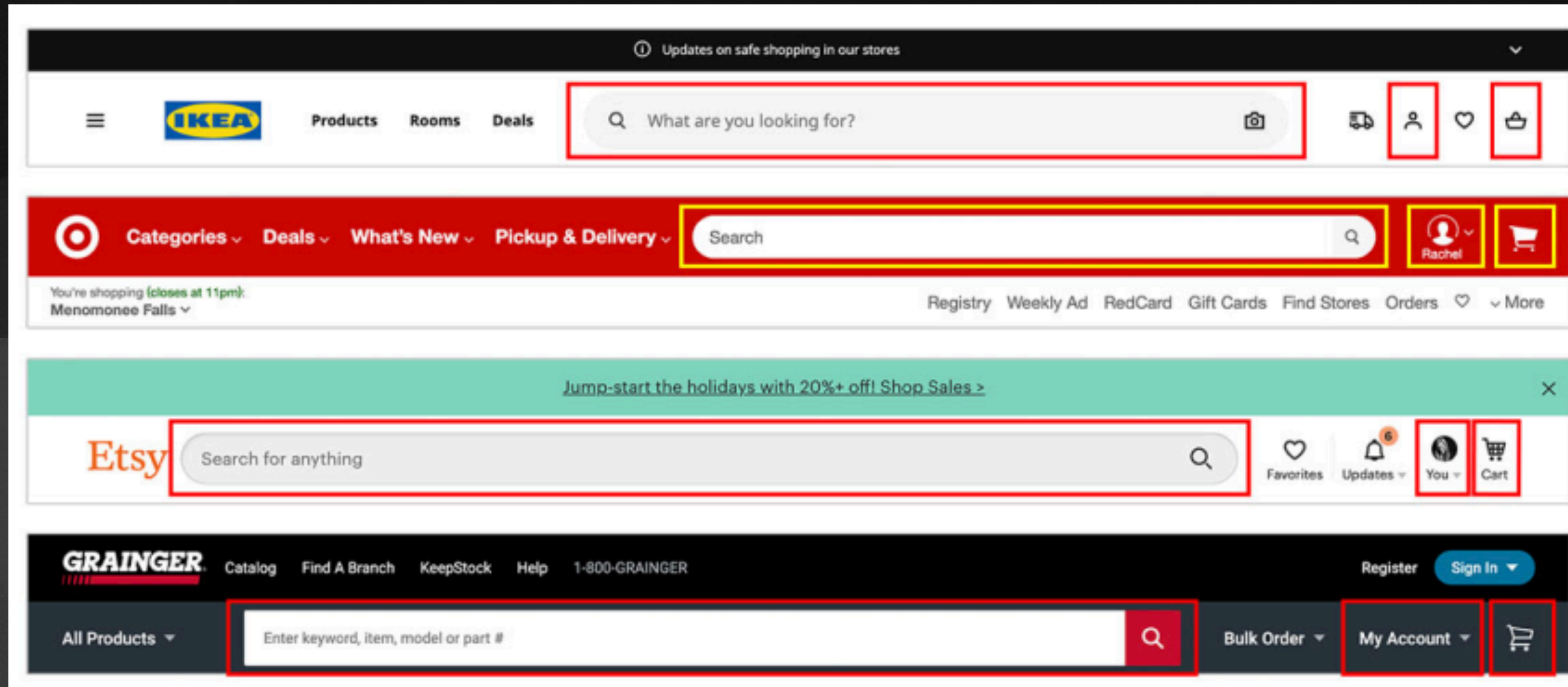
# Consistency and standards



# Consistency and standards



# Consistency and standards



# Consistency and standards

The image displays three sequential screenshots of a patient portal interface, illustrating inconsistencies in button labels and layout. A large red 'BAD' watermark is overlaid on the right side of the screenshots.

- First Screenshot (Left):** Titled 'eZ Arrival', it features a 'Guarantor' icon and the name 'Budi, Raluca'. It includes two questions with radio button options: 'We have this person on file to pay for costs not covered' (Yes/No) and 'Would you like to use insurance to pay for this appointment?' (Use insurance/Do not bill insurance). The bottom navigation bar contains 'NEXT' and 'FINISH LATER' buttons.
- Second Screenshot (Middle):** Also titled 'eZ Arrival', it prompts the user to review allergies. It includes a '+ ADD AN ALLERGY' button, a checkbox for 'This information is correct', and a bottom navigation bar with 'BACK', 'NEXT', and 'FINISH LATER' buttons.
- Third Screenshot (Right):** Titled 'Send Message to Care Team for new non-urgent symptoms', it contains a message verification instruction, a date 'Added 11/3/2010', and an 'i Learn more' link. It includes a checkbox for 'This information is correct' and a bottom navigation bar with 'BACK', 'FINISH LATER', and 'SUBMIT' buttons.

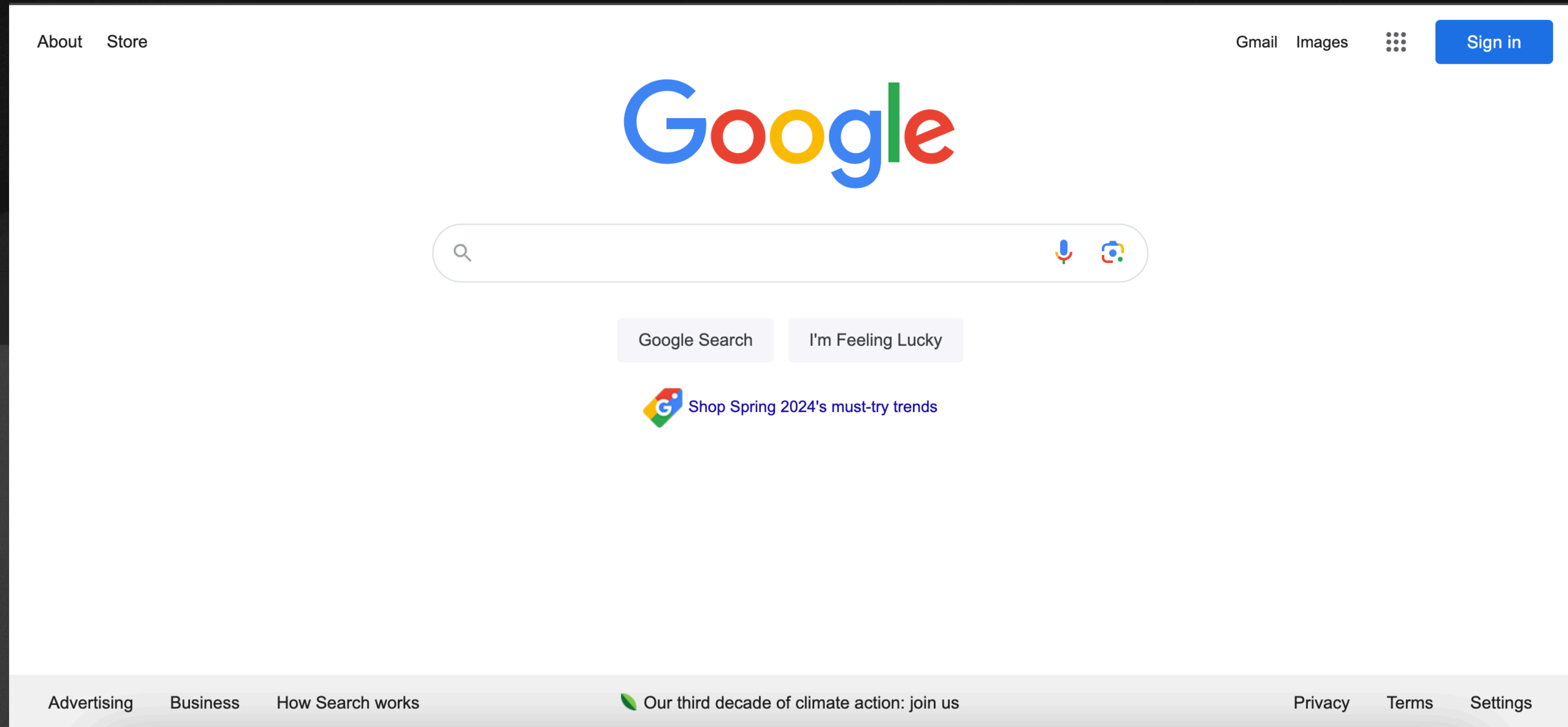


# **5. Aesthetic and minimalist design**

# **Aesthetic and minimalist design**

**Provide relevant information. Remove clutter.**

# Aesthetic and minimalist design



[About](#) [Store](#)

[Gmail](#) [Images](#)



[Sign in](#)


Google

Google Search

I'm Feeling Lucky

 [Shop Spring 2024's must-try trends](#)

[Advertising](#) [Business](#) [How Search works](#)

 [Our third decade of climate action: join us](#)

[Privacy](#) [Terms](#) [Settings](#)

# Aesthetic and minimalist design

## How to Train Your Brain to Remember Almost Anything

Four techniques for storing  
knowledge you might otherwise  
forget

 Thomas Oppong · October 16 · 5 min read ★

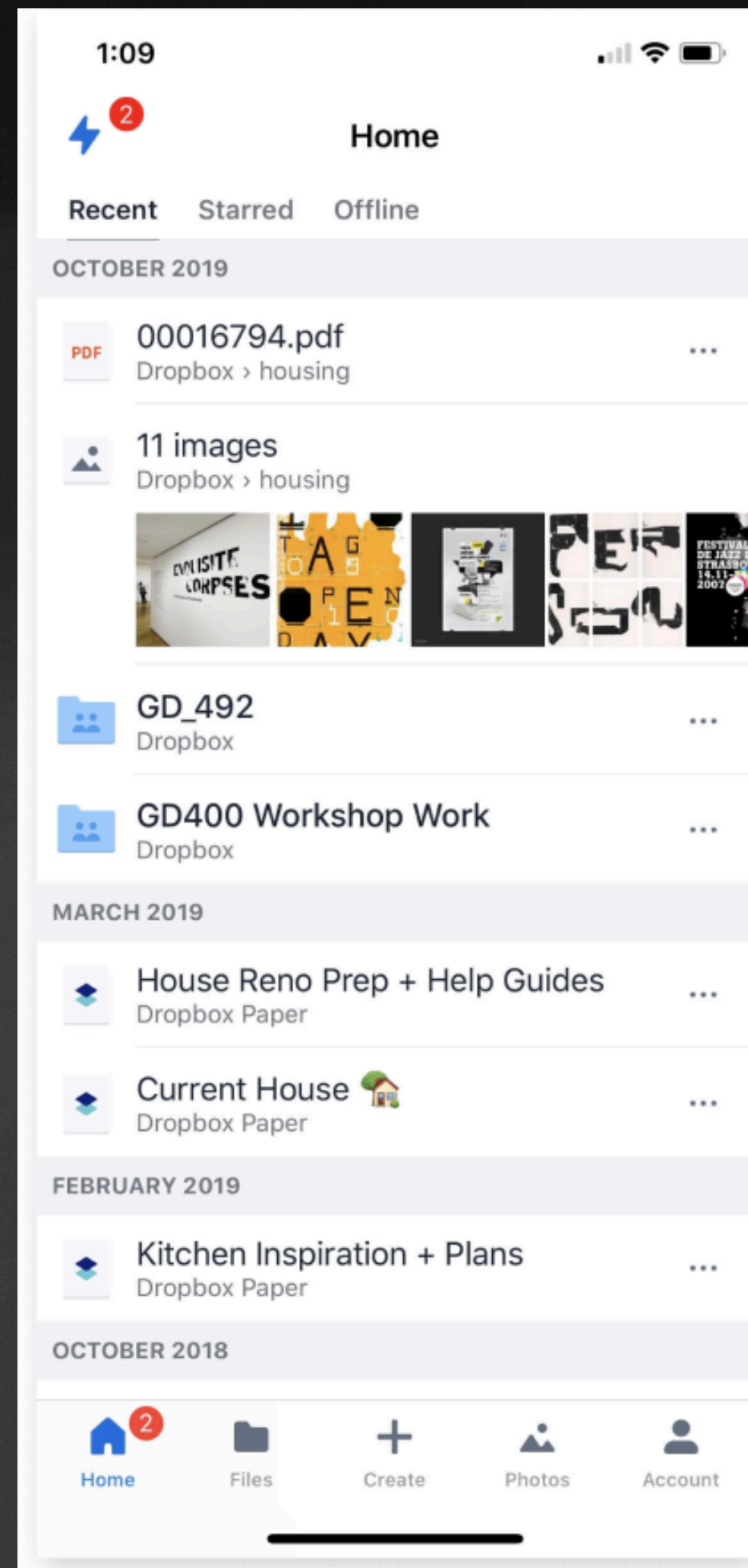


Photo: tunart/Getty

Success is largely based on what you know — everything you know informs the choices you make. And those choices are either getting you closer to what you want or increasing the distance between you and

   AA

# Aesthetic and minimalist design



**BAD**

# Aesthetic and minimalist design

<b>Form 1040</b> Department of the Treasury—Internal Revenue Service (99)		<b>2017</b>	OMB No. 1545-0074	IRS Use Only—Do not write or staple in this space.
For the year Jan. 1–Dec. 31, 2017, or other tax year beginning _____, 2017, ending _____, 20		See separate instructions.		
Your first name and initial	Last name	Your social security number		
If a joint return, spouse's first name and initial	Last name	Spouse's social security number		
Home address (number and street). If you have a P.O. box, see instructions.		Apt. no.	▲ Make sure the SSN(s) above and on line 6c are correct.	
City, town or post office, state, and ZIP code. If you have a foreign address, also complete spaces below (see instructions).		Presidential Election Campaign		
		Check here if you, or your spouse if filing jointly, want \$3 to go to this fund. Checking a box below will not change your tax or refund.		
Foreign country name	Foreign province/state/county	Foreign postal code	<input type="checkbox"/> You <input type="checkbox"/> Spouse	

**BAD**

# Read More...

If you have time, check this out:

**Book: *Refactoring UI***

[https://www.dropbox.com/s/q1gmc3fftuhwxgq/Refactoring UI v1.0.2.pdf?dl=0](https://www.dropbox.com/s/q1gmc3fftuhwxgq/Refactoring%20UI%20v1.0.2.pdf?dl=0)



Refactoring UI

by Adam Wathan & Steve Schoger

# Accessibility















# Why Accessibility?

- By making your product accessible, you are ensuring that users with disabilities have a good user experience
- Many existing sites have accessibility barriers that make them difficult or impossible for some people to use

# Who Determines What's Accessible?

- The World Wide Web Consortium (W3C) is an international community that develops web standards
- Their Web Content Accessibility Guidelines (WCAG) covers a wide range of accessibility best practices: [https://  
www.w3.org/TR/WCAG20/](https://www.w3.org/TR/WCAG20/)

# Groups to Consider for

	Permanent	Temporary	Situational
Touch	 One arm	 Arm injury	 New parent
See	 Blind	 Cataract	 Distracted driver
Hear	 Deaf	 Ear infection	 Bartender
Speak	 Non-verbal	 Laryngitis	 Heavy accent

# Accessibility and Color

# Don't Use Color Alone to Convey Information

Sign up for PayPal, it's free.

## Personal Account

Shop, receive money, or just pay someone back for lunch. All without sharing your payment info.



Continue

Huh?

# Don't Use Color Alone to Convey Information

Sign up for PayPal, it's free.

## Personal Account

Shop, receive money, or just pay someone back for lunch. All without sharing your payment info.

Email



Create your password



Confirm your password



Code



Continue

Oh.

# Don't Use Color Alone to Convey Information



---

API



---

Dashboard



---

Gateway

---



---

API



---

Dashboard



---

Gateway

---



# ...but don't take it too far, either



Show app list in menu	<input checked="" type="checkbox"/>	On
Show recently added apps	<input type="checkbox"/>	Off
Show most used apps	<input checked="" type="checkbox"/>	On
Show app notifications	<input type="checkbox"/>	Off



Show app list in menu	<input type="checkbox"/>	
Show recently added apps	<input type="checkbox"/>	
Show most used apps	<input type="checkbox"/>	
Show app notifications	<input type="checkbox"/>	



# Color Contrast

- We determine whether there is enough visual contrast between two colors in our UI using **contrast ratios**
- Contrast ratios can range from 1 to 21 (commonly written 1 : 1 or 21 : 1)

# Sufficient contrast between text and background

- WCAG defines **4.5 : 1** as the minimum contrast ratio a piece of <24px text can have. The minimum ratio for text >24px is **3 : 1**.

Insufficient contrast  
between text and  
background

✗ 2.97 : 1

Sufficient contrast  
between text and  
background

✓ 11.07 : 1

# Sufficient contrast between text and background

This means the **lightest possible gray** you can use on a white background is #767676

✓ 4.54 : 1

And if it's over 24px large, you can go as light as #949494

✓ 3.03 : 1

# How do we determine a contrast ratio?

The screenshot shows the WebAIM online tool interface. It features two color selection boxes: 'Foreground Color' with hex code #8A8AFF and a lightness slider, and 'Background Color' with hex code #FFFFFF and a lightness slider. A central box displays the 'Contrast Ratio' as 2.93:1 with a 'permalink' link below it. Below this, there are two sections for text contrast: 'Normal Text' and 'Large Text'. Each section shows 'WCAG AA: Fail' and 'WCAG AAA: Fail' next to a sample text box containing 'The five boxing wizards jump quickly.' in blue text on a white background.

WebAIM online tool

The screenshot shows the Figma Contrast Plugin interface. It has a title bar with a checkmark icon and the word 'Contrast'. Below the title bar are 'Select' and 'Scan' buttons. The main area displays two color swatches: a red one with 'Abc' and hex code #C22929, and a lighter red one with 'Abc' and hex code #FOA8A4. Below the swatches, it shows 'Contrast Ratio' as 2.97 : 1. Underneath, there are three rows of 'FAIL' buttons: 'Normal Text' (two buttons), 'Large Text' (two buttons), and 'Graphics' (one button). At the bottom, there is a toggle switch for 'Enable Smart Sample for layer' and a help icon.

Figma Contrast Plugin

# This applies to image backgrounds as well



# This applies to image backgrounds as well

Lighten and darken as necessary



Do this by overlaying a white or black rectangle of x% opacity :))

# Mobile Typography

# Mobile Typography

- **16px is a good base size to start with!**
- Increase to 18-20px for long-form reading
- Decrease as low as 10px for tertiary elements
- Always test on your real device to get a feel for sizing!



# Coding for Custom UI

# Custom Button Style

# 1 Defining a Custom Button Style

```
struct MyCustomButtonStyle: ButtonStyle {  
    func makeBody(configuration: Self.Configuration) -> some View {  
        configuration.label  
            .padding()  
            .background(configuration.isPressed ? Color.gray : Color.blue)  
            .foregroundColor(.white)  
            .clipShape(RoundedRectangle(cornerRadius: 10))  
            .scaleEffect(configuration.isPressed ? 0.95 : 1.0)  
    }  
}
```

## 2 Using a Custom Button Style

```
struct ContentView: View {
    var body: some View {
        Button("Press Me") {
            print("Button pressed!")
        }
        .buttonStyle(MyCustomButtonStyle())
    }
}
```

# Custom View Modifier

# 1 Defining a Custom View Modifier

```
struct MyCustomModifier: ViewModifier {  
    var backgroundColor: Color = .green  
  
    func body(content: Content) -> some View {  
        content  
        .padding()  
        .background(backgroundColor)  
        .clipShape(Circle())  
        .shadow(radius: 10)  
    }  
}
```

## 2 Wrapping a Custom View Modifier

```
extension View {  
    func myCustomStyle(backgroundColor: Color = .green) -> some View {  
        self.modifier(MyCustomModifier(backgroundColor: backgroundColor))  
    }  
}
```

# 3 Using a Custom View Modifier

```
struct ContentView: View {  
    var body: some View {  
        Text("Hello, World!")  
            .myCustomStyle(backgroundColor: .blue)  
    }  
}
```



# More...

- App Icon
- App Launch Animation
- Custom Color Set
- Compatibility for multiple OS versions/platforms

**Thank You!**