

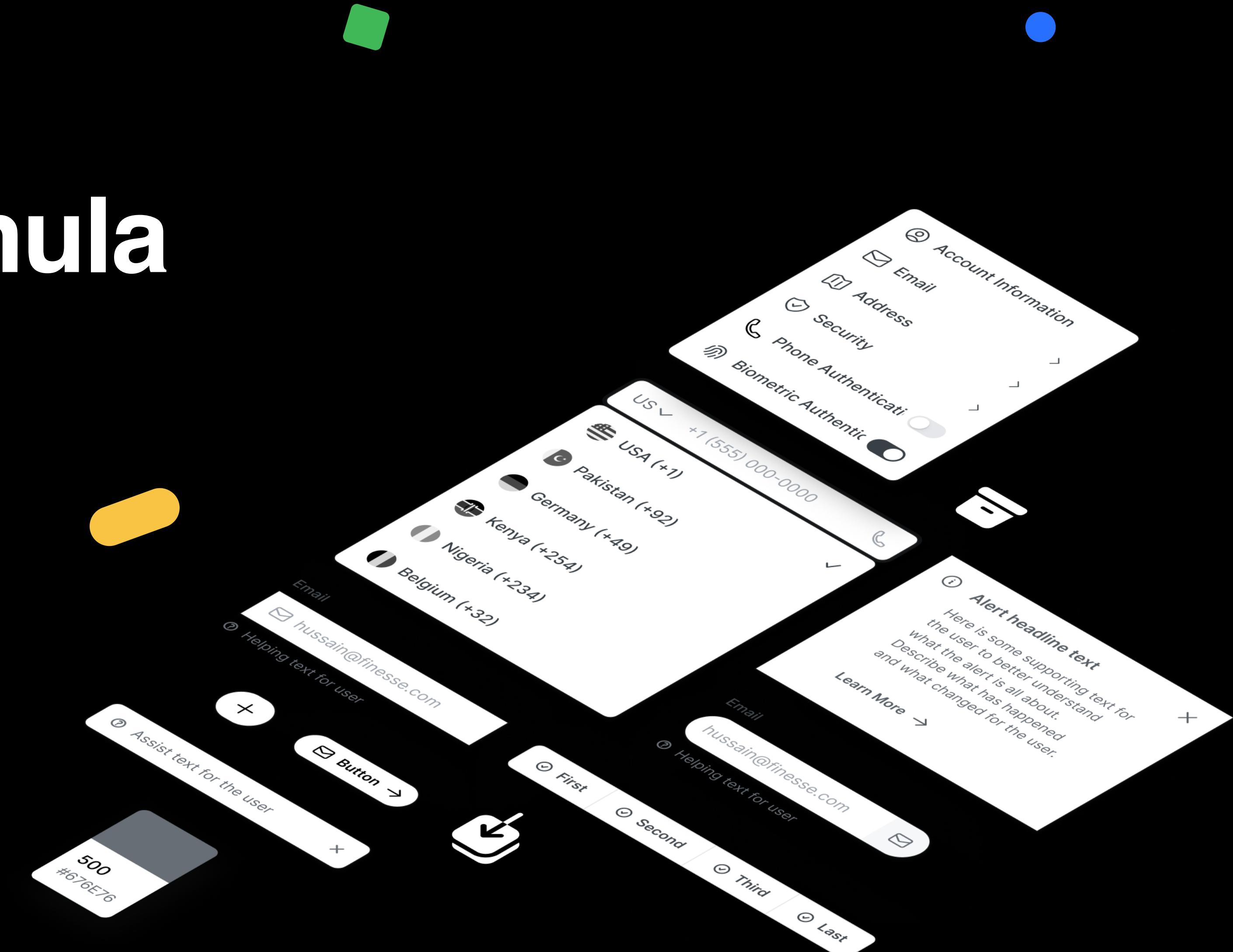


Swift Cairo
iOS Developers Community

The Winning Formula for Mobile Design system

👤 Mohammed Gamal

areebtechnology.com





About Me

I'm **Mohamed Gamal**, a **Senior iOS Software Engineer** at **Areeb Technology**, A **Certified Scrum Developer® (CSD®)**, and occasionally a tech article author.

When I'm not immersed in Swift or discussing Design Systems, you'll find me swimming, biking, or running — I'm an enthusiastic triathlete! I thrive on balancing the precision of code with the endurance of sports.

Let's make this session as dynamic as a triathlon! 

Brief overview of the session goals

Understand the Concept of a Design system💡

Why We Use Design Systems🎯

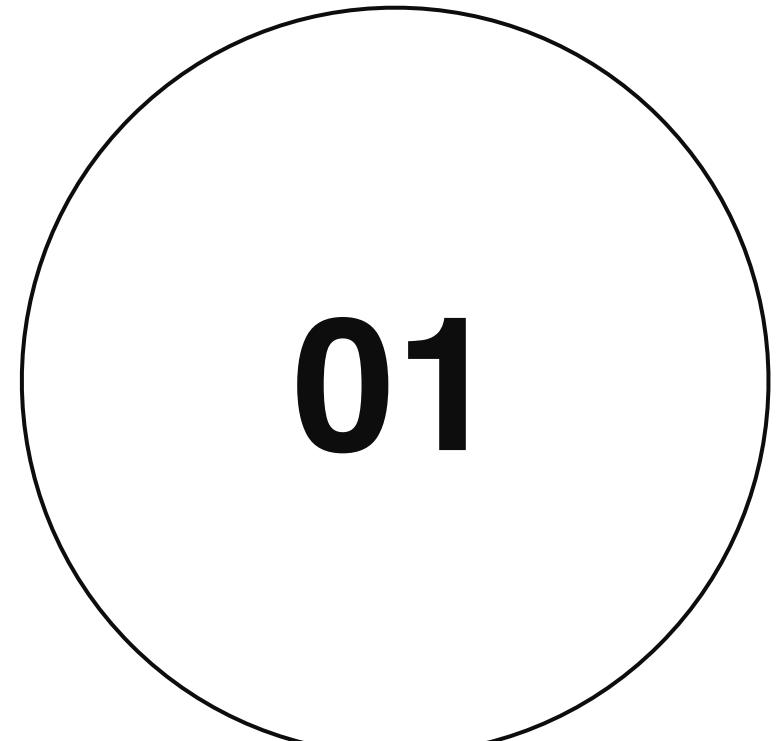
How to Implement a Design System🔧



What?

Why?

How?



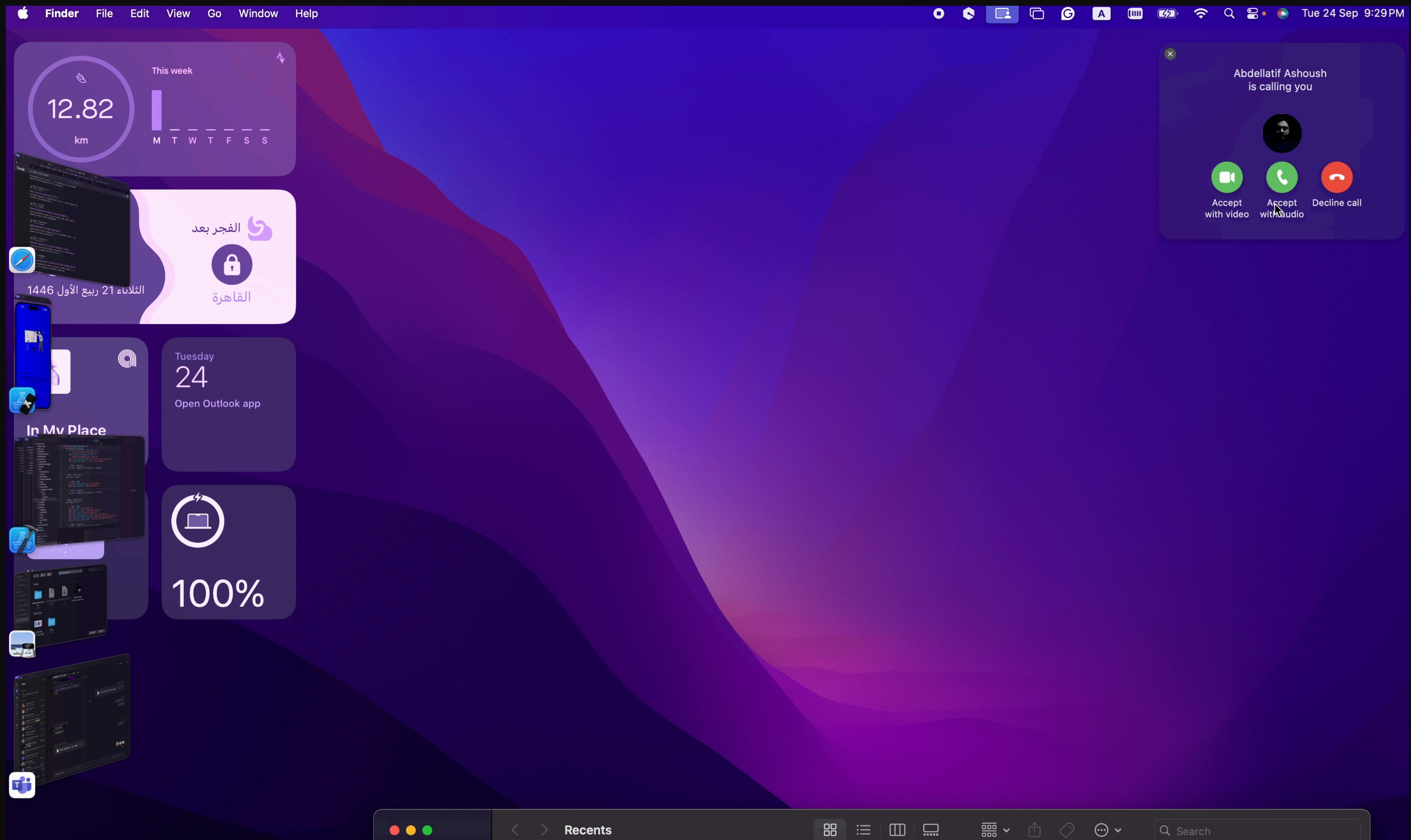
01

What exactly is a design system?

- A design system is a collection of reusable UI components and guidelines that ensure consistency across your application.
- Think of it as your UI toolkit: Colors, Typography, Icons, Spacing, and more—centralized and organized.
- A design system isn't just about making things pretty—it's about creating a toolkit that lets us build quickly and consistently.



What If We Don't Use a Design System



Why Use a Design System?

02

1. **Consistency:** Achieve a unified look and feel across every part of the app.
2. **⚡ Efficiency:** Reuse components to save time and avoid repeating design decisions.
3. **↗ Scalability:** Grow your app easily while maintaining consistent design.

03

How do I use a design system?

Foundation



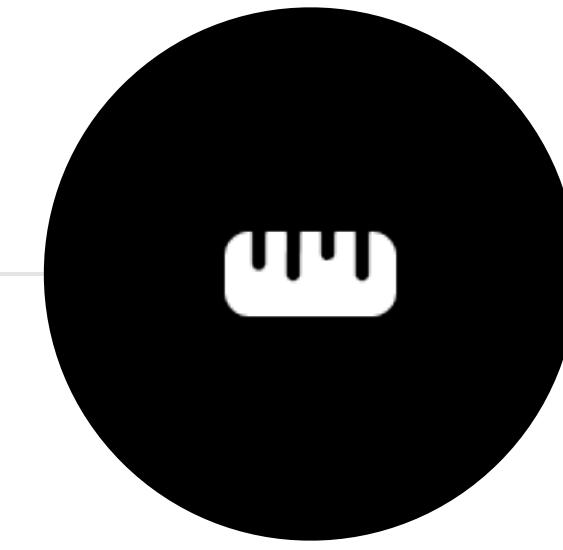
Colors



Typography



Icons



Measurements

Foundation



Colors



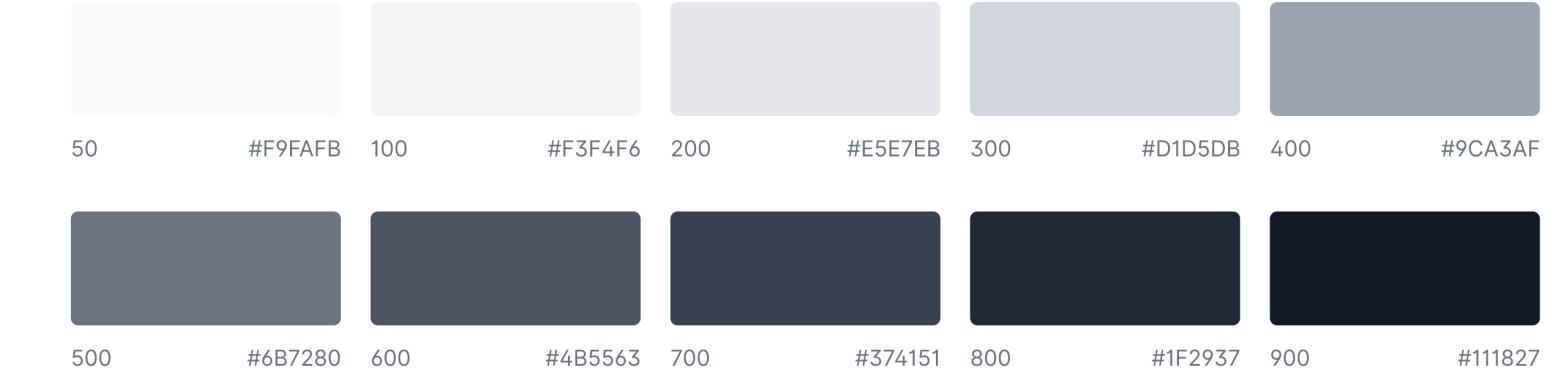
Colors

Shipfaster UI uses a flexible color palette to achieve clean interfaces and captivating brand experiences.

Primary Colour Palette

Neutral AAA

These colors are used as supporting secondary colors in backgrounds, text colors, separators, models, etc



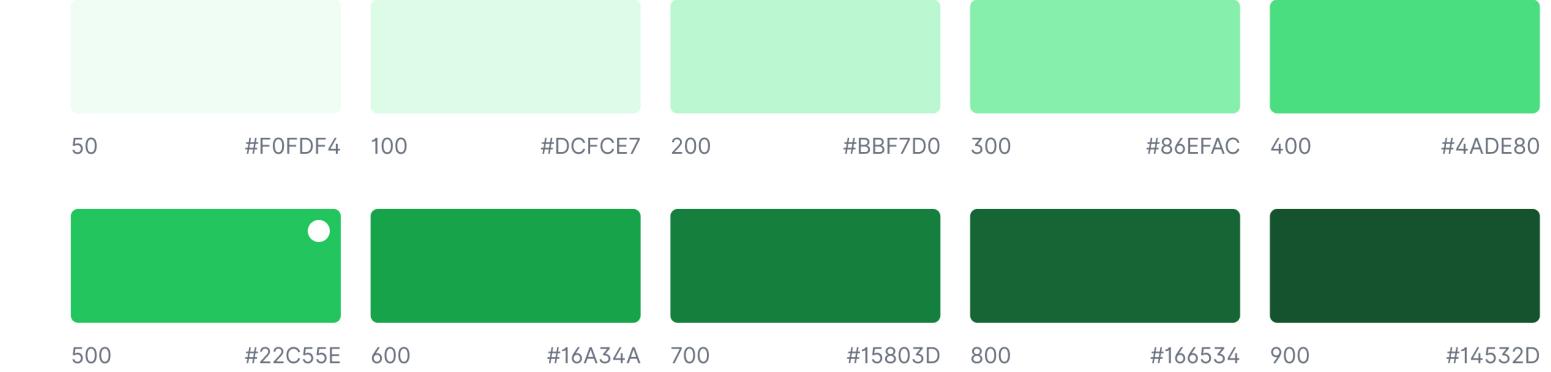
Primary AAA

The primary color palette is used across all the interactive elements such as CTA's, links, inputs, active states, etc



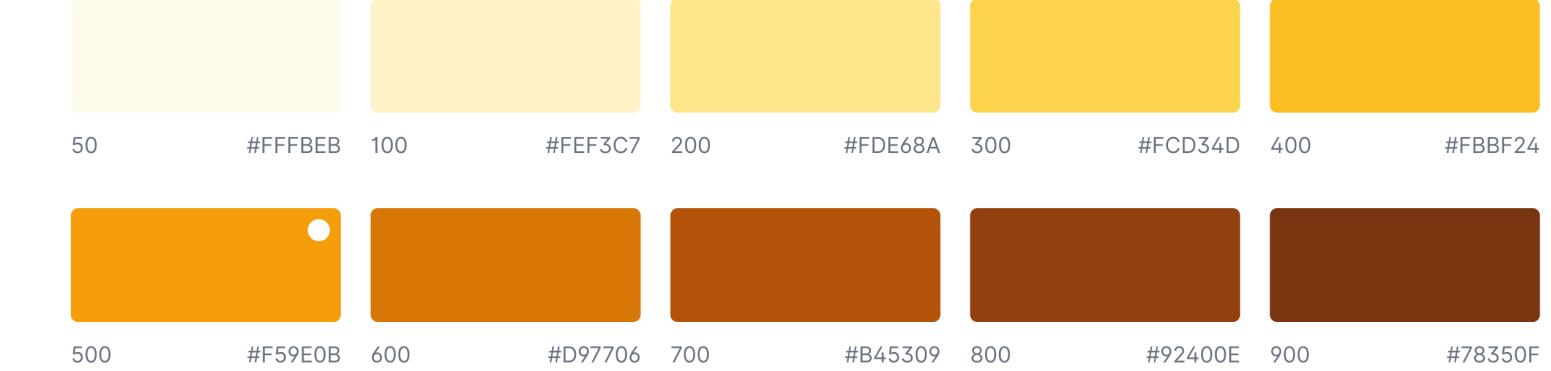
Success AAA

These colors convey an emotion of positivity. Generally used across success, complete states.



Warning AAA

These colors convey the feeling of caution. Generally used across warning states.



Foundation



Typography



Headings - Mobile

A set of harmonious typography scale with font weight variations. These text style perfectly fit for 4px grid and go well with each other.

Heading H1

Font size: 36px | Line height: 44px | Tracking: -2%

Heading H1

Heading H1

Heading H1

Heading H1

Heading H2

Font size: 32px | Line height: 40px | Tracking: -2%

Heading H2

Heading H2

Heading H2

Heading H2

Heading H3

Font size: 28px | Line height: 36px | Tracking: -2%

Heading H3

Heading H3

Heading H3

Heading H3

Foundation



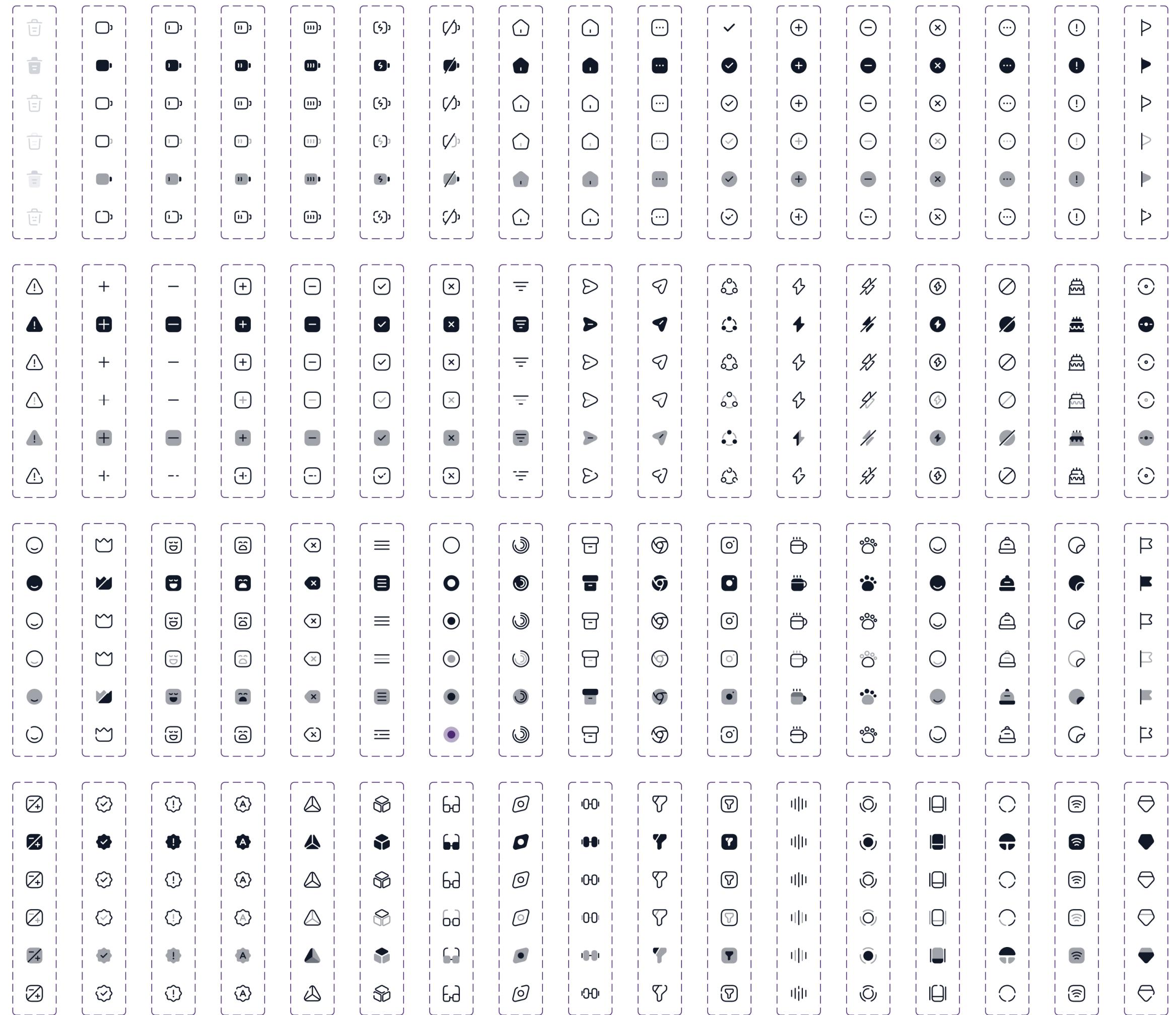
Icons



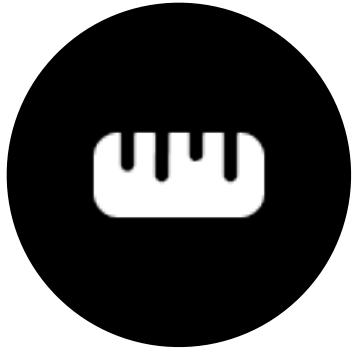
Icons

A carefully selected icon library that contains beautiful pixel-perfect SVG icons

Essentional



Foundation



Measurements

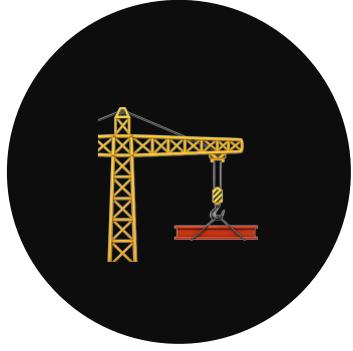


Measurements

A comprehensive spacing system with proposional to a 4px scale which properly aligns to the layout and elements look harmonous.

Name	Token	Size	Color
1	\$spacing - x small	4px	#e0e0ff
2	\$spacing - small	8px	#e0e0ff
3	\$spacing - medium	12px	#e0e0ff
4	\$spacing - lg	16px	#e0e0ff
5	\$spacing - xl	20px	#e0e0ff
6	\$spacing - 2xl	24px	#e0e0ff
7	\$spacing - 3xl	28px	#e0e0ff
8	\$spacing - 4xl	32px	#e0e0ff
10	\$spacing - 5xl	40px	#e0e0ff
12	\$spacing - 6xl	48px	#e0e0ff
13	\$spacing - 7xl	52px	#e0e0ff
16	\$spacing - 8xl	64px	#e0e0ff
20	\$spacing - huge	80px	#e0e0ff
24	\$spacing - x huge	96px	#e0e0ff
32	\$spacing - 2x huge	128px	#e0e0ff
40	\$spacing - 3x huge	160px	#e0e0ff
48	\$spacing - 4x huge	192px	#e0e0ff

Foundation



Design System Structure

```
1 //  
2 // DesignSystemFoundations.swift  
3 // SwiftCairoDesignSystem  
4 //  
5 // Created by Mohamed Gamal on 23/09/2024.  
6 //  
7  
8 import Foundation  
9  
10 enum DesignSystem {  
11     enum Foundation {}  
12 }  
13  
14 extension DesignSystem.Foundation {  
15     enum Colors {}  
16     enum Typography {}  
17     enum Icons {}  
18     enum Spacing{}  
19 }  
20
```

Foundation



Colors Assets

The screenshot shows the 'Colors' section of the Foundation library. The sidebar on the left lists color categories: AccentColor, AppIcon, Colors, Destructive, Neutral, Primary, and Shade. The 'Destructive' category is currently selected, highlighted with a blue bar at the top of the list. Below it are sub-categories: DestructiveX50, DestructiveX100, DestructiveX200, DestructiveX300, DestructiveX400, DestructiveX500, DestructiveX600, DestructiveX700, DestructiveX800, DestructiveX900. To the right of the sidebar, five color swatches are displayed in a grid, each labeled with its name and appearance variants: DestructiveX50 (Any Appearance, Dark, Universal), DestructiveX100 (Any Appearance, Dark, Universal), DestructiveX200 (Any Appearance, Dark, Universal), DestructiveX300 (Any Appearance, Dark, Universal), and DestructiveX400 (Any Appearance, Dark, Universal). A 'Filter' button is located at the bottom of the sidebar.

Color Category	Sub-Categories	Swatch Name	Appearance	Variant
Destructive	DestructiveX50, DestructiveX100, DestructiveX200, DestructiveX300, DestructiveX400, DestructiveX500, DestructiveX600, DestructiveX700, DestructiveX800, DestructiveX900	DestructiveX50	Any Appearance	Dark
Destructive	DestructiveX50, DestructiveX100, DestructiveX200, DestructiveX300, DestructiveX400, DestructiveX500, DestructiveX600, DestructiveX700, DestructiveX800, DestructiveX900	DestructiveX100	Any Appearance	Dark
Destructive	DestructiveX50, DestructiveX100, DestructiveX200, DestructiveX300, DestructiveX400, DestructiveX500, DestructiveX600, DestructiveX700, DestructiveX800, DestructiveX900	DestructiveX200	Any Appearance	Dark
Destructive	DestructiveX50, DestructiveX100, DestructiveX200, DestructiveX300, DestructiveX400, DestructiveX500, DestructiveX600, DestructiveX700, DestructiveX800, DestructiveX900	DestructiveX300	Any Appearance	Dark
Destructive	DestructiveX50, DestructiveX100, DestructiveX200, DestructiveX300, DestructiveX400, DestructiveX500, DestructiveX600, DestructiveX700, DestructiveX800, DestructiveX900	DestructiveX400	Any Appearance	Dark

Why Use The Colors Assets?



Consistency

Colors are used. Across the app.



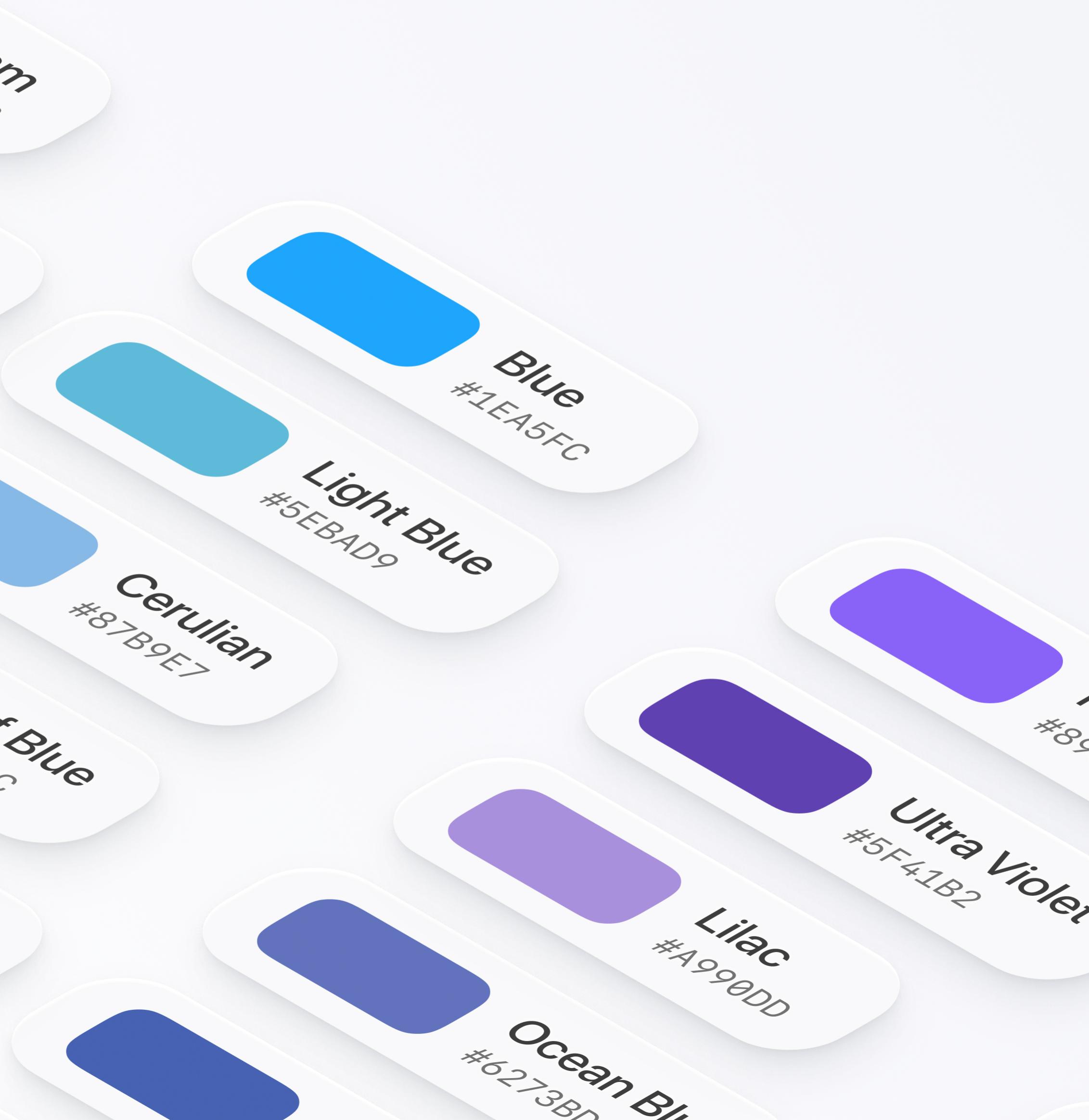
Centralized Management

Update colors in one place.



Dark Mode Support

Easily manage light and dark themes.



Foundation



Colors

```
// MARK: - Primary Colors Family
extension DesignSystem.Foundation.Colors {
    enum Primary {

        // MARK: - Cases
        case x50
        case x100
        case x200
        case x300
        case x400
        case x500
        case x600
        case x700
        case x800
        case x900

        // MARK: - Property
        var color: Color {
            switch self {
            case .x50:
                return Color("PrimaryX50")
            case .x100:
                return Color("PrimaryX100")
            case .x200:
                return Color("PrimaryX200")
            case .x300:
                return Color("PrimaryX300")
            case .x400:
                return Color("PrimaryX400")
            case .x500:
                return Color("PrimaryX500")
            case .x600:
                return Color("PrimaryX600")
            case .x700:
                return Color("PrimaryX700")
            case .x800:
                return Color("PrimaryX800")
            case .x900:
                return Color("PrimaryX900")
            }
        }
    }
}
```

Foundation



Typography

```
extension DesignSystem.Foundation.Typography {
    enum Heading {}
    enum Paragraph {}
}

// MARK: - Heading Fonts
extension DesignSystem.Foundation.Typography.Heading {
    enum XH1 {
        // MARK: - Cases
        case extraBold
        case bold
        case semiBold
        case medium
    }

    // MARK: - Properties
    var font: UIFont {
        switch self {
        case .extraBold:
            return UIFont(name: "Inter-ExtraBold", size: 36.0) ?? .systemFont(ofSize: 36.0, weight: .bold)
        case .bold:
            return UIFont(name: "Inter-Bold", size: 36.0) ?? .systemFont(ofSize: 36.0, weight: .bold)
        case .semiBold:
            return UIFont(name: "Inter-SemiBold", size: 36.0) ?? .systemFont(ofSize: 36.0, weight: .semibold)
        case .medium:
            return UIFont(name: "Inter-Medium", size: 36.0) ?? .systemFont(ofSize: 36.0, weight: .medium)
        }
    }
}
```

Foundation



Icons

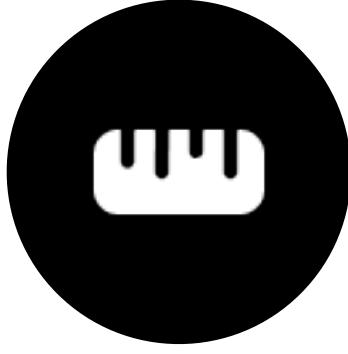
```
import Foundation
import UIKit

extension DesignSystem.Foundation(Icons {

    // MARK: - Business Icons
    enum Business {

        // MARK: - Cases
        case trendUp
        case chartsTab
        case chartPoint
        // MARK: - Properties
        var image: UIImage? {
            switch self {
            case .trendUp:
                return UIImage(named: "ic_home_user_card")
            case .chartsTab:
                return UIImage(named: "ic_product_details")
            case .chartPoint:
                return UIImage(named: "ic_charts_point")
            }
        }
    }
}
```

Foundation



Measurements

```
import Foundation

extension DesignSystem.Foundation.Spacing {

    // MARK: - VStack Padding
    enum Padding {
        // MARK: - Cases
        case top
        case bottom
        case leading
        case trailing

        // MARK: - Properties
        var space: CGFloat {
            switch self {
            case .top:
                return 16.0
            case .bottom:
                return 16.0
            case .leading:
                return 16.0
            case .trailing:
                return 16.0
            }
        }
    }
}
```

🧩 Why Use Enums in a Design System?

1. Structure 🏗️

2. Type Safety 🔒

3. Readability 👓

4. Maintainability 🔧

5. Consistency 🔄

```
extension Primary {
    case x50, x100, x500
    var color: UIColor {
        switch self {
            case .x50: return UIColor(named: "PrimaryX50")!
            case .x100: return UIColor(named: "PrimaryX100")!
            case .x500: return UIColor(named: "PrimaryX500")! // Updated value
        }
    }
}
```

```
@IBOutlet private weak var skipButton: UIButton! {
    didSet {
        skipButton.backgroundColor = DesignSystem.Foundation.Colors.Primary.x500.color
    }
}

@IBOutlet private weak var headTitleLabel: UILabel! {
    didSet {
        headTitleLabel.font = DesignSystem.Foundation.Typography.Heading.XH1.bold.font
    }
}
```

One more thing...

Design Tokens

What are Design Tokens?

Design token is a named value that represents a visual style or design property of your app

Colors



#2970FF

↓



Primary-500

↓



Buttons/Primary/Fill/Non Distractive/Normal

Typography

Font/Family/Inter

↓

Font/Weight

↓

Font/Size

↓

Font/Line-Height

↓

Font/Tracking

↓

Font/Paragraph-Spacing

↓

Buttons/Primary/Typography/Non Distractive/Normal

Spacing

Spacing/4px

↓

Spacing/1/4px

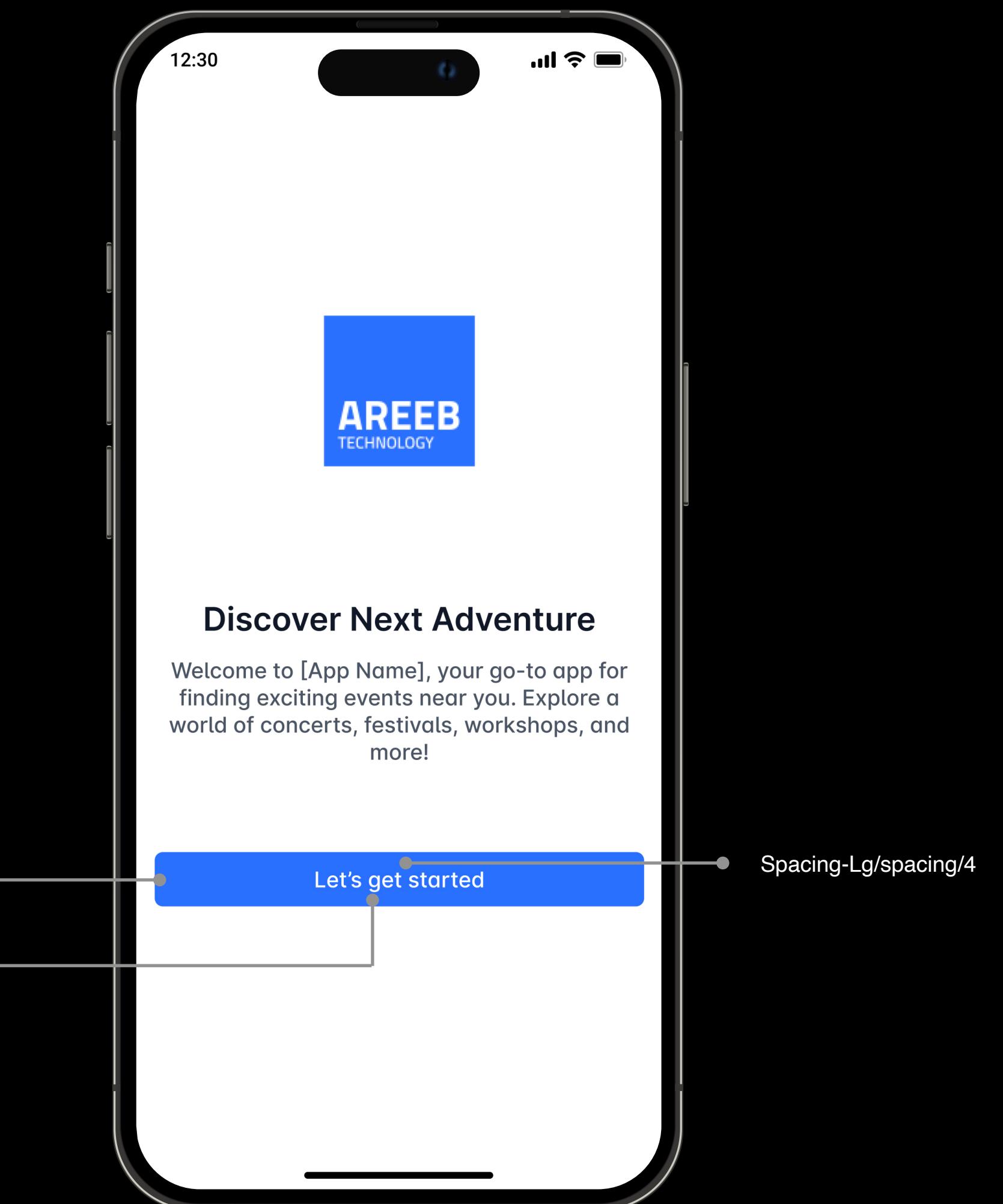
↓

Spacing-Lg/spacing/4

Design Tokens

```
struct DesignTokens {  
    struct Colors {  
        static let primaryButtonBackground = DesignSystem.Foundation.Colors.Primary.x500.color  
        static let primaryButtonText = DesignSystem.Foundation.Colors.Neutral.x500.color  
    }  
  
    struct Fonts {  
        static let primaryButtonFont = DesignSystem.Foundation.Typography.Heading.XH1.bold.font  
    }  
  
    struct Spacing {  
        static let buttonPadding: CGFloat = DesignSystem.Foundation.Spacing.Padding.leading.space  
        static let buttonCornerRadius: CGFloat = DesignSystem.Foundation.Spacing.Redius.buttons.space  
    }  
}
```

```
final class PrimaryButton: UIButton {  
  
    // MARK: - Init  
    public override init(frame: CGRect) {  
        super.init(frame: frame)  
  
        setupPrimaryButton()  
    }  
  
    public required init?(coder: NSCoder) {  
        super.init(coder: coder)  
  
        setupPrimaryButton()  
    }  
  
    // MARK: - Setup Primary Button  
    private func setupPrimaryButton() {  
        titleLabel?.font = DesignTokens.Fonts.primaryButtonFont  
        backgroundColor = DesignTokens.Colors.primaryButtonBackground  
        layer.cornerRadius = DesignTokens.Spacing.buttonCornerRadius  
    }  
}
```





Thank you



areebtechnology.com

All rights reserved 2024 © areebtechnology.com