

App Name and Description:

Bands is a budget tracking app that enables users to make more informed financial decisions by having easy access to their past transactional data. Through Bands, users may add their earnings and expenses and see how this affects their monthly budget, an attribute they set when first creating an account, by seeing real time changes in their balance.

Packages and Dependencies:

- Firebase (11.5) -> <https://github.com/firebase/firebase-ios-sdk>
- iOS 18.1
- Swift 5
- Xcode 16.1

Special Instructions:

Make sure to run the app on the iPhone 16 Pro simulator in portrait mode. There is no test account since users' transactional data is stored in CoreData and their user attributes (monthly budget and balance) are stored in User Defaults. New users can create an account using the "Create Account" button in the login screen. Note that because this app uses CoreData and User Defaults to store user data in memory, creating multiple accounts on one device will independently store data for each account, but rather display the same data no matter which account logged in on the device, so I highly recommend making only one account.

Bugs:

A bug I was unable to resolve before the project deadline lies in the navigation logic. The app is unable to recognize which screen is the root of the navigation stack depending on the user's authentication status. Thus, it treats the LoginView as the root view, causing there to be a back button in MainView that leads back to LoginView, and the back button in the other screens

also navigate back to MainView. To temporarily fix this issue, go to settings and change the monthly budget. For some reason, this removes the back button between MainView and LoginView and the navigation within the app works just fine.

I tried using a single state-driven approach to define the root view (LoginView or MainView) based on the user's authentication status. This state variable was to be defined in FinalProjectApp then other screens would inherit it using a binding variable. However, this approach caused many other bugs and fatal errors that affected Firebase configuration and the persistence controller that I couldn't debug.

Required Feature Checklist

- ☒ "Settings" screen. The two behaviors implemented are:
(fill in) Light/Dark Mode, Changing monthly budget
- ☒ Non-default fonts and colors used

Two major elements used:

- ☒ Login/register path with Firebase
- ☐ Core Data
- ☐ User Profile path using camera and photo library
- ☐ Multithreading
- ☒ SwiftUI

Minor Elements used:

- ☐ Two additional view types such as sliders, segmented controllers, etc. The two implemented are: (fill in) Date picker, Switches (light/dark toggle)

At least one of the following:

- ☒ Table View
- ☐ Collection View
- ☐ Tab VC
- ☐ Page VC

At least one of the following:

- ☒ Alerts
- ☐ Popovers
- ☐ Stack Views
- ☒ Scroll Views
- ☐ Haptics
- ☒ User Defaults

At least one of the following:

- ☐ Local notifications
- ☐ Core Graphics
- ☐ Gesture Recognition
- ☐ Animation
- ☒ Calendar
- ☐ Core Motion
- ☐ Core Location / MapKit
- ☐ Core Audio
- ☐ Firebase (if not used to fulfill Major Element requirement)
- ☒ Core Data (if not used to fulfill Major Element requirement)
- ☐ Other (such as QR code, Koloda, etc.) with approval from the instructor – list them