

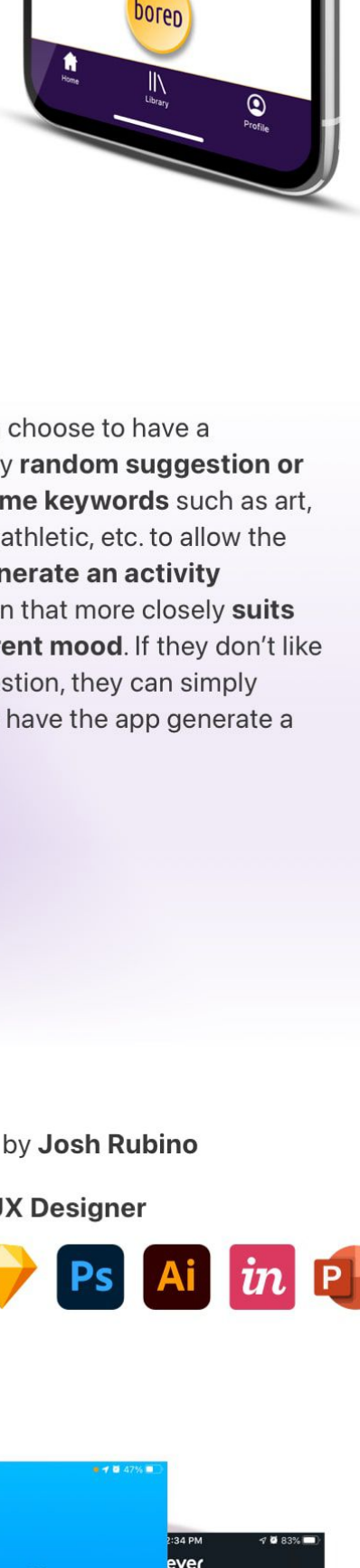
bored

An activity generating mobile app that requires zero decision-making



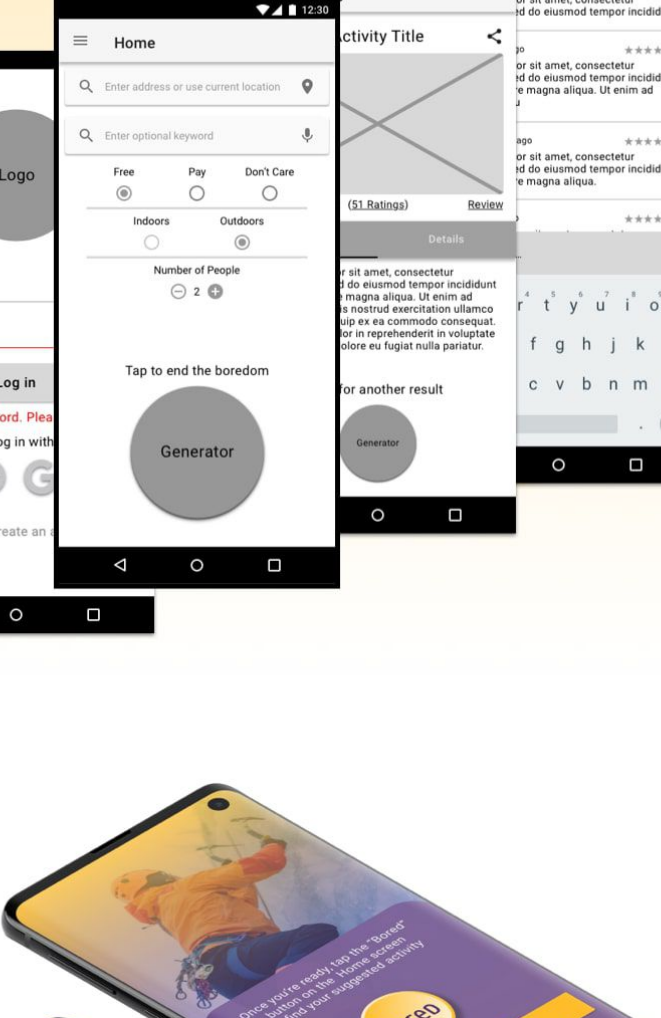
Bored was built individually for both **iOS** and **Android** mobile devices adhering to **iOS Human Interface** and **Material Design** guidelines respectively.

The goal of this app is to provide an **activity or experience** to a user they would not have thought to do otherwise. It is a tool that **gets people out into the world** (using their location or inputting a location) **to experience life**.

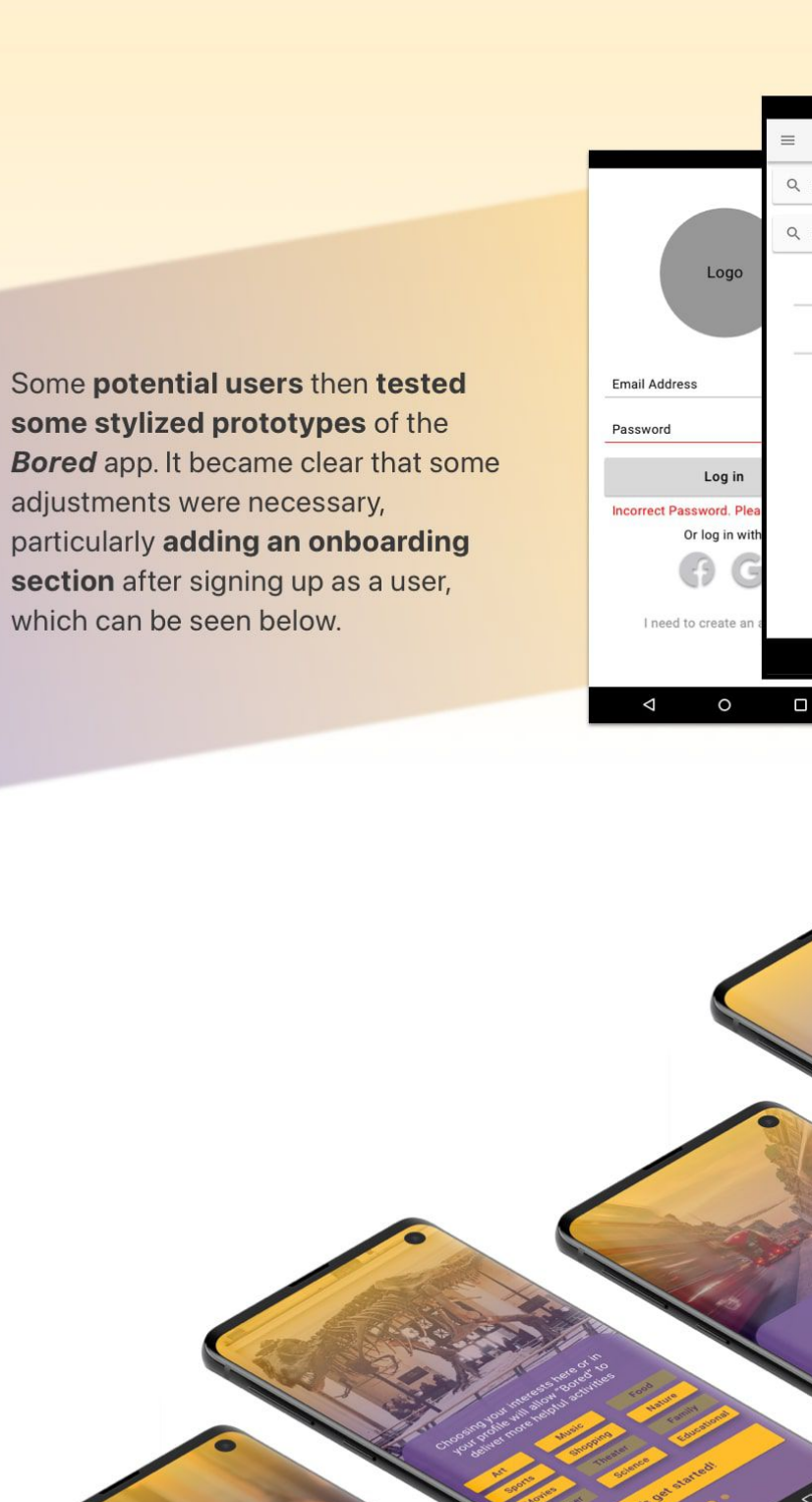


Users can choose to have a completely **random suggestion** or **select some keywords** such as art, outdoors, athletic, etc. to allow the app to **generate an activity suggestion** that more closely **suits their current mood**. If they don't like the suggestion, they can simply choose to have the app generate a new one.

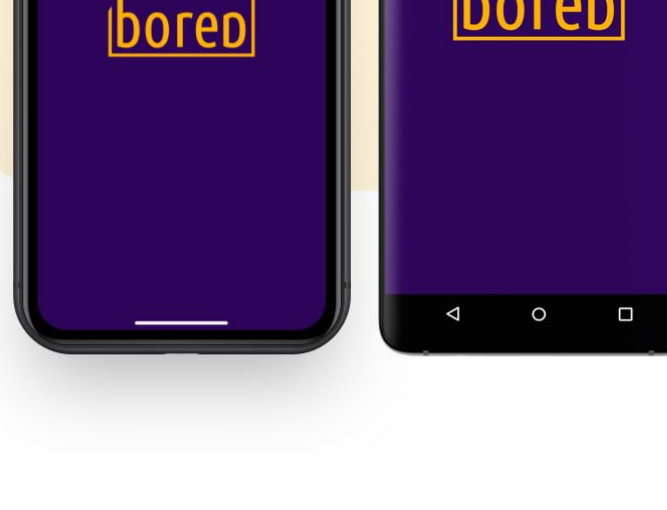
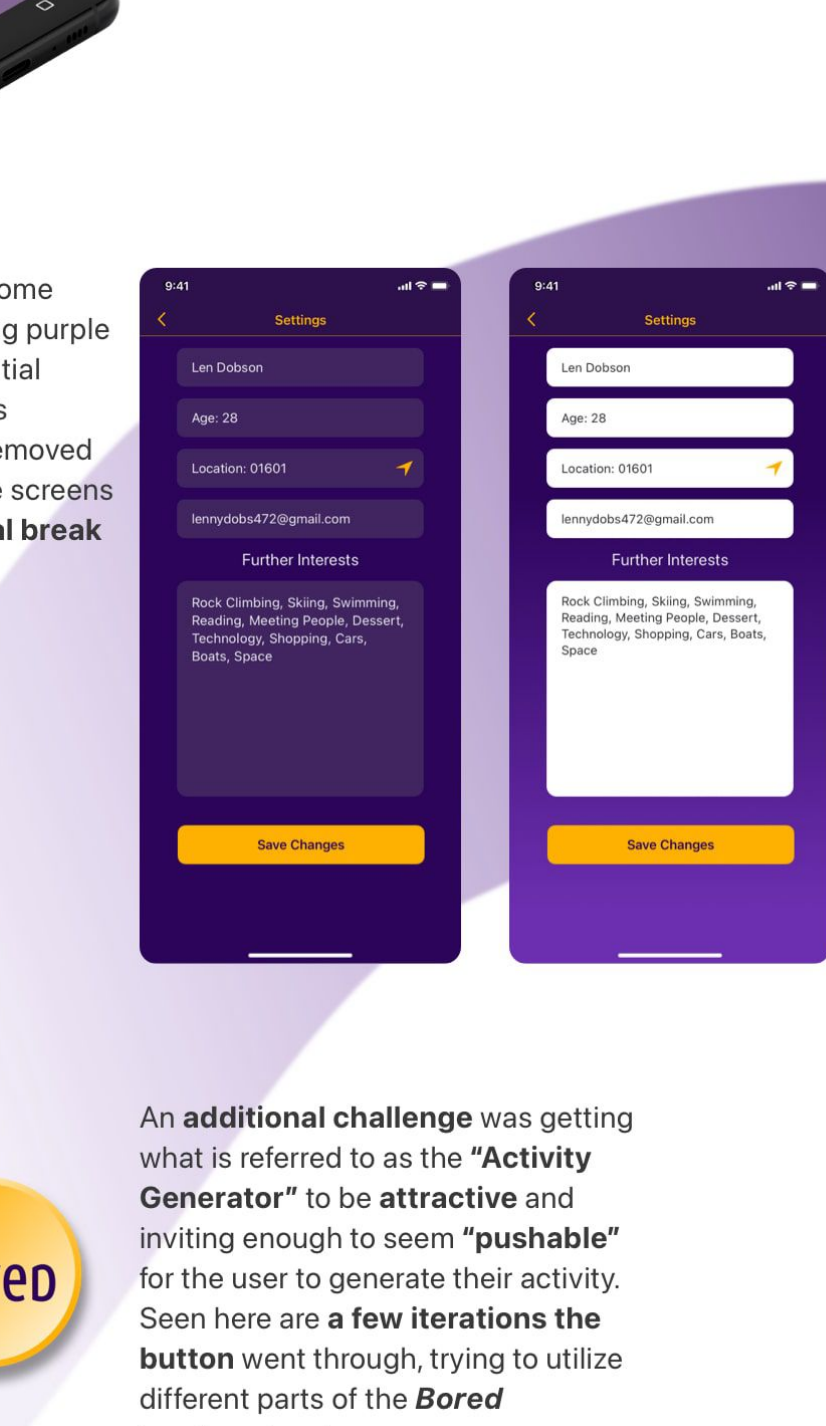
This research helped find an **opening in the market** for an app that gives just **one suggestion** and takes the **stress of making a decision out of the equation**.



Here are just a few wireframe examples from the app. To the left is **iOS** and below you can see the **Android** version.



Some **potential users** then tested some **stylized prototypes** of the **Bored** app. It became clear that some adjustments were necessary, particularly **adding an onboarding section** after signing up as a user, which can be seen below.



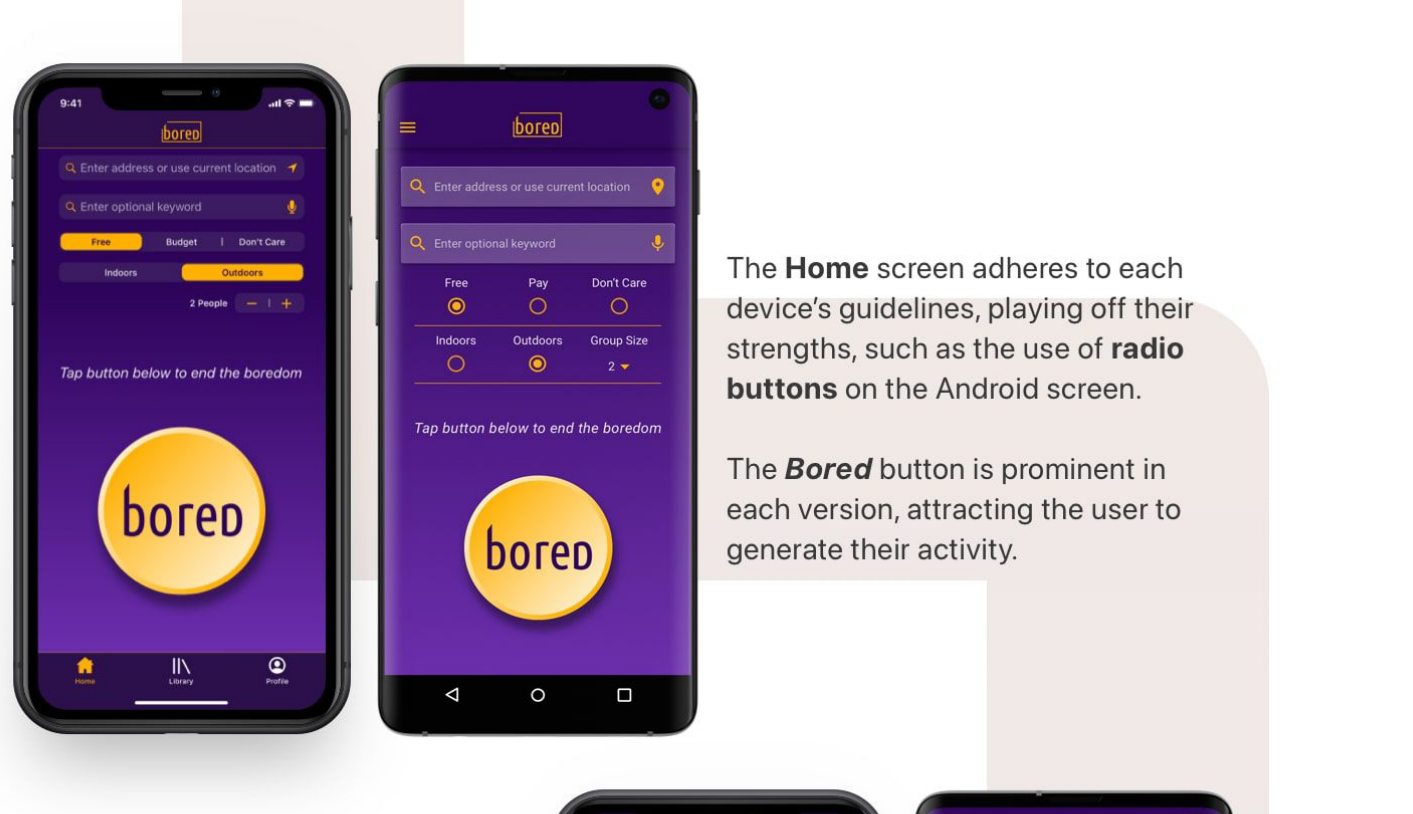
An **additional challenge** was getting what is referred to as the **"Activity Generator"** to be **attractive** and inviting enough to seem **"pushable"** for the user to generate their activity. Seen here are a **few iterations** the button went through, trying to utilize different parts of the **Bored** logo/wordmark.

Bored was developed for both **iOS** and **Android** mobile devices, adhering to the guidelines of each as well as using the staple font of each: **SF Pro** for iOS and **Roboto** for Android. These mockups show the subtle differences of each screen.

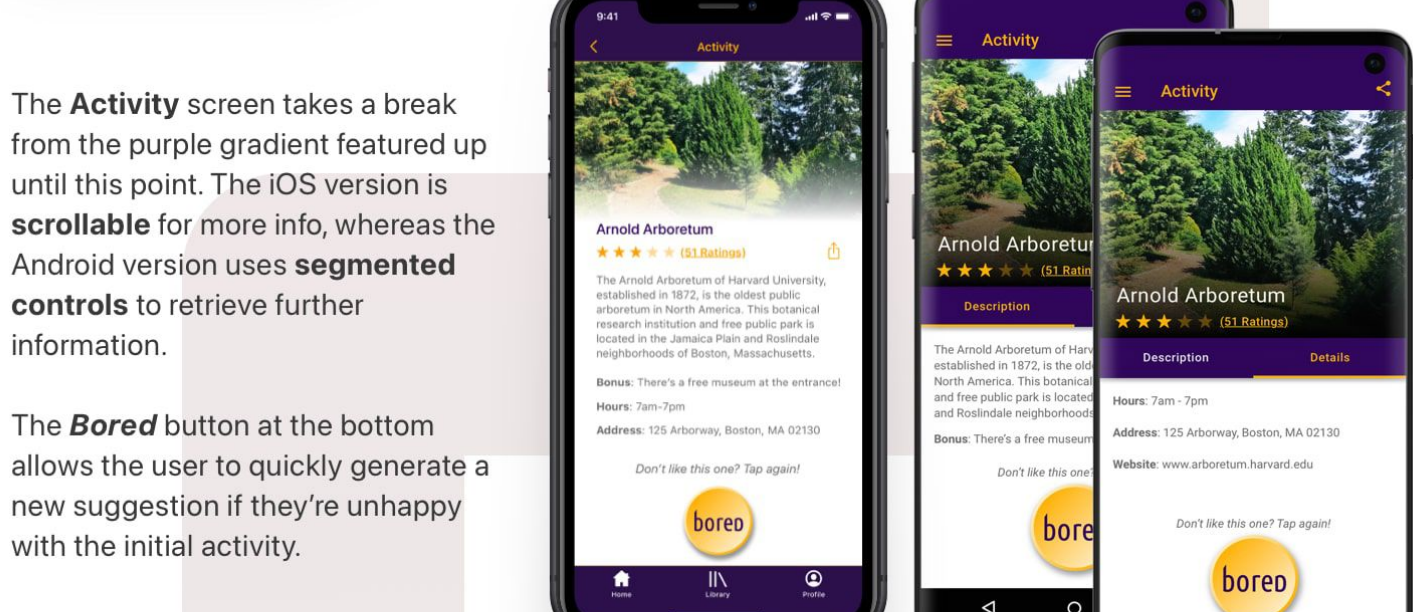


The **Splash** screens are ostensibly identical. iOS on the left and Android on the right as will be seen with the following screens.

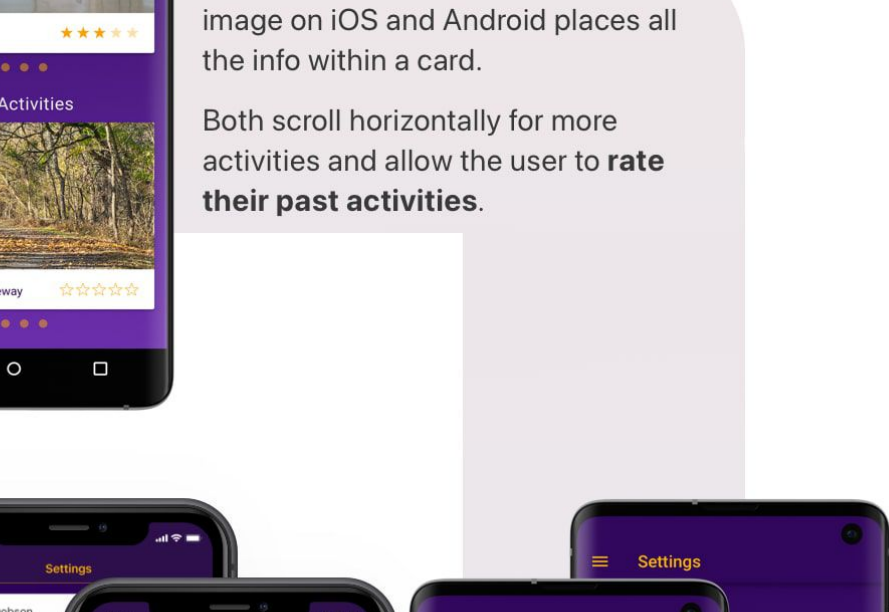
The **Landing** screens differ in that the Android version has a deeper purple gradient overlaying the image.



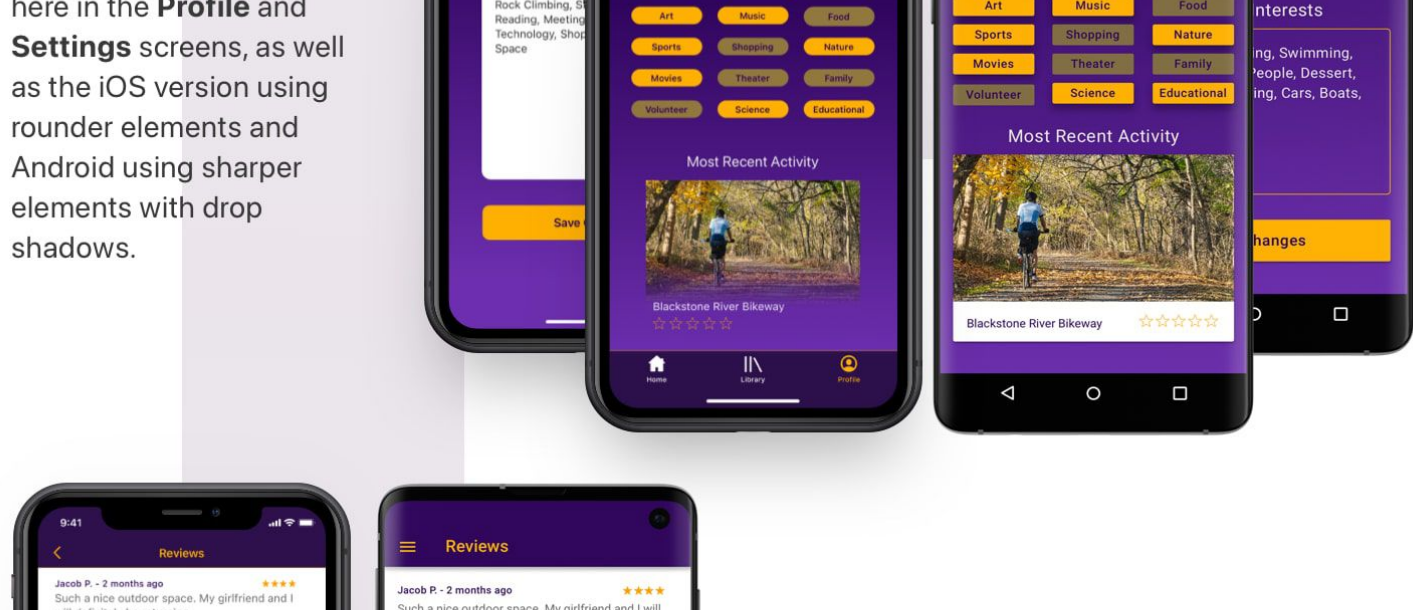
The **Sign up** and **Log in** screens, as with the other screen with entry forms, differ in that the iOS version (left) features black text over white, with Android's (below) white text over a colored background.



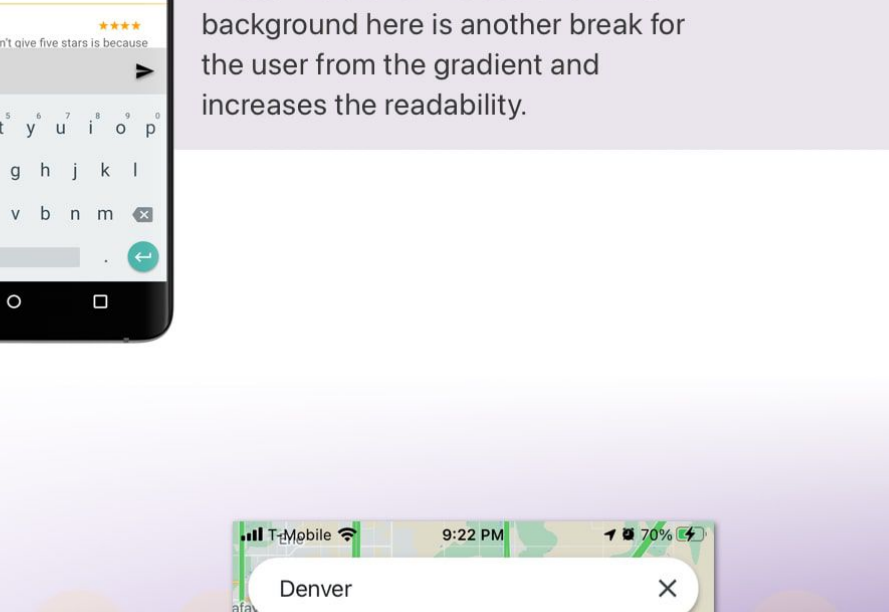
The two versions use **different alerts** to **notify the user** that they've entered incorrect login info. Both work well for their corresponding design.



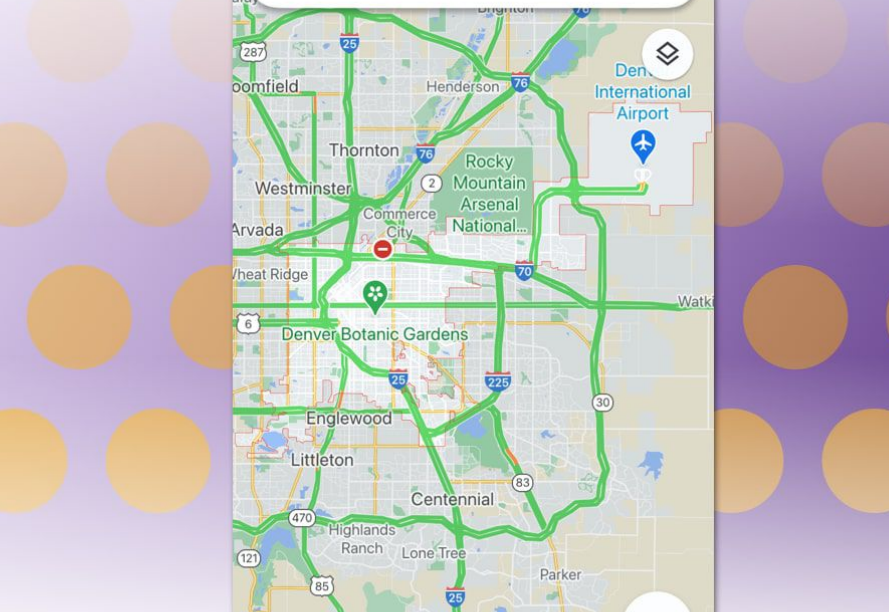
The **Onboarding** process is below (iOS above Android). These screens introduce the users to the app in regards to **setting location**, **getting the best activity suggestions** and **curating those suggestions**.



The **Activity** screen takes a break from the purple gradient featured up until this point. The iOS version is **scrollable** for more info, whereas the Android version uses **segmented controls** to retrieve further information.



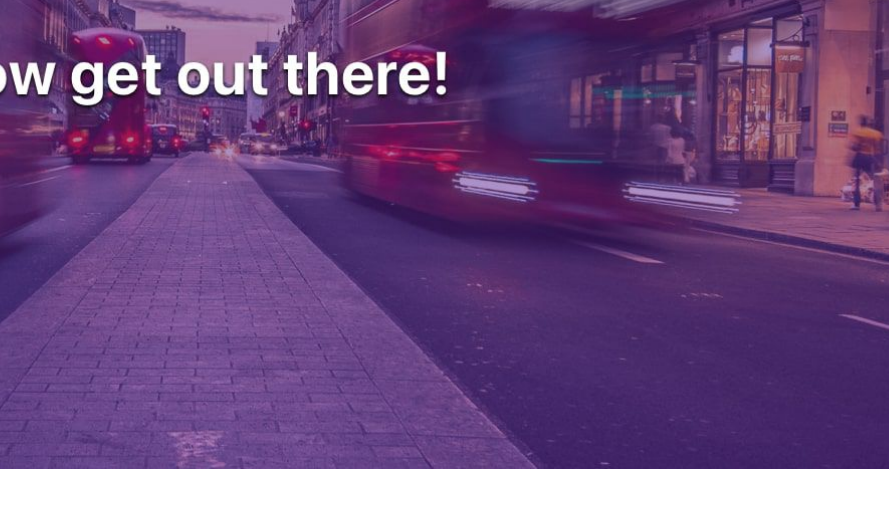
The **Bored** button at the bottom allows the user to quickly generate a new suggestion if they're unhappy with the initial activity.



The **My Library** screen takes a break from the purple gradient featured up until this point. The iOS version is **scrollable** for more info, whereas the Android version uses **segmented controls** to retrieve further information.



Differences pointed out in earlier screens are clear here in the **Profile** and **Settings** screens, as well as the iOS version using **rounder elements** and Android using **sharper elements** with **drop shadows**.



The **Reviews** screen is nearly identical in both versions. The use of a white background here is another break for the user from the gradient and increases the readability.

Future iterations could see things such as **map functionality** and the ability to **make purchases** or **reservations** from the app.

