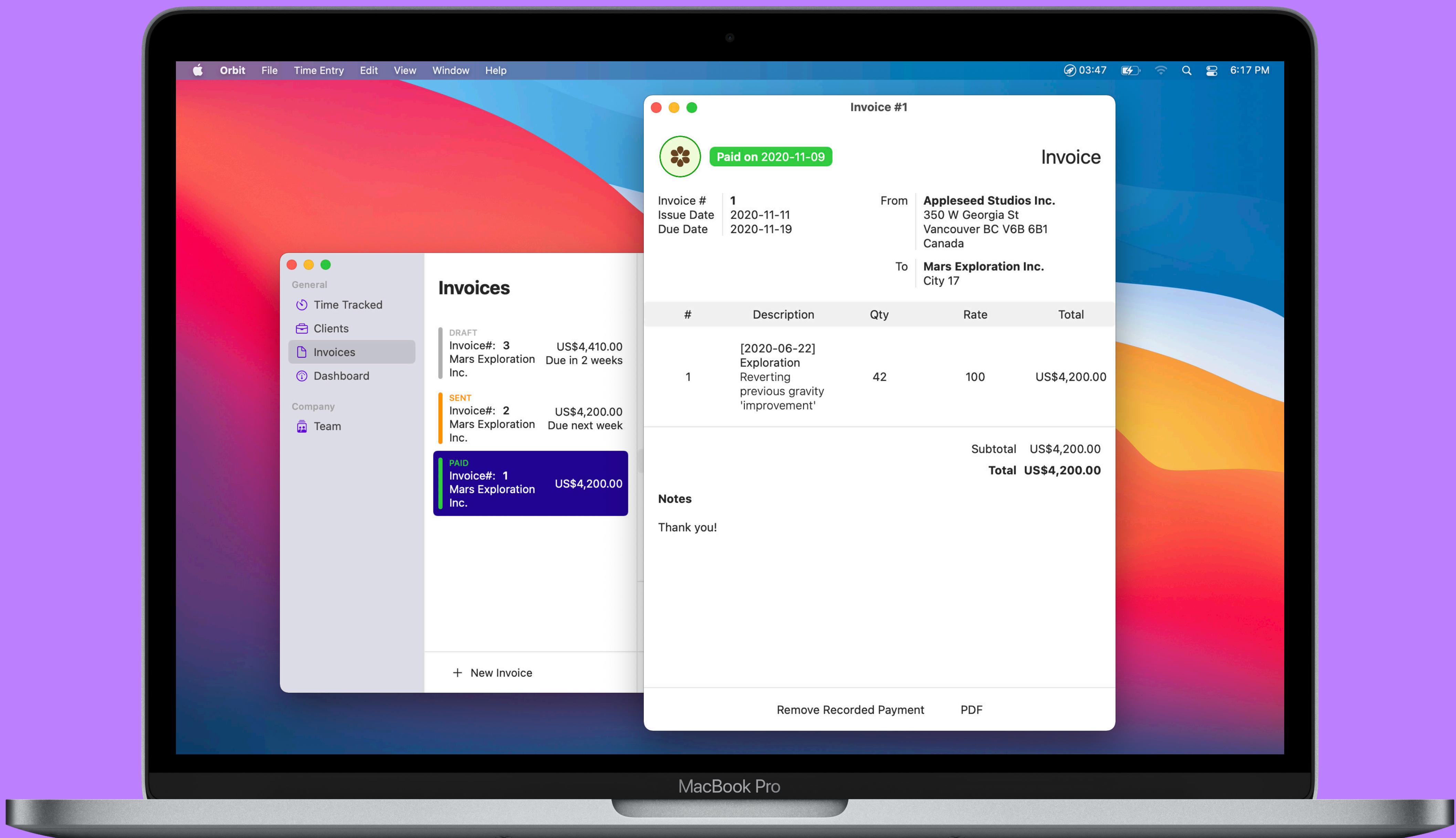


MALIN SUNDBERG – @MALINSUNDBERG

Multi-Platform Development, the SwiftUI Way



Hi, I'm Malin



The image displays three devices running the Orbit application, showcasing its cross-platform functionality.

Mac Desktop (Left):

- General:** Shows a sidebar with Time Tracked, Clients, Invoices (selected), and Dashboard.
- Invoices:** A list of invoices:
 - DRAFT:** Invoice# 3, US\$4,410.00, Mars Exploration Inc., Due in 2 weeks.
 - SENT:** Invoice# 2, US\$4,200.00, Mars Exploration Inc., Due next week.
 - PAID:** Invoice# 1, US\$4,200.00, Mars Exploration Inc.
- Buttons:** + New Invoice, Remove.

iPhone (Middle):

- Invoice #1:** Shows an invoice for Mars Exploration Inc. with a green button indicating it was paid on 2020-11-09.
- Notes:** Thank you!

iPad (Right):

- Orbit:** Shows a sidebar with Time Entries, Invoices (selected), and Dashboard (highlighted).
- Dashboard:** Displays time tracking data for the current week.
 - Total Time:** 31 hr, 22 min (9 hr per day).
 - Amount Earned (Split by Project Currency):** Earnings in CAD CA\$1,753.72 and Earnings in USD \$2,516.23.
 - Time Spent by Project:** A pie chart showing time spent across three projects:
 - Bear With Me (Wildlife Finder Inc.)
 - Lightweight Pizza Website (Mars Exploration Inc.)
 - Lightweight Pizza App (Mars Exploration Inc.)

The image displays the Orbit application across three different platforms: a Mac desktop, an iPhone X, and an iPhone 11 Pro. The Mac desktop shows the main dashboard with sections for Invoices, Time Entries, Clients, and Team. The iPhone X shows a detailed view of an invoice, and the iPhone 11 Pro shows a comprehensive dashboard with time tracking and reporting.

Mac Desktop Dashboard:

- Invoices:** Shows three invoices: DRAFT (Invoice# 3), SENT (Invoice# 2), and PAID (Invoice# 1).
- Time Entries:** Shows a list of clients and team members.
- Clients:** Shows a list of clients.
- Team:** Shows a list of team members.

iPhone X - Invoice View:

Invoice #1
Paid on 2020-11-09

iPhone 11 Pro - Dashboard:

9:41 Thu Oct 1

Orbit

- Time Entries
- Invoices
- Dashboard**
- Clients

Company

- Team

September 27 – Today

Total Time
31 hr, 22 min

Time per Day

Amount Earned (Split by Project Currency)
Based on time tracked for projects with an hourly rate

Earnings in CAD
CA\$1,753.72

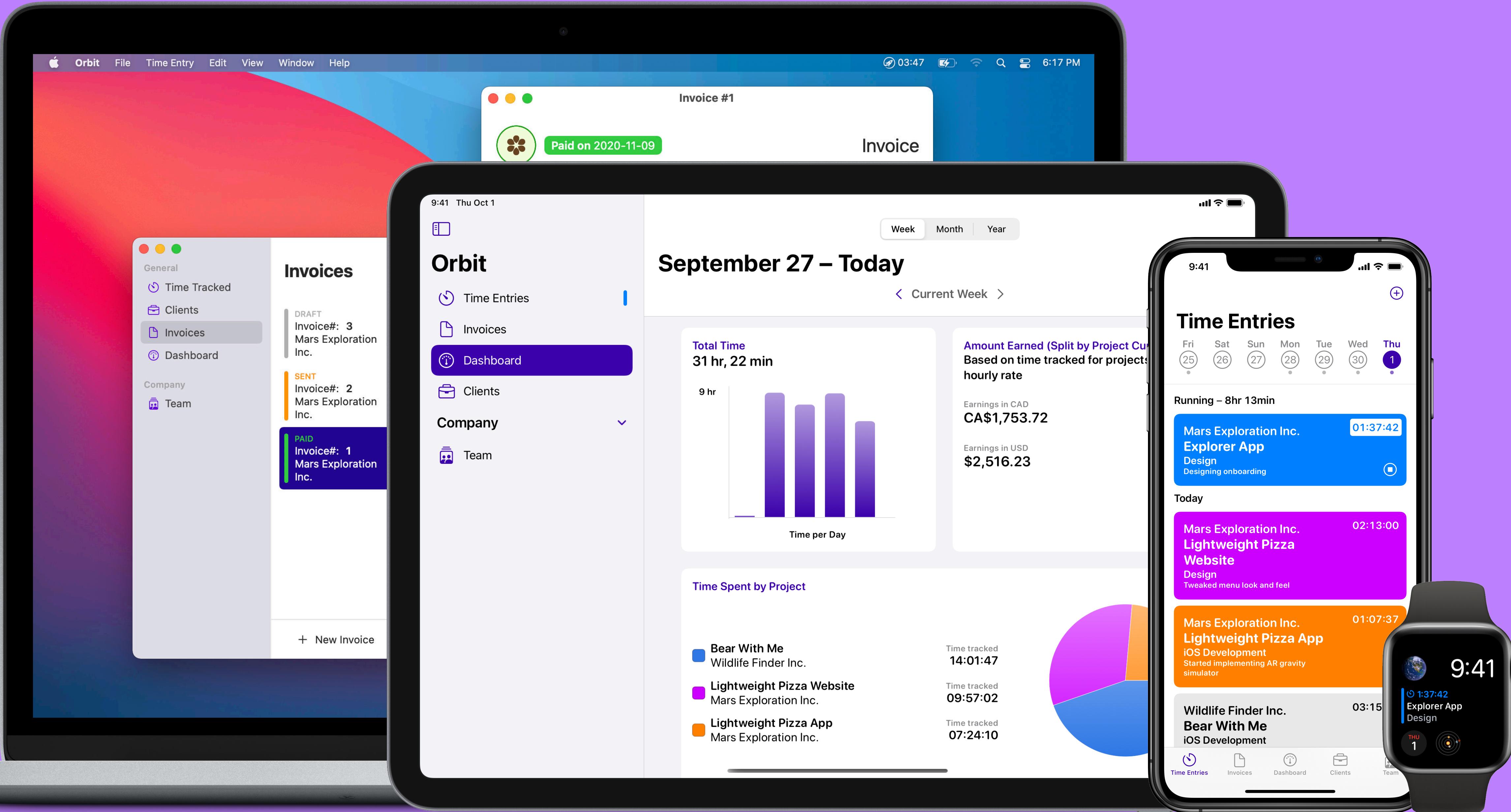
Earnings in USD
\$2,516.23

Time Spent by Project

Project	Time Tracked
Bear With Me (Wildlife Finder Inc.)	14:01:47
Lightweight Pizza Website (Mars Exploration Inc.)	09:57:02
Lightweight Pizza App (Mars Exploration Inc.)	07:24:10

Time Entries (iPhone 11 Pro):

Date	Client	Project	Duration
Fri 25	Mars Exploration Inc.	Explorer App	01:37:42
Today	Mars Exploration Inc.	Lightweight Pizza Website	02:13:00
Today	Mars Exploration Inc.	Lightweight Pizza App	01:07:37
	Wildlife Finder Inc.	Bear With Me	03:15:00



SwiftUI

A Primary Tool for Development on all
Apple Platforms

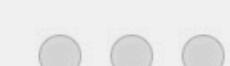
Some Hints

Some Examples





Shortcut Name



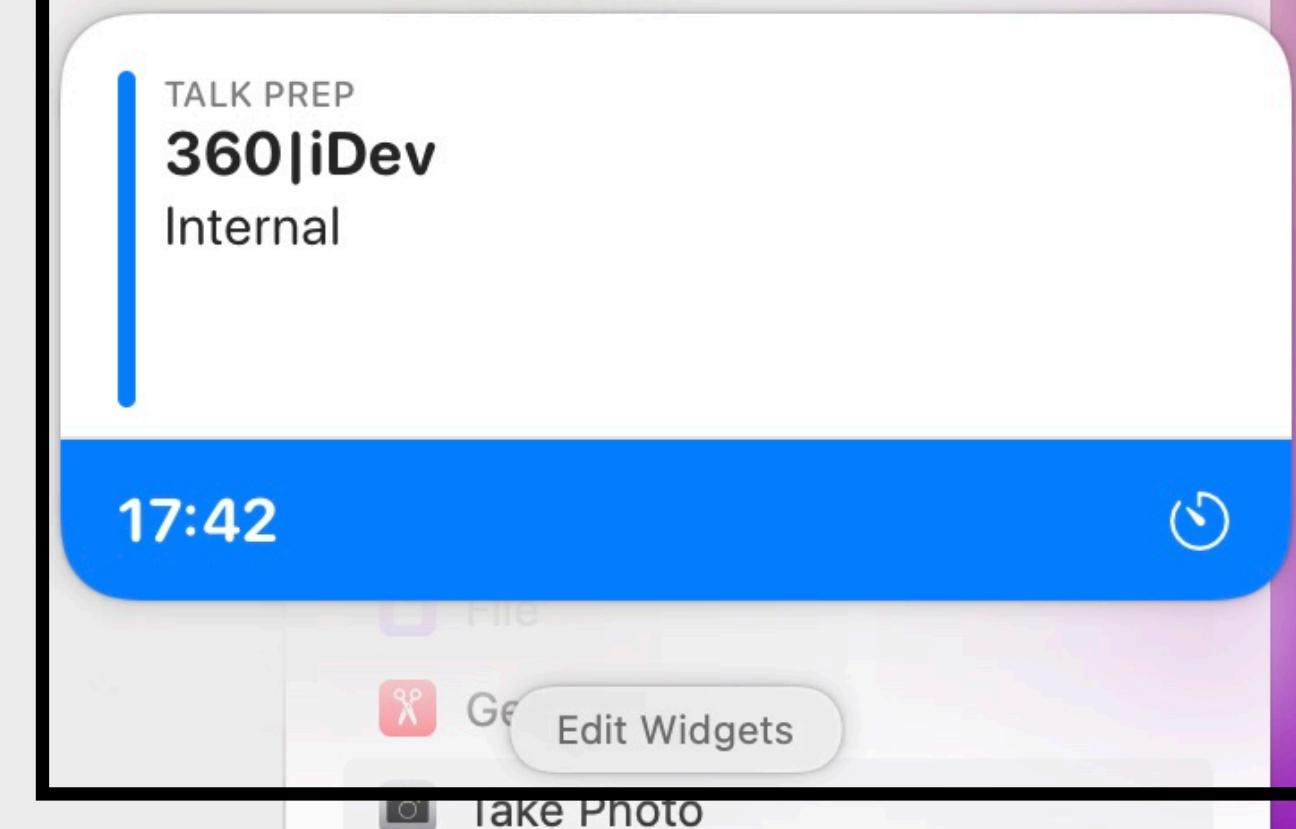
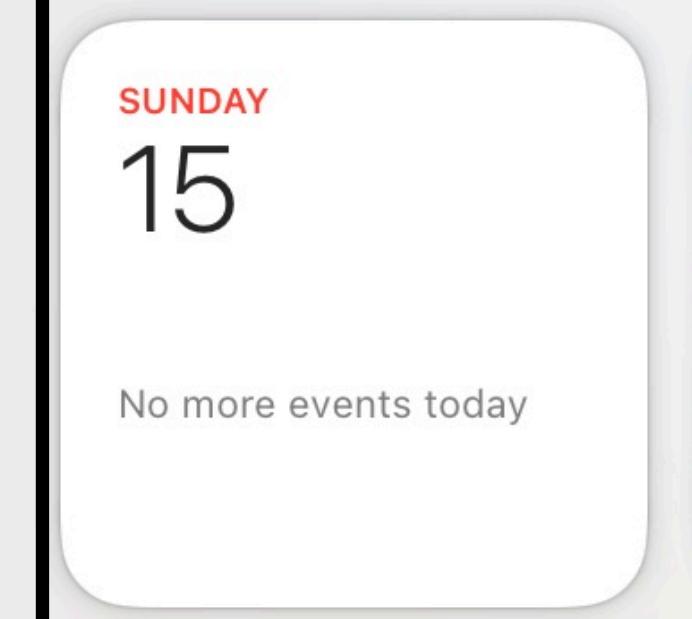
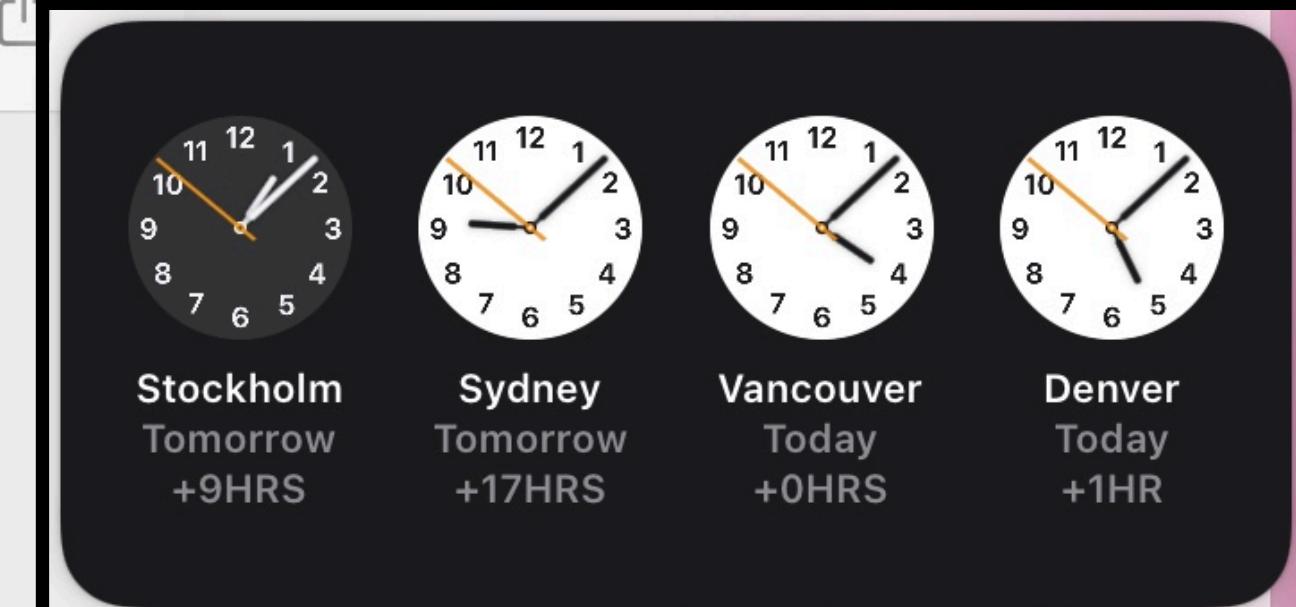
My Shortcuts



Folders



Add actions from the right to build your shortcut.



SwiftUI

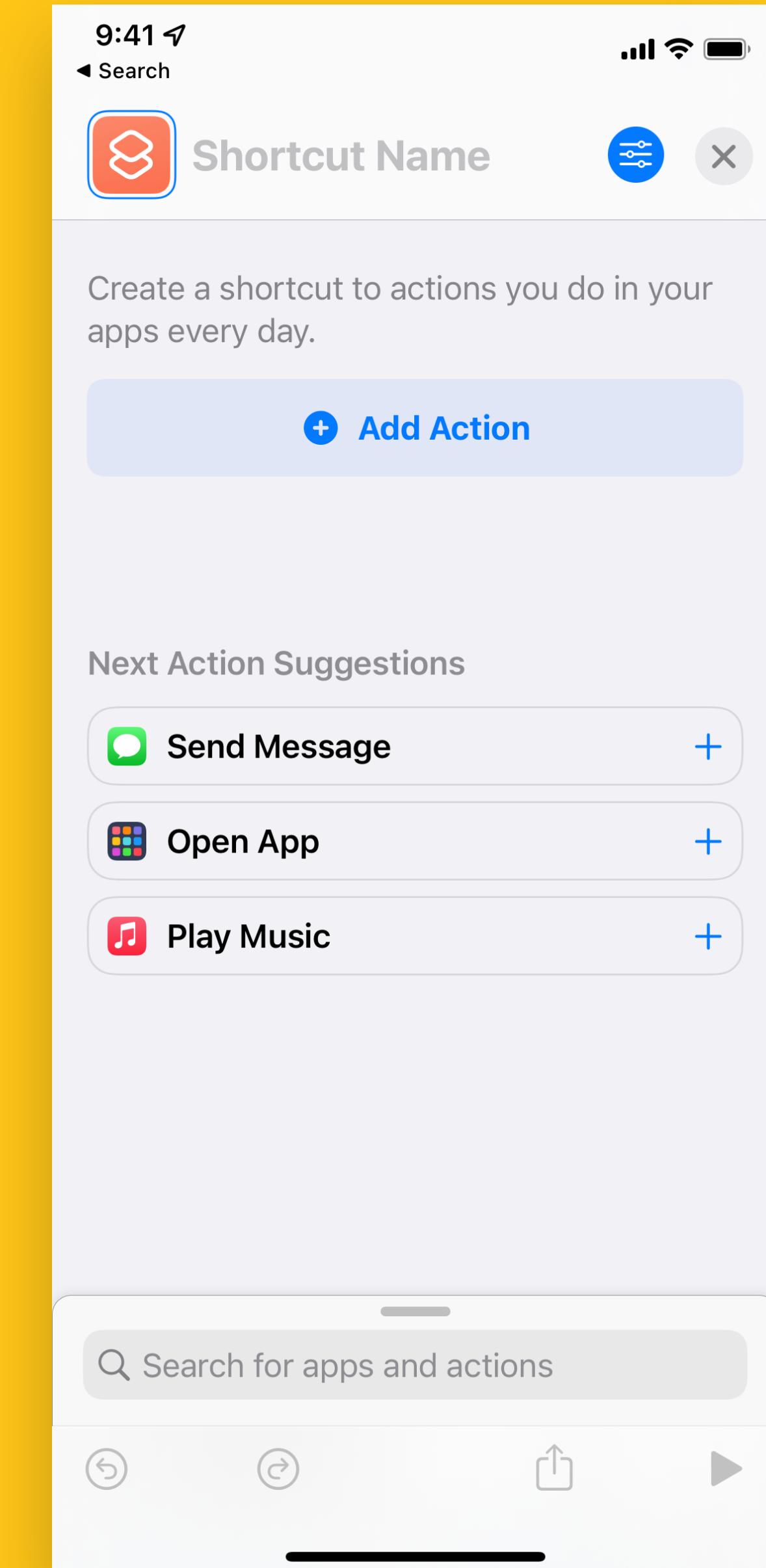
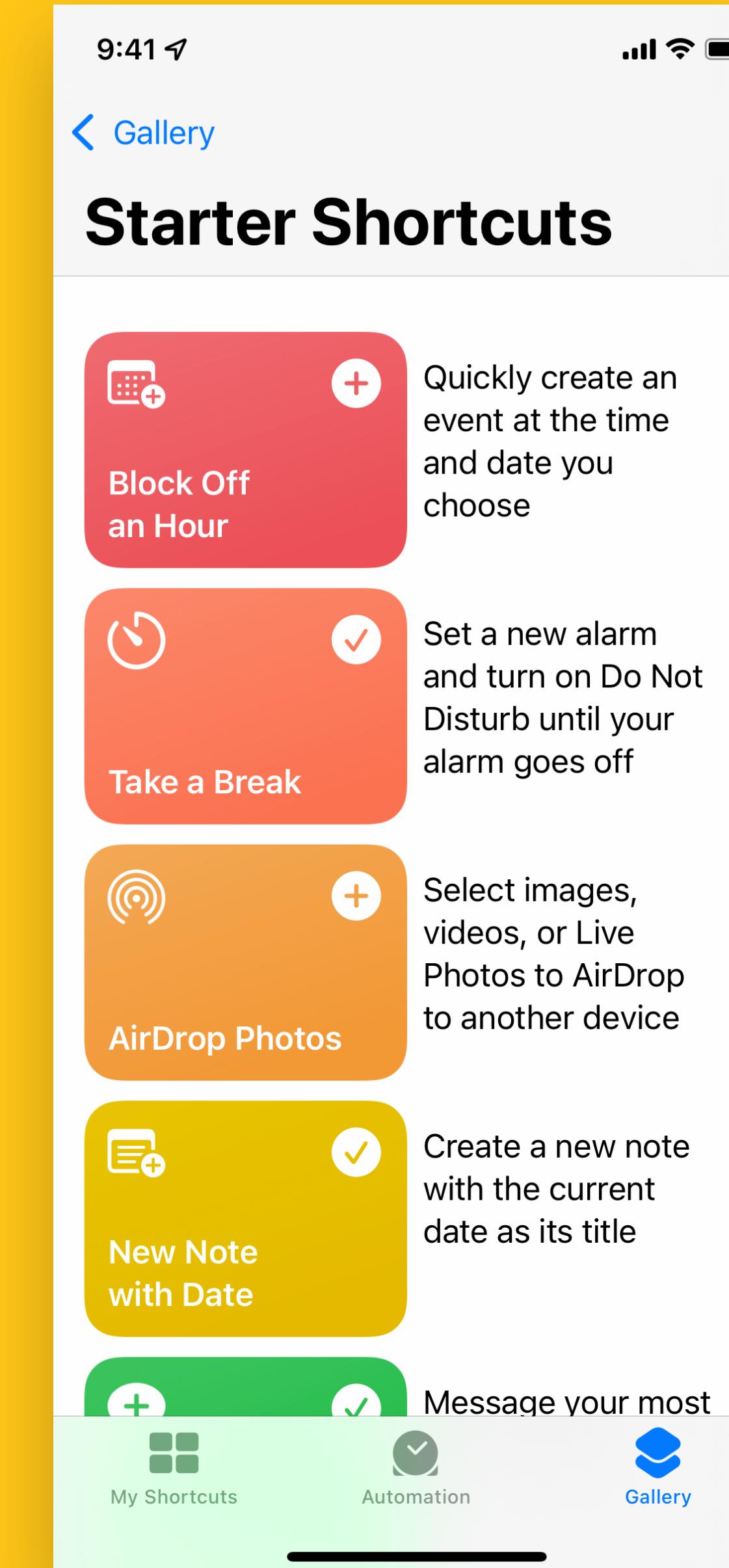
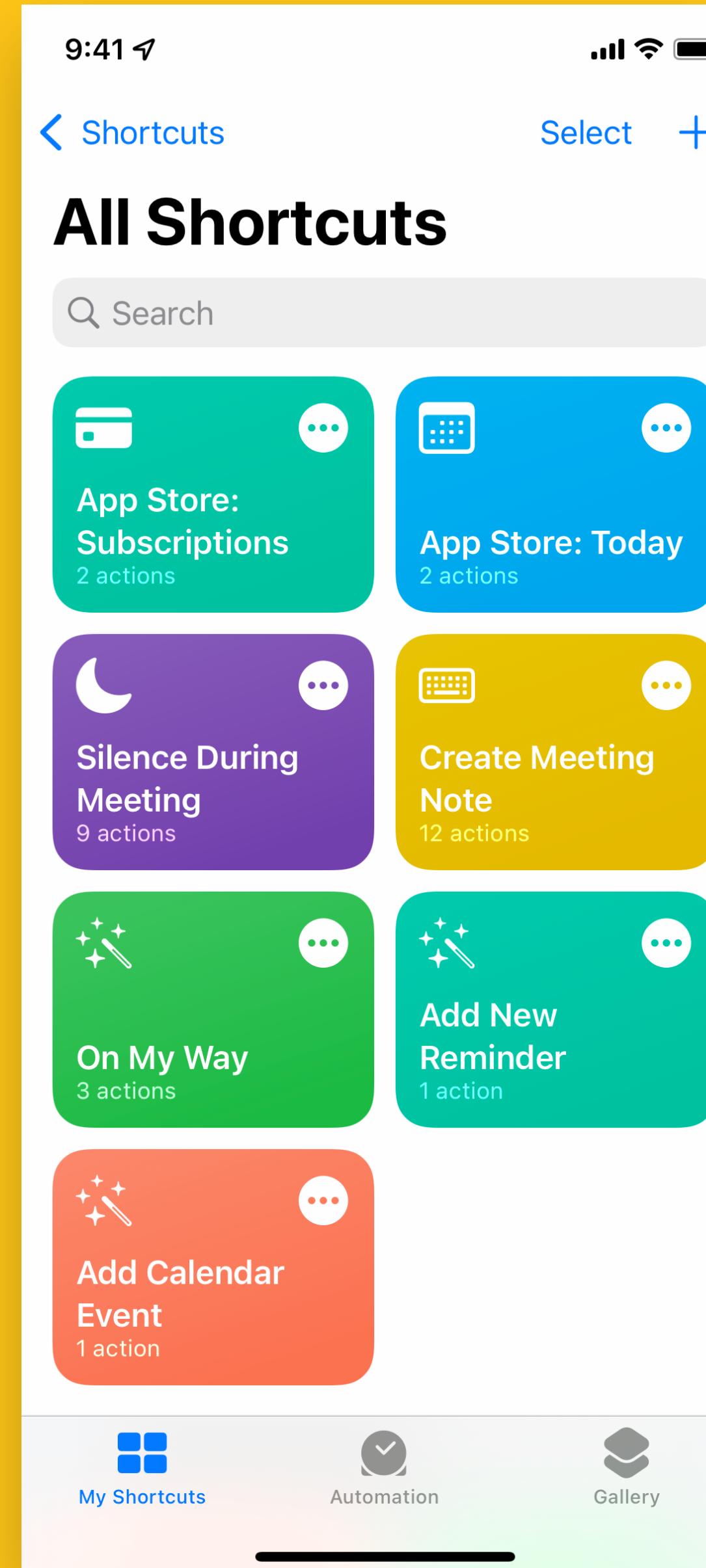
**Bringing Development for Different Platforms
Closer Together**

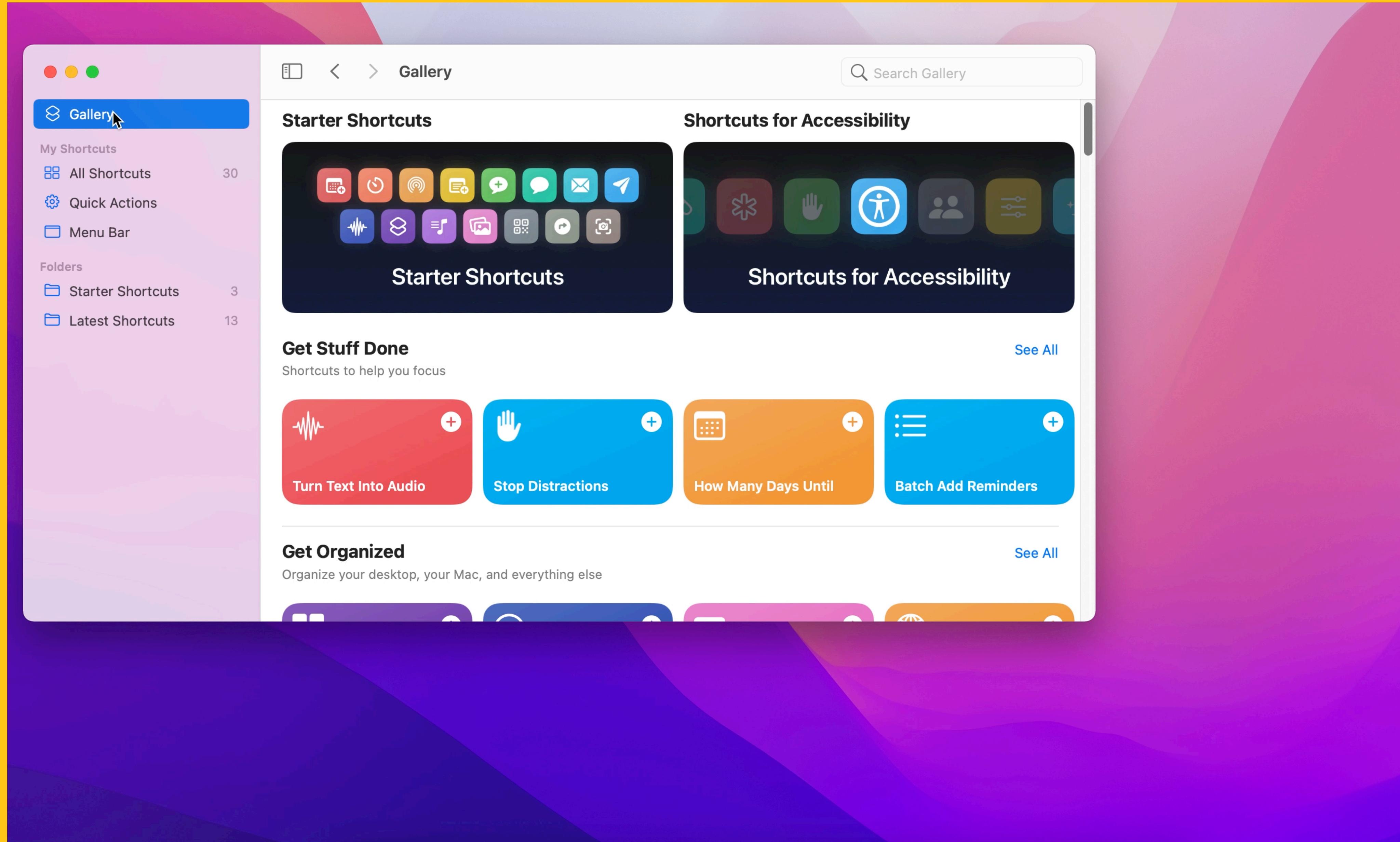
Why Is This Interesting?

“Multi-Platform Development”

Write It Once and Run It Anywhere

Learn It Once and Use It Everywhere





UI Components

Using SwiftUI for Multi-Platform Development

Choose a template for your new project:

Multiplatform

iOS

macOS

watchOS

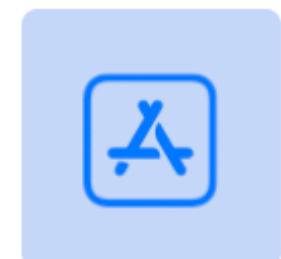
tvOS

DriverKit

Other

Filter

Application



App



Document App



Game



Safari Extension
App

Framework & Library



Swift Package

Cancel

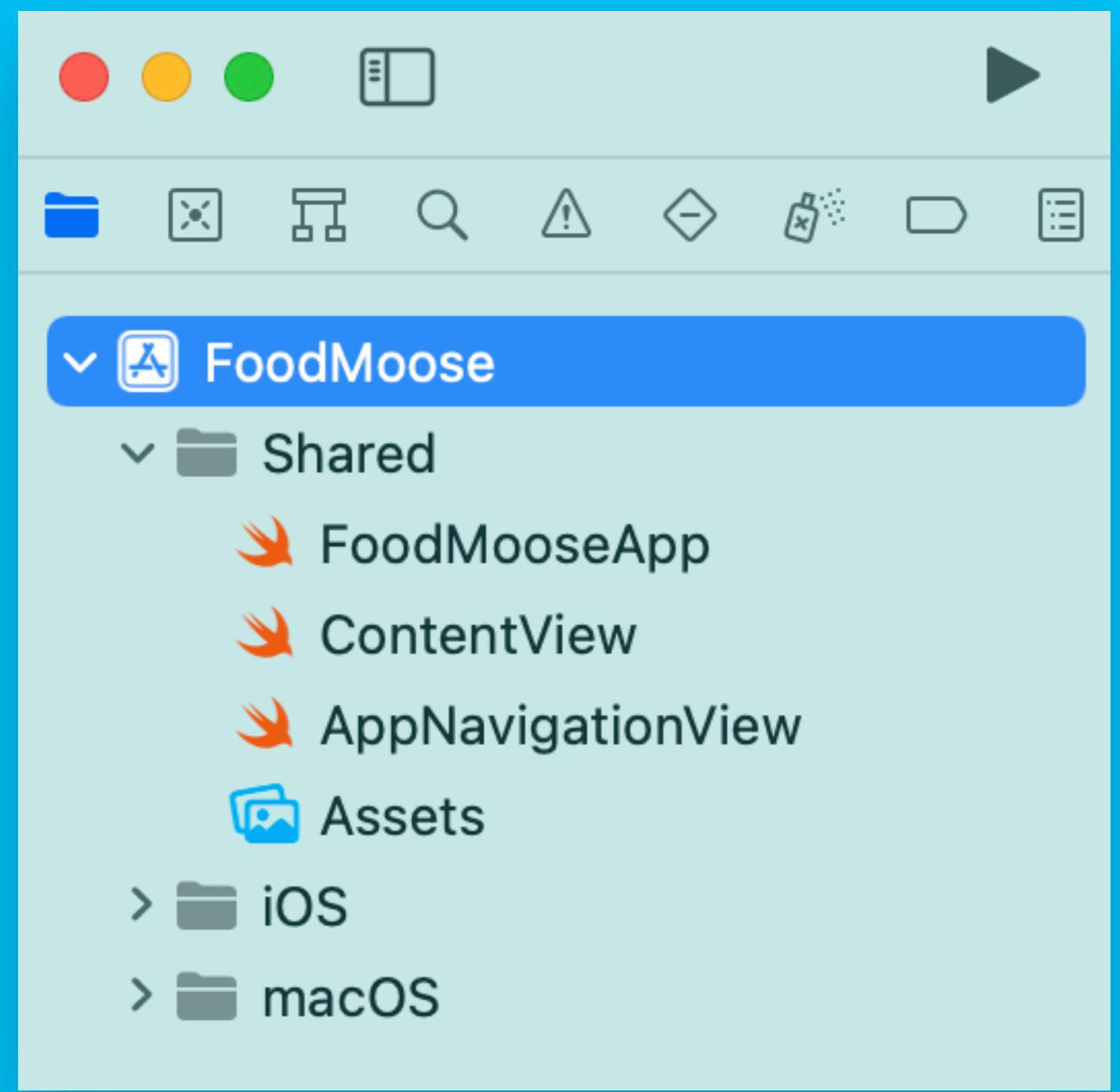
Previous

Next

```
@main
struct FoodMooseApp: App{
    var body: some Scene {
        WindowGroup {
            ContentView()
        }
    }
}
```



```
struct ContentView: View {  
    var body: some View {  
        AppNavigationView( )  
    }  
}
```

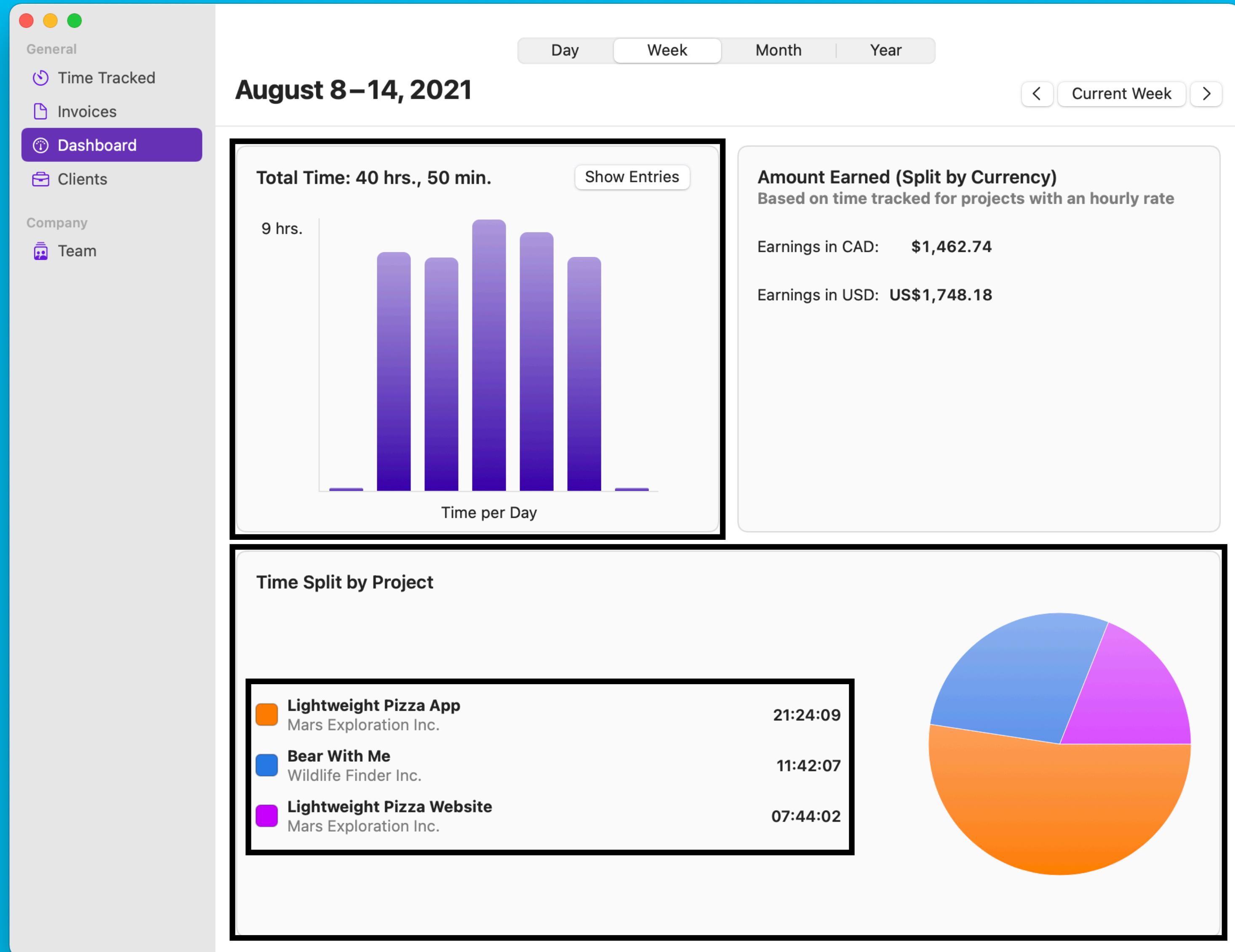


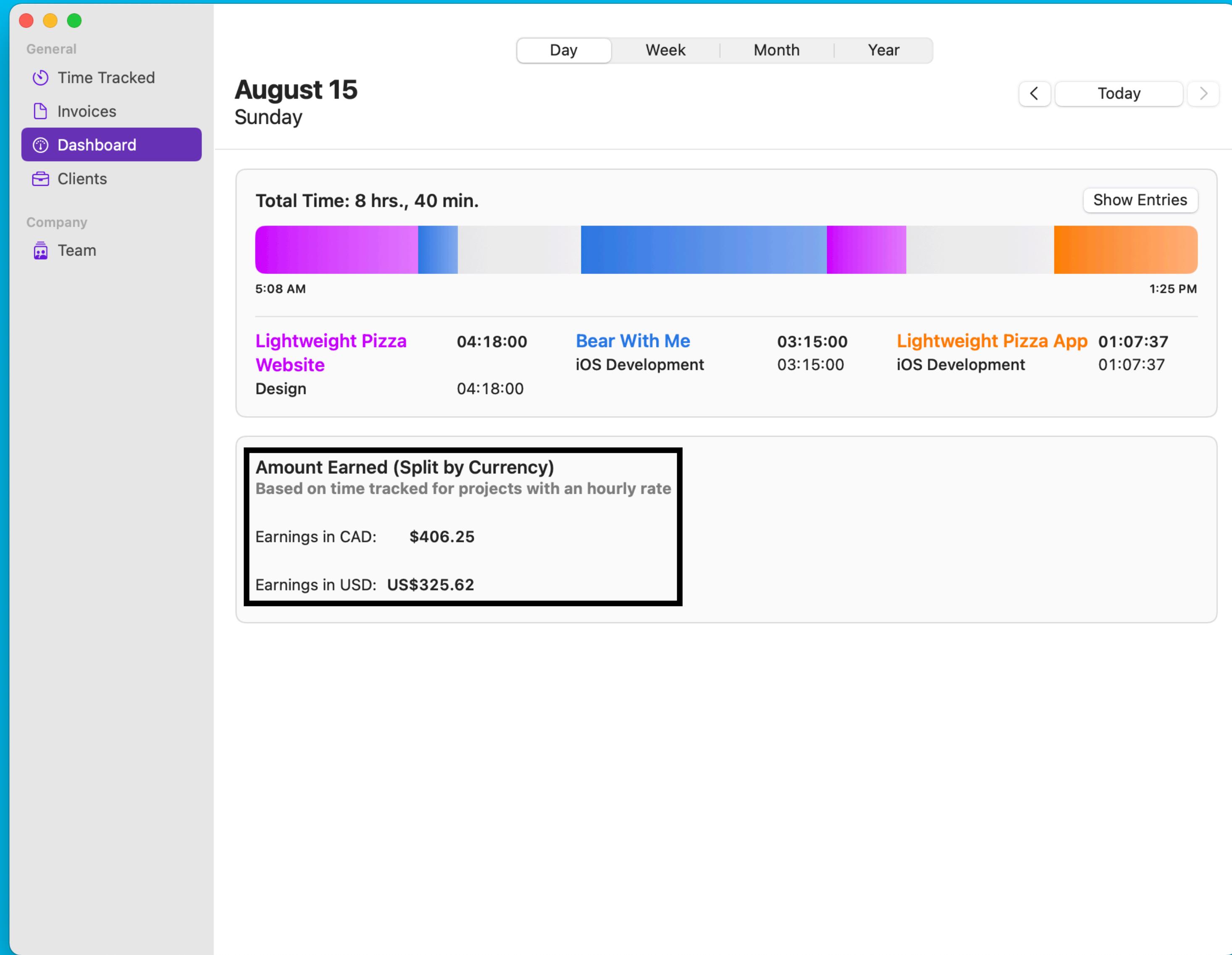
Focus on One Platform

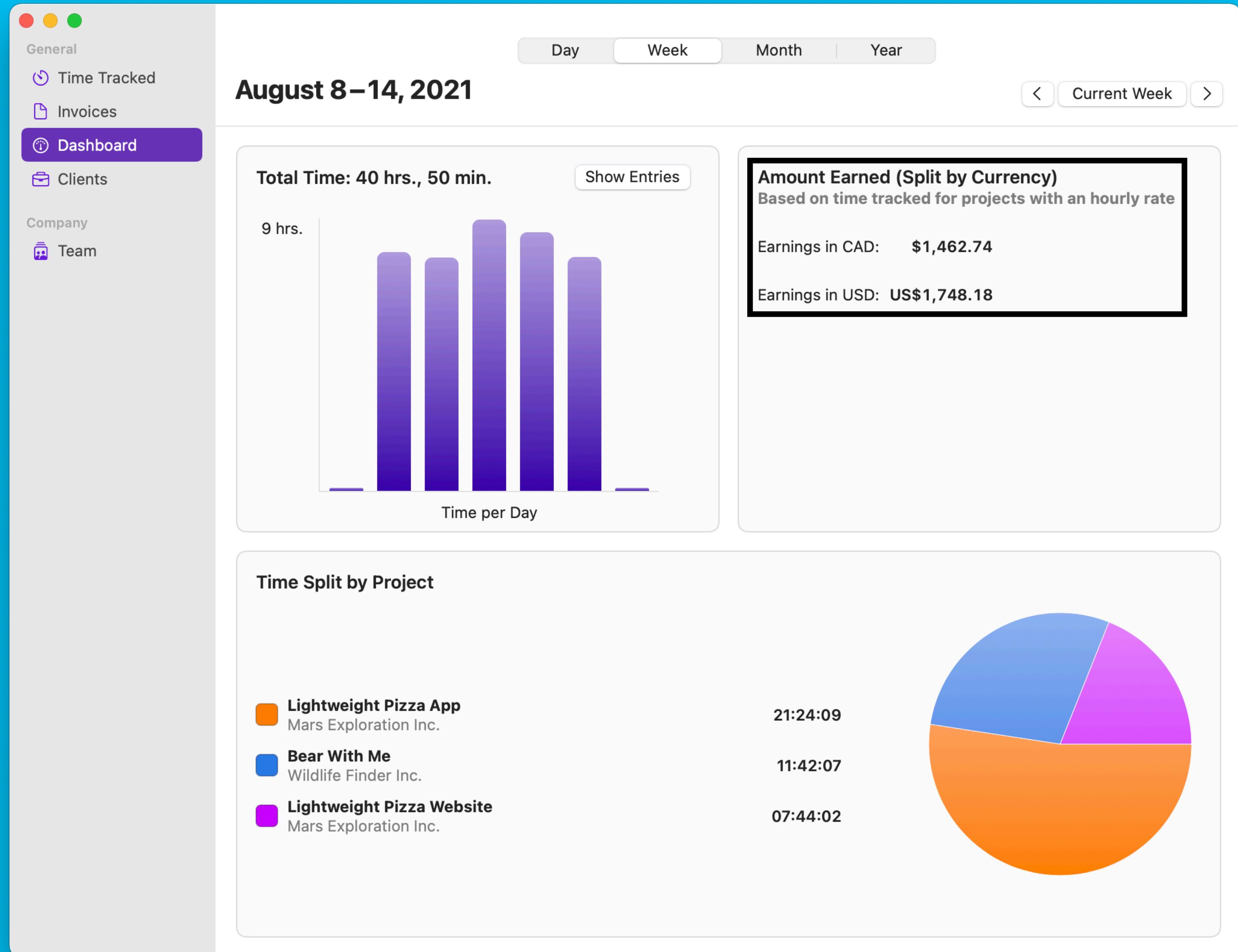
Which Platform Do We Start With?

Structuring our UI Code on a Single Platform

Components



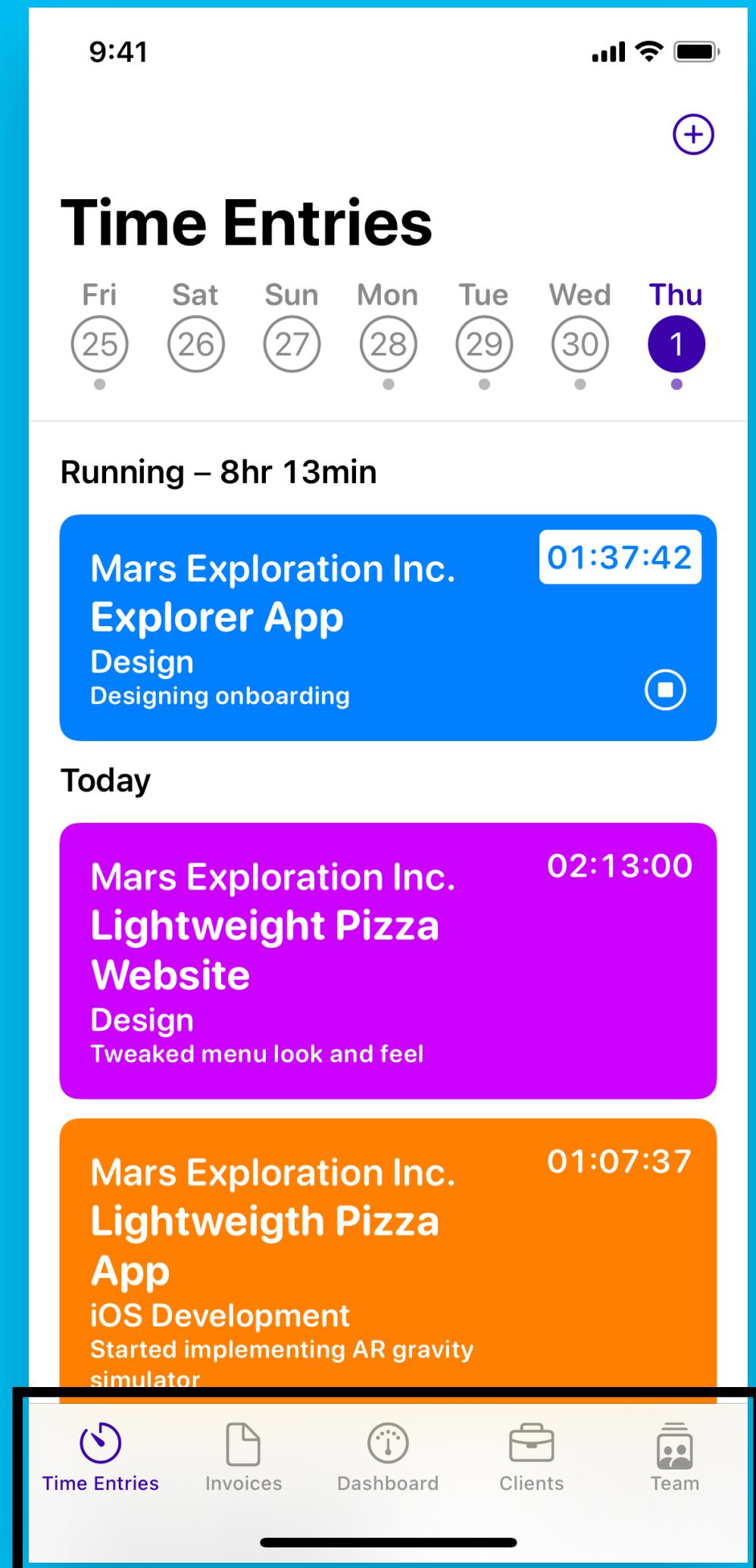




Moving to a Second Platform

Structuring Multi-Platform Code

Separate Flows



The screenshot shows the Time Entries application window on a Mac. On the left is a sidebar with "General" and "Time Tracked" selected, followed by "Clients", "Invoices", "Dashboard", "Company", and "Team". The main area shows a timeline from Saturday, Nov 14 at 00:00:00 to Friday, Nov 20 at 08:27:19. Each day has a timestamp. Below the timeline are four entries:

- Sat Nov 14**: Mars Exploration Inc. - **Explorer App** (Design, 01:51:42, ongoing)
- Sun Nov 15**: Mars Exploration Inc. - **Lightweight Pizza Website** (Design, 02:13:00, completed)
- Mon Nov 16**: Mars Exploration Inc. - **Lightweight Pizza App** (iOS Development, 01:07:37, completed)
- Tue Nov 17**: Wildlife Finder Inc. - **Bear With Me** (iOS Development, 03:15:00, completed)

On the right side of the timeline, there are "Stop" and "Resume" buttons for each day, and a "New Time Entry" button at the bottom right.

9:41 Thu Oct 1

Orbit

Time Entries

Invoices

Dashboard

Clients

Company

Team

Time Entries

Fri (25) Sat (26) Sun (27) Mon (28) Tue (29) Wed (30) Thu (1)

Running – 8hr 13min

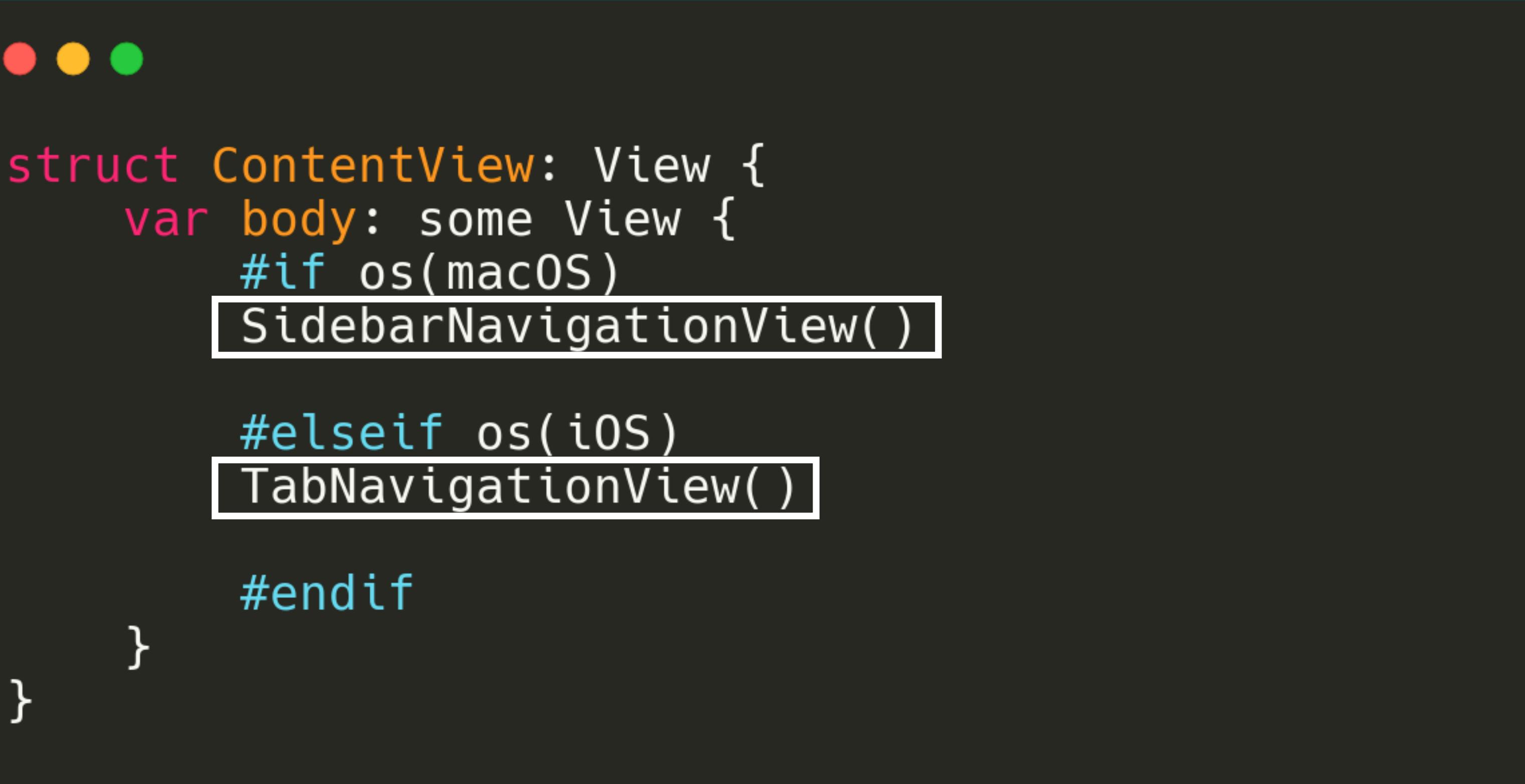
Mars Exploration Inc.
Explorer App
Design
Designing onboarding 01:37:42

Mars Exploration Inc.
Lightweight Pizza Website
Design
Tweaked menu look and feel 02:13:00

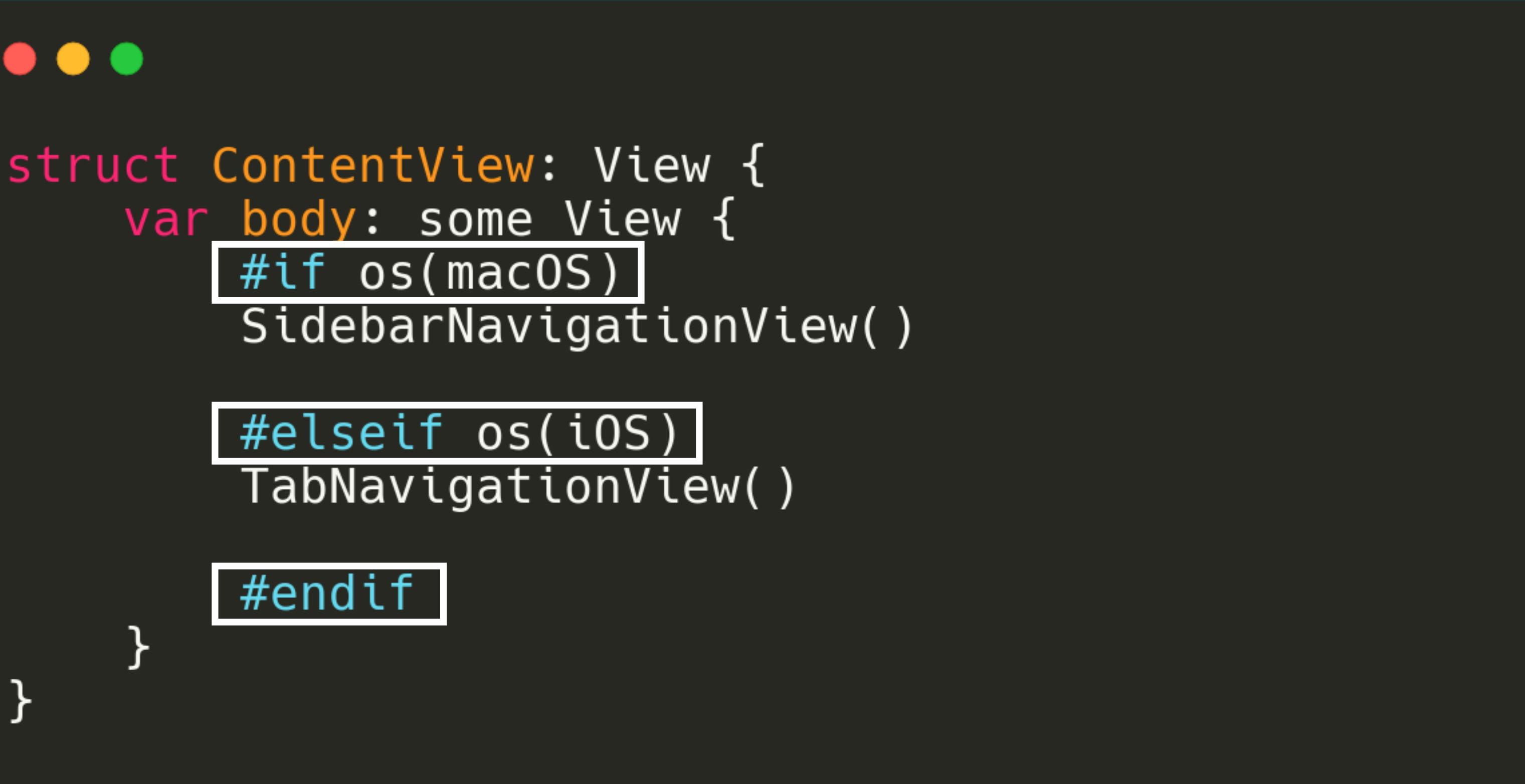
Mars Exploration Inc.
Lightweighth Pizza App
iOS Development
Started implementing AR gravity simulator 01:07:37

Wildlife Finder Inc.
Bear With Me
iOS Development
Continued implementing explore view 03:15:00

The screenshot shows the Orbit mobile application interface. The top navigation bar displays the date as 'Thu Oct 1' and the time as '9:41'. The sidebar on the left contains links for 'Time Entries' (highlighted with a blue bar), 'Invoices', 'Dashboard', 'Clients', 'Company' (with a dropdown arrow), and 'Team'. The main content area is titled 'Time Entries' and shows a weekly calendar from Friday to Thursday. Below the calendar, a summary states 'Running – 8hr 13min'. Four time entries are listed: 1. Mars Exploration Inc. - Explorer App (Designing onboarding, duration 01:37:42). 2. Mars Exploration Inc. - Lightweight Pizza Website (Design, Tweaked menu look and feel, duration 02:13:00). 3. Mars Exploration Inc. - Lightweighth Pizza App (iOS Development, Started implementing AR gravity simulator, duration 01:07:37). 4. Wildlife Finder Inc. - Bear With Me (iOS Development, Continued implementing explore view, duration 03:15:00). Each entry includes a small circular icon with a plus sign in the top right corner.



```
struct ContentView: View {
    var body: some View {
        #if os(macOS)
        SidebarNavigationView( )
        #elseif os(iOS)
        TabNavigationView( )
        #endif
    }
}
```



```
struct ContentView: View {
    var body: some View {
        #if os(macOS)
        SidebarNavigationView( )
        #elseif os(iOS)
        TabNavigationView( )
        #endif
    }
}
```



```
struct ContentView: View {
    #if os(iOS)
        @Environment(\.horizontalSizeClass) var horizontalSizeClass
    #endif

    var body: some View {
        #if os(macOS)
            SidebarNavigationView()
        #elseif os(iOS)
            if horizontalSizeClass == .compact {
                TabNavigationView()
            } else {
                SidebarNavigationView()
            }
        #endif
    }
}
```



```
struct ContentView: View {
    #if os(iOS)
    @Environment(\.horizontalSizeClass) var horizontalSizeClass
    #endif

    var body: some View {
        #if os(macOS)
        SidebarNavigationView()
        #elseif os(iOS)
        if horizontalSizeClass == .compact {
            TabNavigationView()
        } else {
            SidebarNavigationView()
        }
        #endif
    }
}
```

Same Flow, Different Looks

Compiler Directives



```
#if os(macOS)  
#elseif os(iOS)  
#endif
```

Compiler Directives



```
var body: some View {
    Text("Hello, 360|iDev!")
        #if os(iOS)
            .font(.title)
        #elseif os(macOS)
            .font(.headline)
        #endif
}
```



```
struct DashboardGraph: View {
    let chartValues: [ChartData]

    @State private var selectedValue: ChartData?

    var body: some View {
        HStack(alignment: .bottom) {

            ForEach(chartValues) { chartValue in
                GraphChartBar(value: chartValue)
                    .opacity(selectedValue == chartValue ? 1 : 0.5)
            }
        }
    }
}
```



```
GraphChartBar(value: chartValue)
    .opacity(selectedValue == chartValue ? 1 : 0.5)
#if os(iOS)
    .onTapGesture {
        withAnimation {
            updateSelectionState(for: chartValue)
        }
    }
#elseif os(macOS)
    .onHover { isHovering in
        withAnimation {
            updateSelectionState(for: chartValue, isHovering: isHovering)
        }
    }
#endif
```

View Modifiers


```
var body: some View {
    Text("Hello, 360|iDev!")
        #if os(iOS)
            .font(.title)
        #elseif os(macOS)
            .font(.headline)
        #endif
}
```



```
var body: some View {  
    Text("Hello, 360|iDev!")  
        .textWithBackground()  
}
```



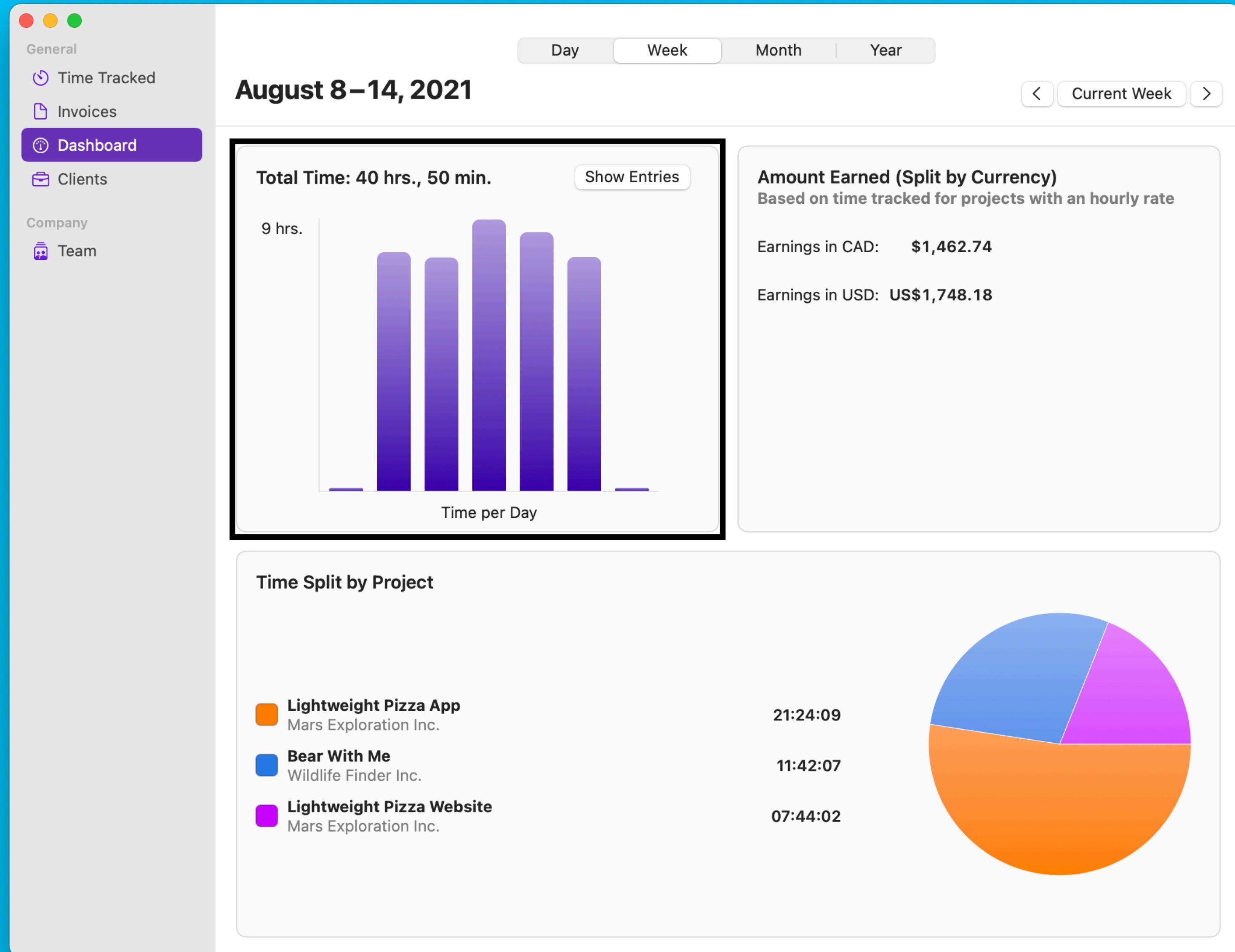
```
extension Text {
    func textWithBackground( ) -> some View {
        #if os(iOS)
        modifier(TextWithBackground( ))
        #elseif os(macOS)
        modifier(SubtleTextWithBackground( ))
        #endif
    }
}
```

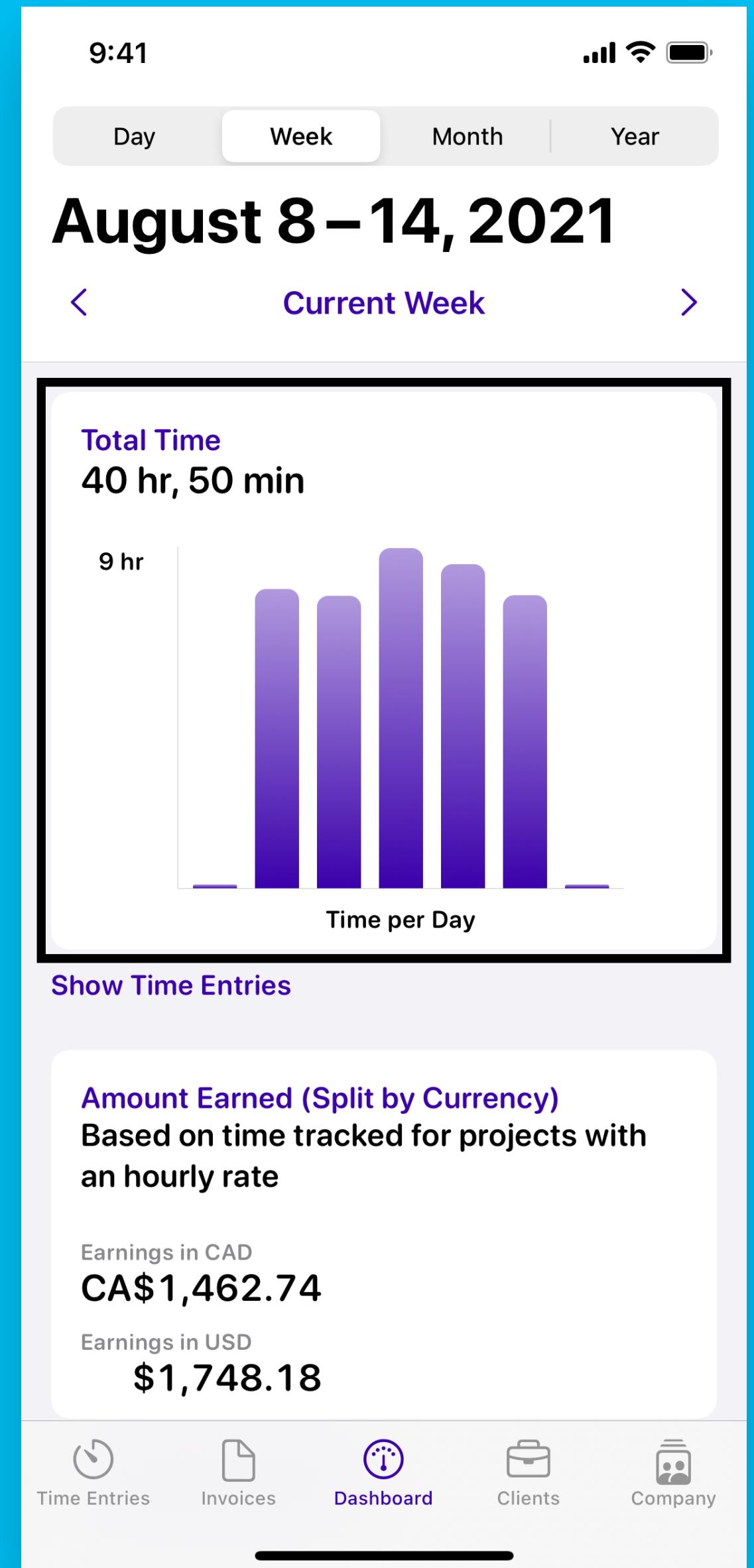


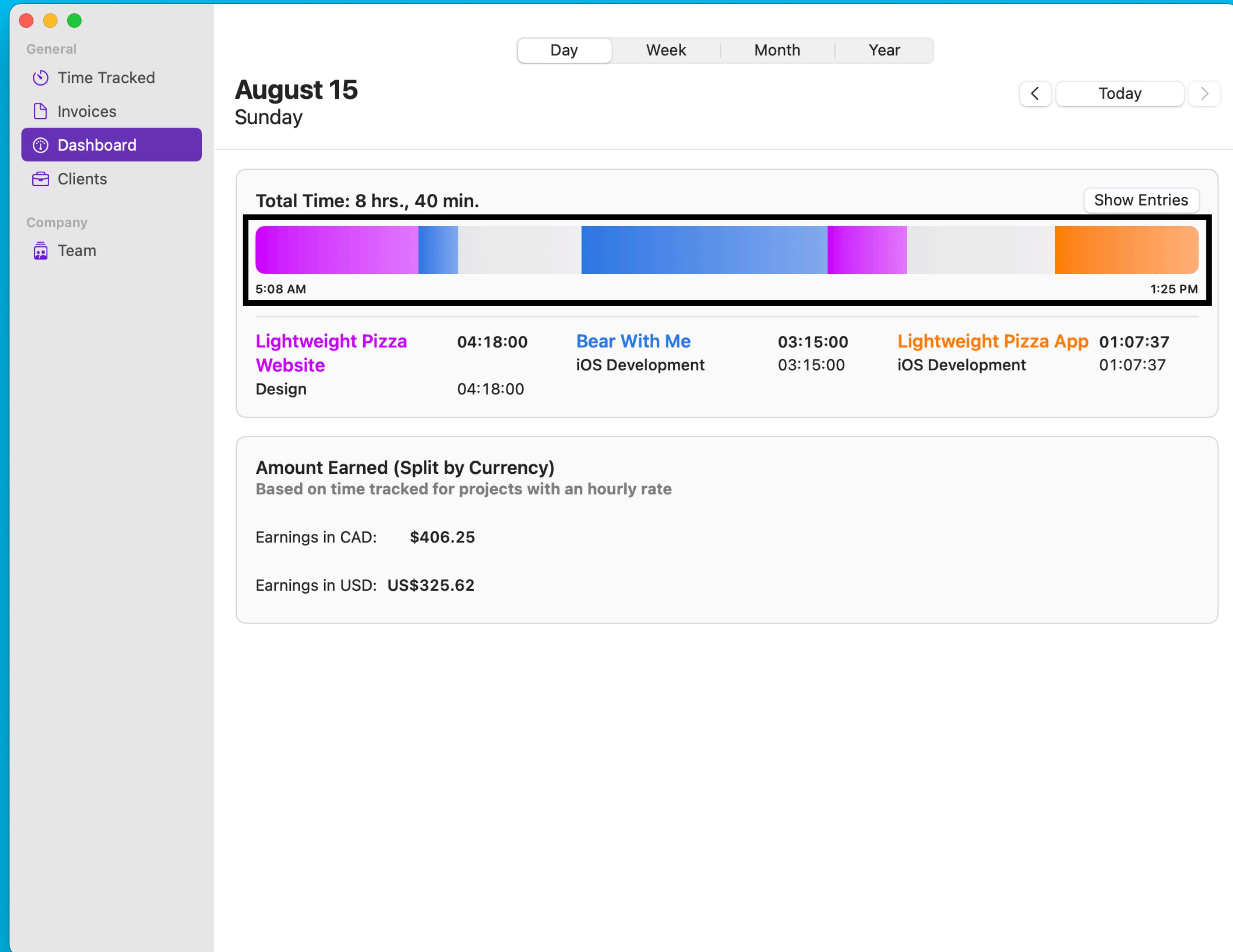
```
struct TextWithBackground: ViewModifier {  
    func body(content: Content) -> some View {  
        content  
            .font(.title)  
            .foregroundColor(.white)  
            .padding(.vertical, 2)  
            .padding(.horizontal, 12)  
            .background(RoundedRectangle(cornerRadius: 5, style: .continuous).fill(.purple))  
    }  
}
```

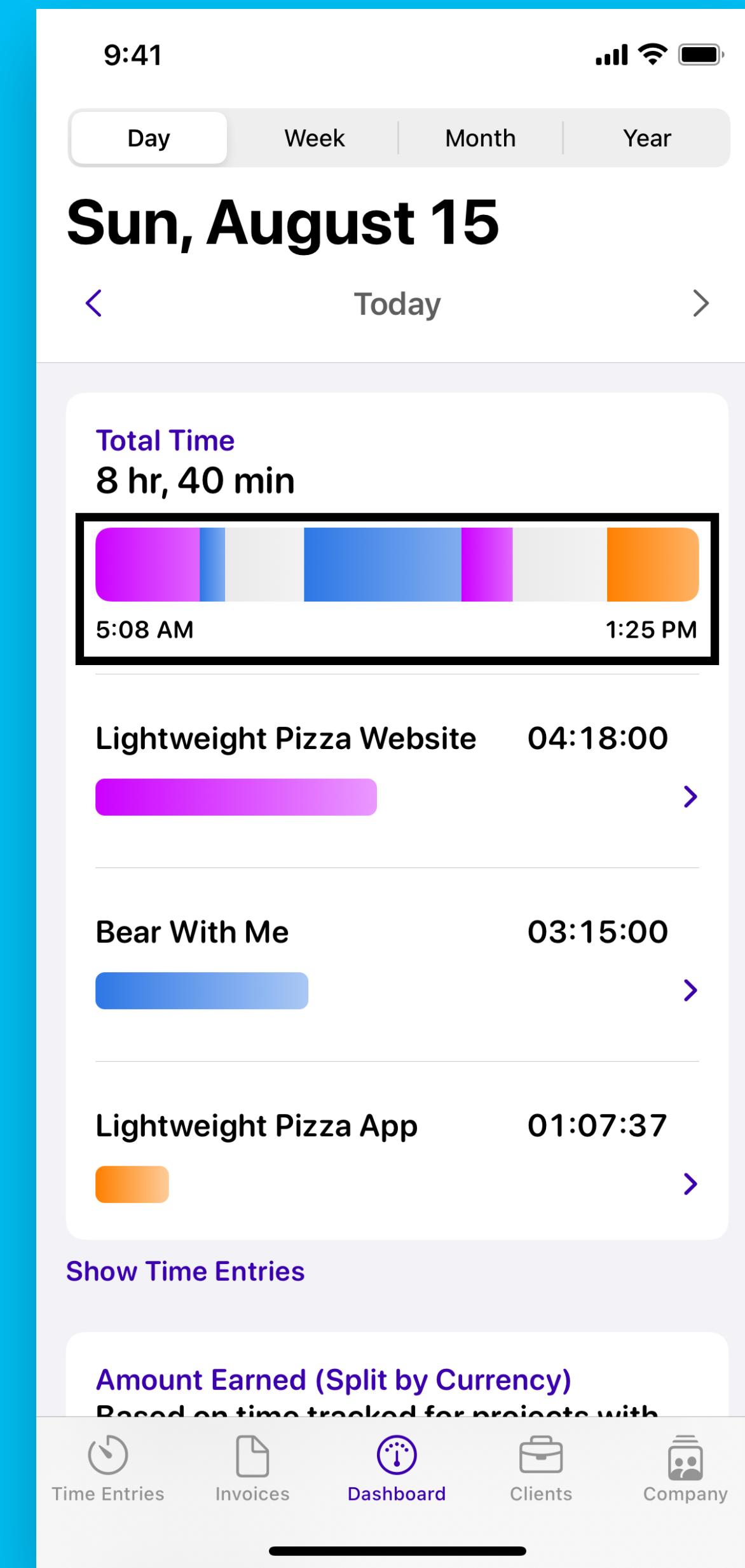
Which Technique Do We Choose?

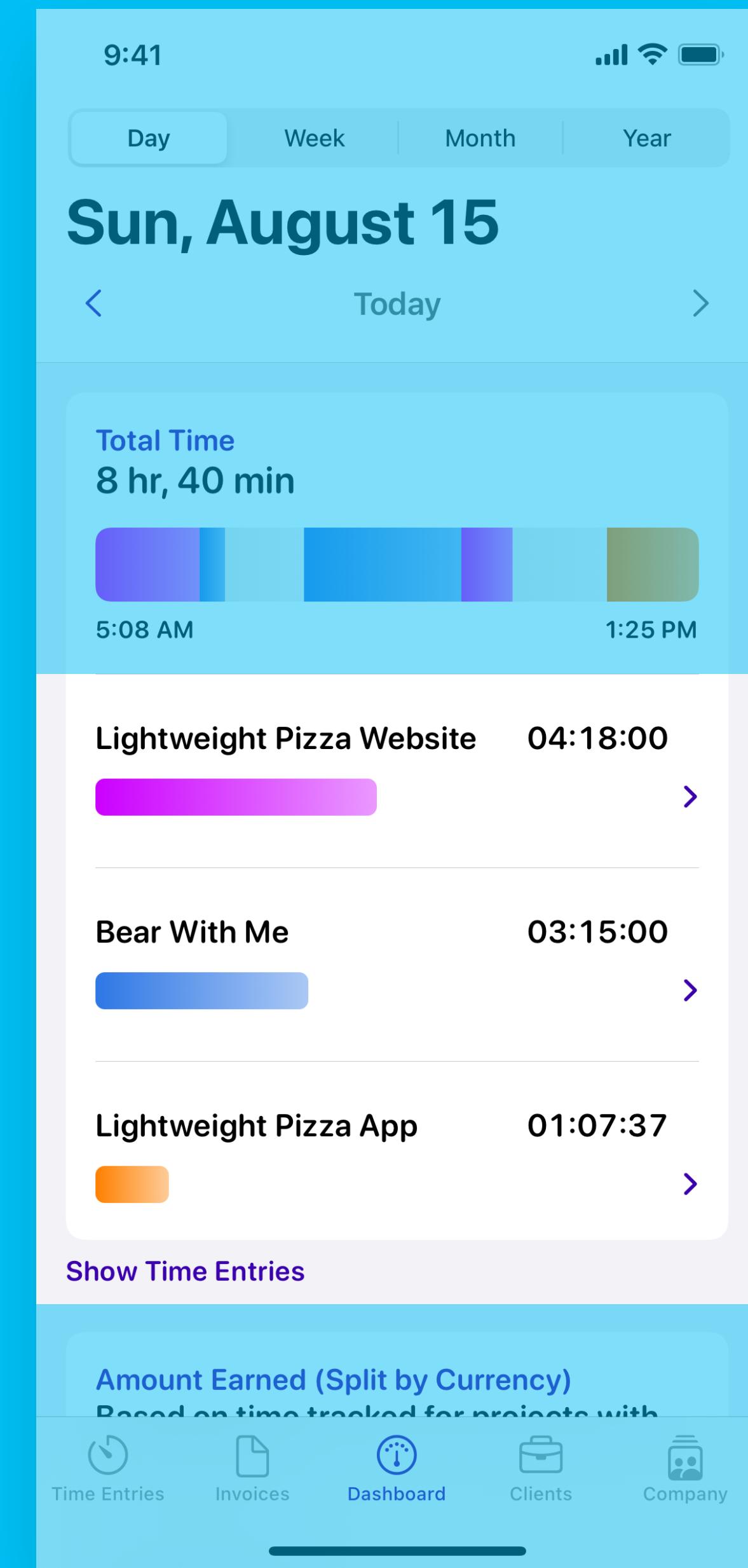
Reusing UI Components

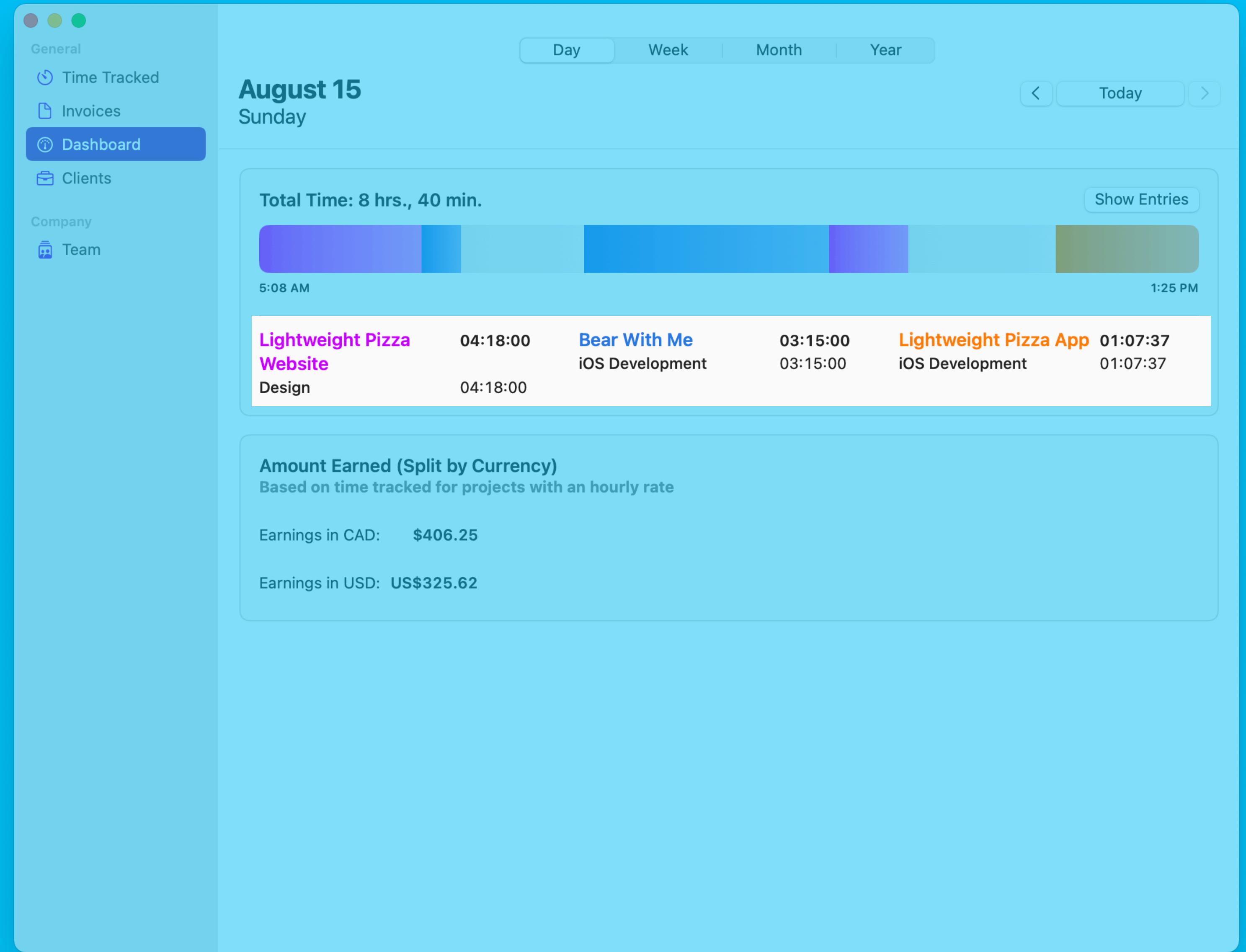






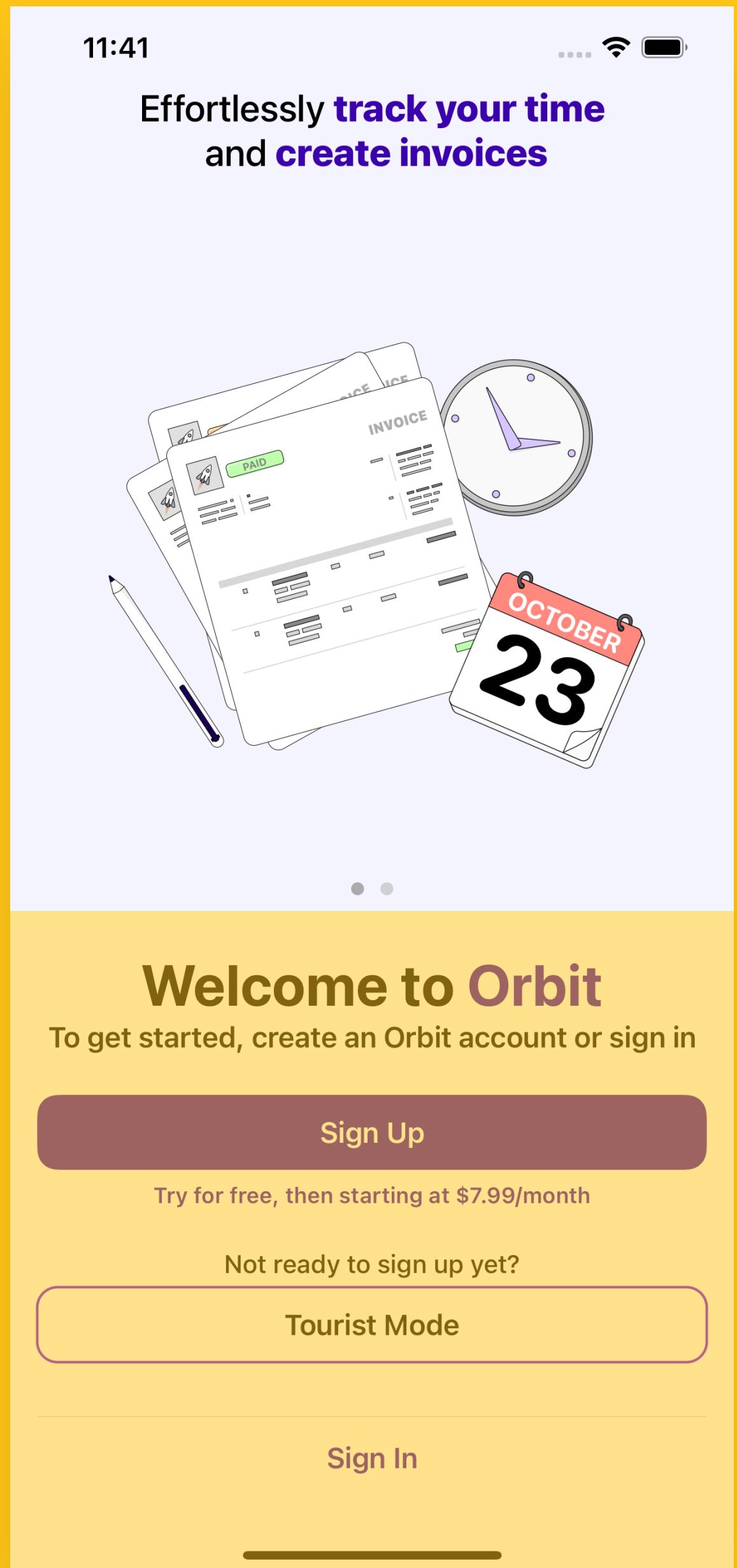


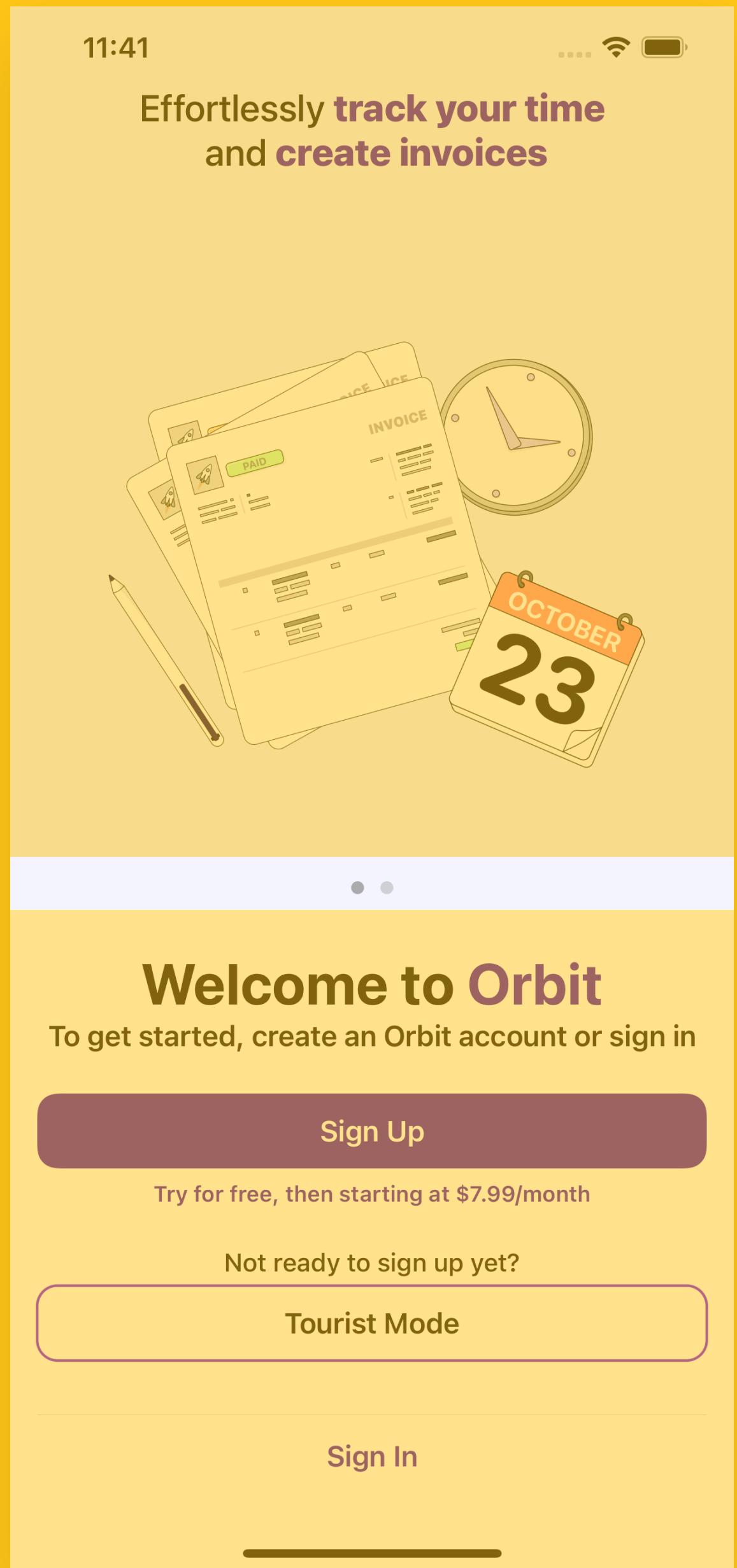


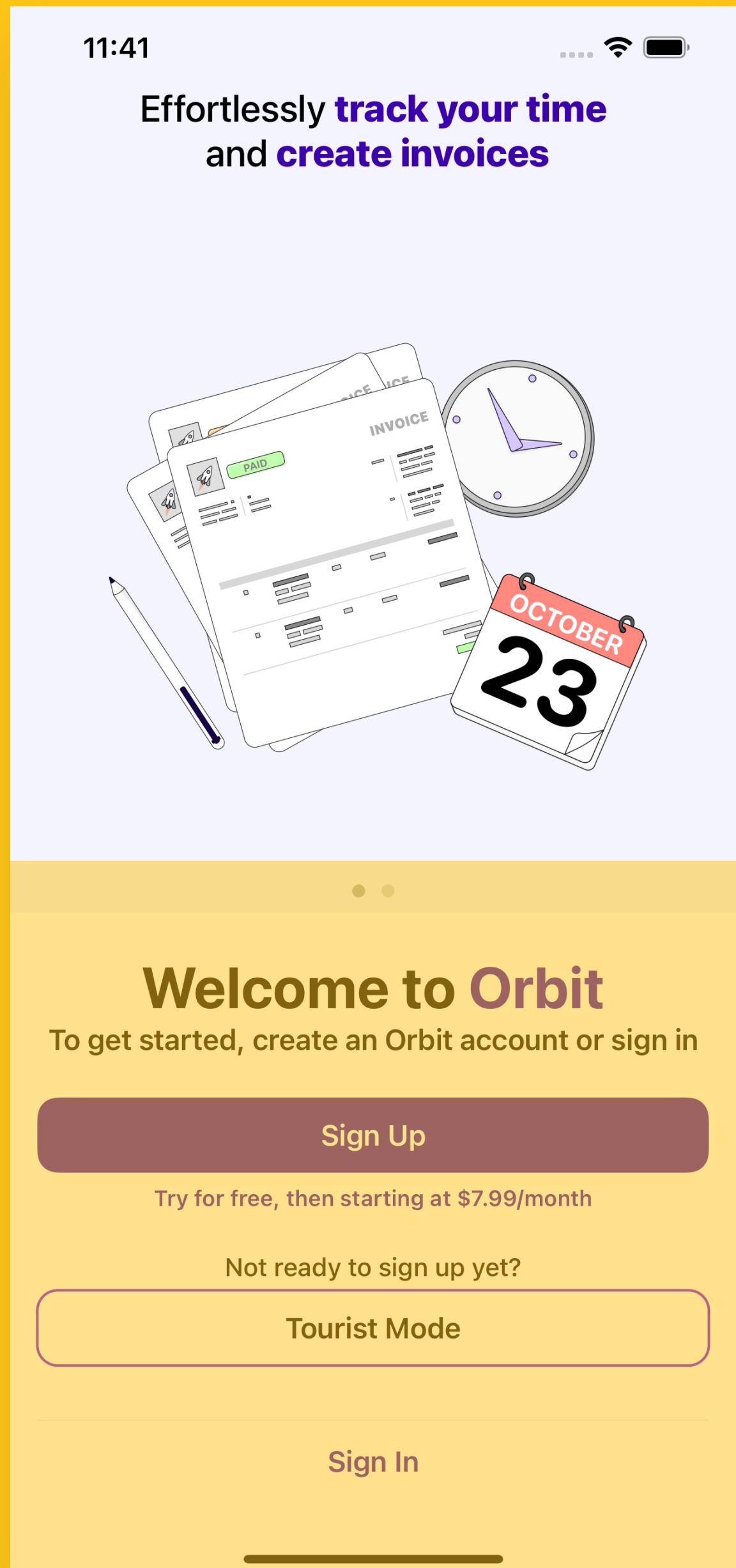


SwiftUI + Friends

UIViewRepresentable & UIViewControllerRepresentable



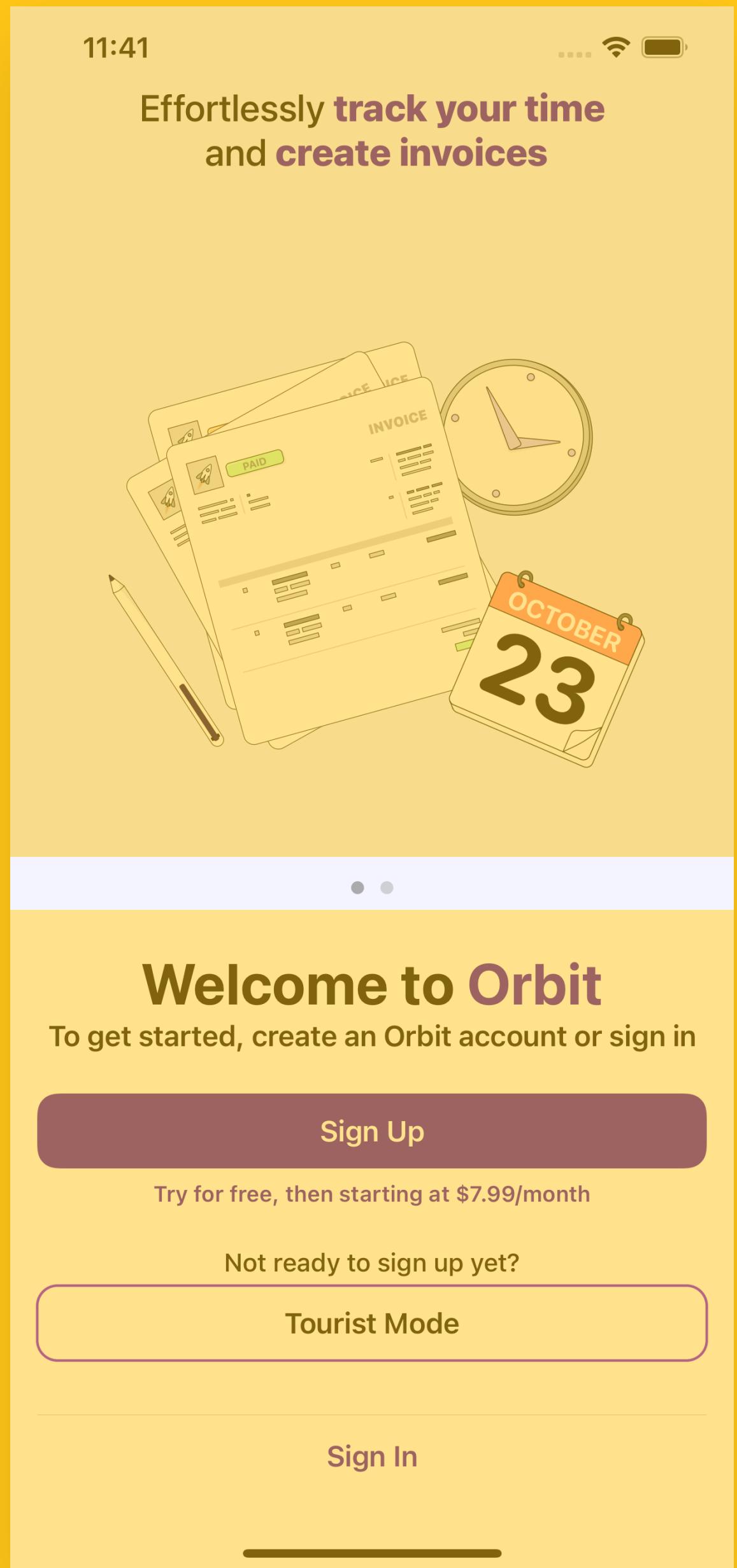






```
struct WrappedView: UIViewRepresentable {  
    func makeUIView(context: Context) -> UIPageControl  
    func updateUIView(_ uiView: UIPageControl, context: Context)  
}
```

Coordinator



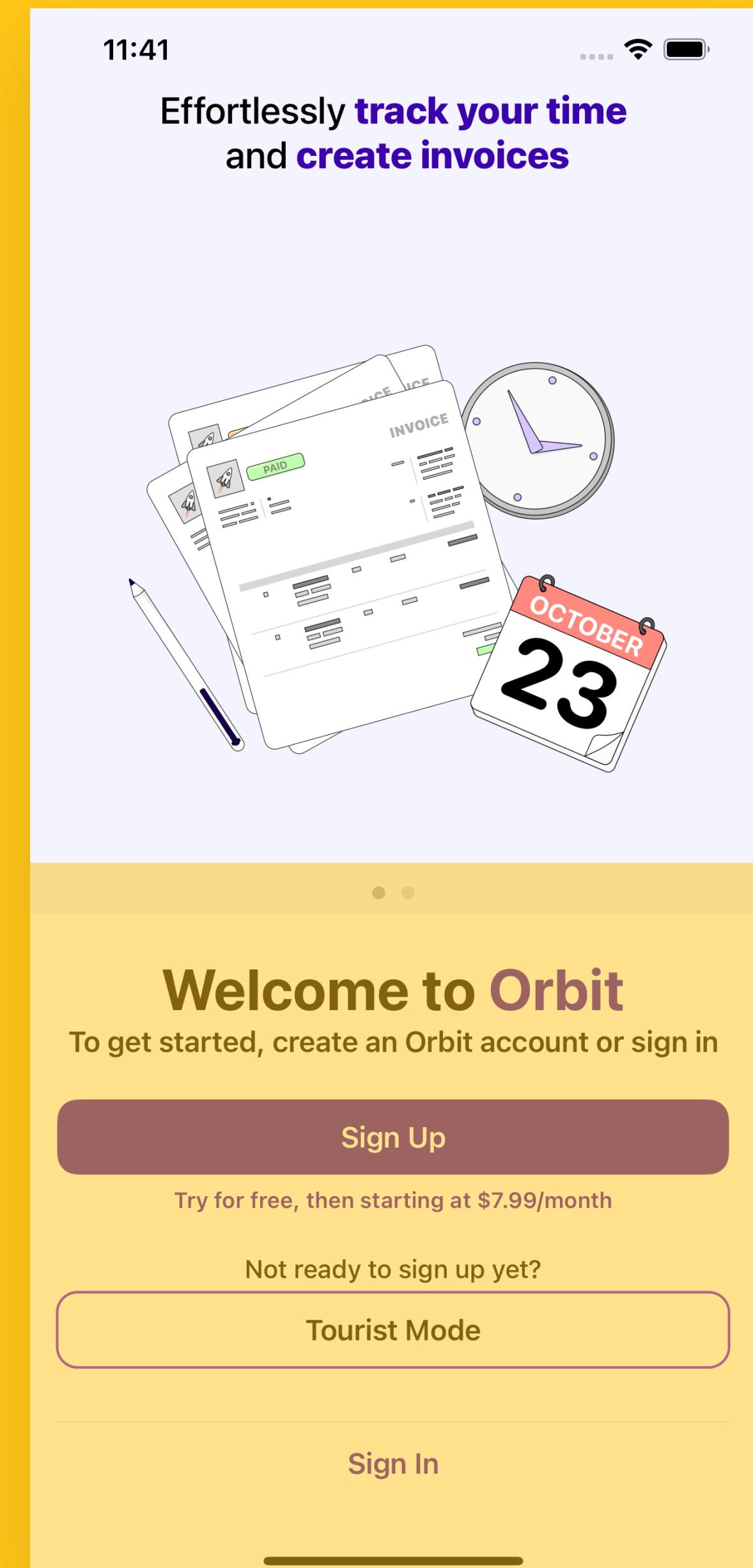
Coordinator



```
struct WrappedView: UIViewRepresentable {
    func makeCoordinator() -> Coordinator
}
```



```
struct WrappedView: UIViewRepresentable {  
    func makeCoordinator() -> Coordinator  
  
    class Coordinator: NSObject {  
    }  
}
```



UIApplicationDelegateAdaptor



```
@main
struct OrbitApp: App {
    #if os(iOS)
    @UIApplicationDelegateAdaptor(AppDelegate.self) var appDelegate
    #elseif os(macOS)
    @NSApplicationDelegateAdaptor(AppDelegate.self) var appDelegate
    #endif

    var body: some Scene {
        WindowGroup {
            ContentView()
        }
    }
}
```



```
class AppDelegate: NSObject, UIApplicationDelegate {}
```

UIHostingController

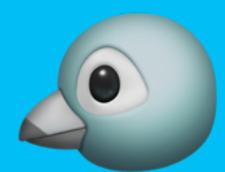


UIHostingController(rootView: view)

File > New > Project...

Thank You

Thank You



@malinsundberg



malinsundberg.com



timeinorbit.com

Music by
Andrew Applepie