orelo

By : Ali Gholami

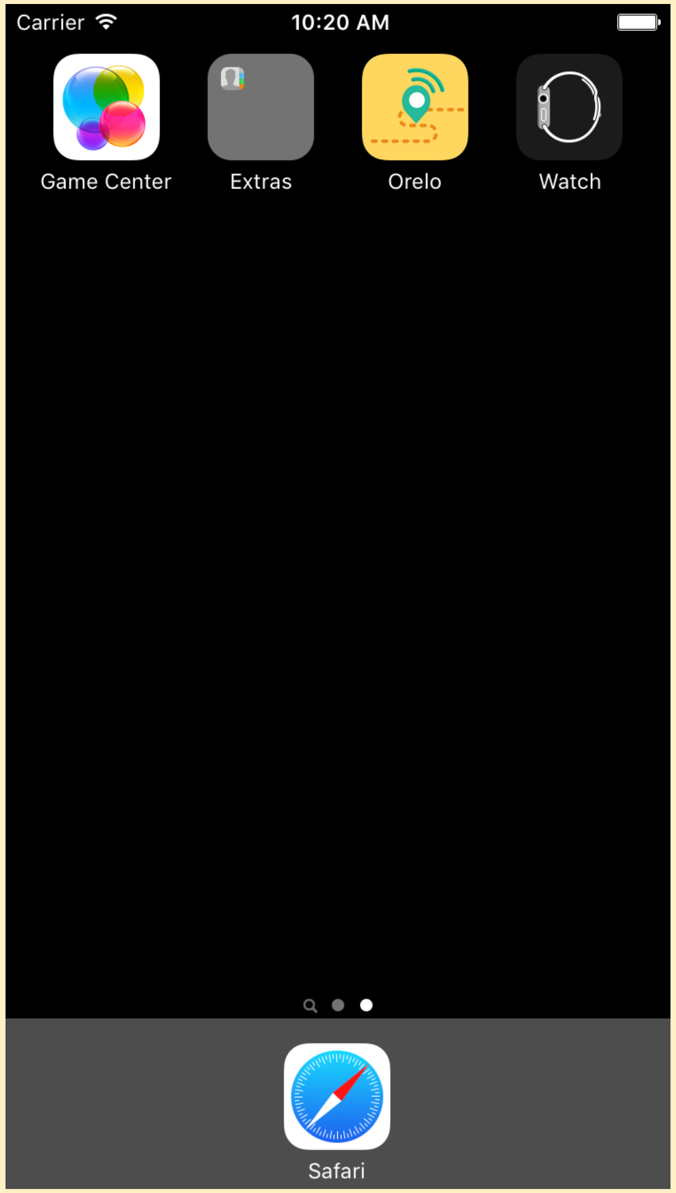
What is Orelo?

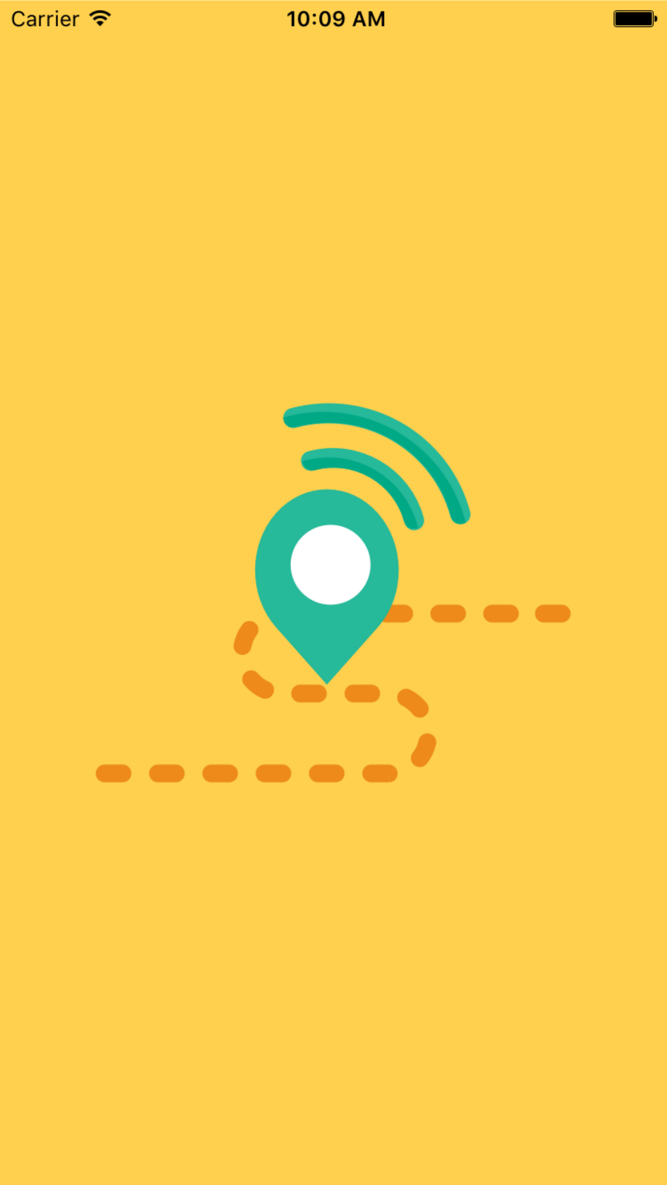
- **Orelo** means **ear** in Esperanto 👂
- A **“baby” social network** for recording and sharing ambient sounds
- It’s an **iPhone** application, developed with Swift on Xcode

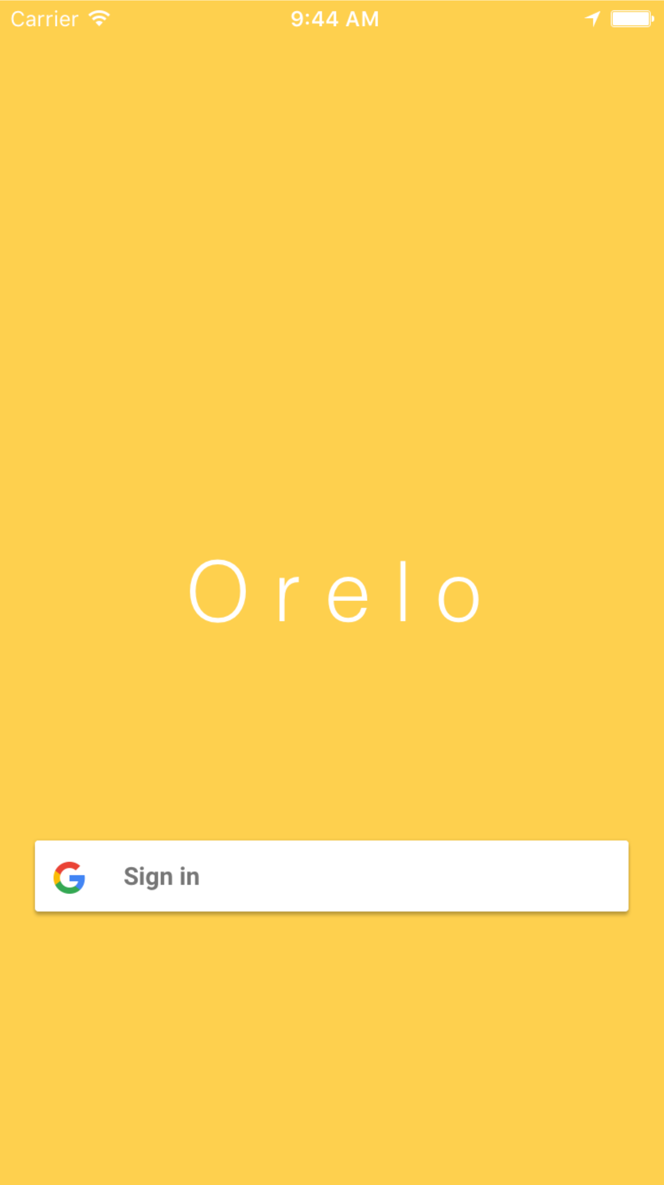
Launching Orelo

- Orelo launches to **login** view controller
- **Authentication** is handled by Firebase/auth using Google Sign In button
- User will be redirect to Google login page and come back to Orelo after allowing it to use account’s info
- Main view controller of the application is a **Tab Bar View Controller** with three tabs

Launching Orelo





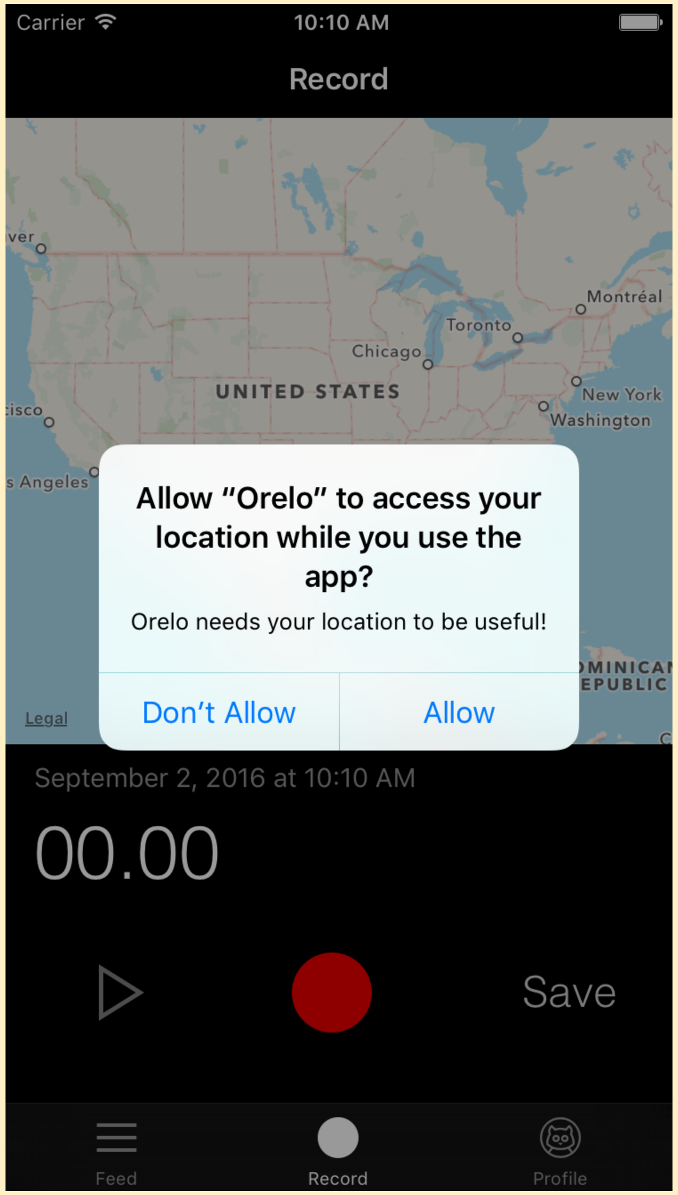


*all screenshots have been taken on iPhone 6 simulator

Recording New Voice

- Orelo needs user’s **permission** to collect location information by CoreLocation
- Recording is **bounded** by 60 Seconds
- When Recording is done, you may **Play** it or **Save** it
- Save button will pop up an **UIAlertView** with text field to choose a title for your voice
- Voice info will be recorded using **CoreData** schema

Record New Voice



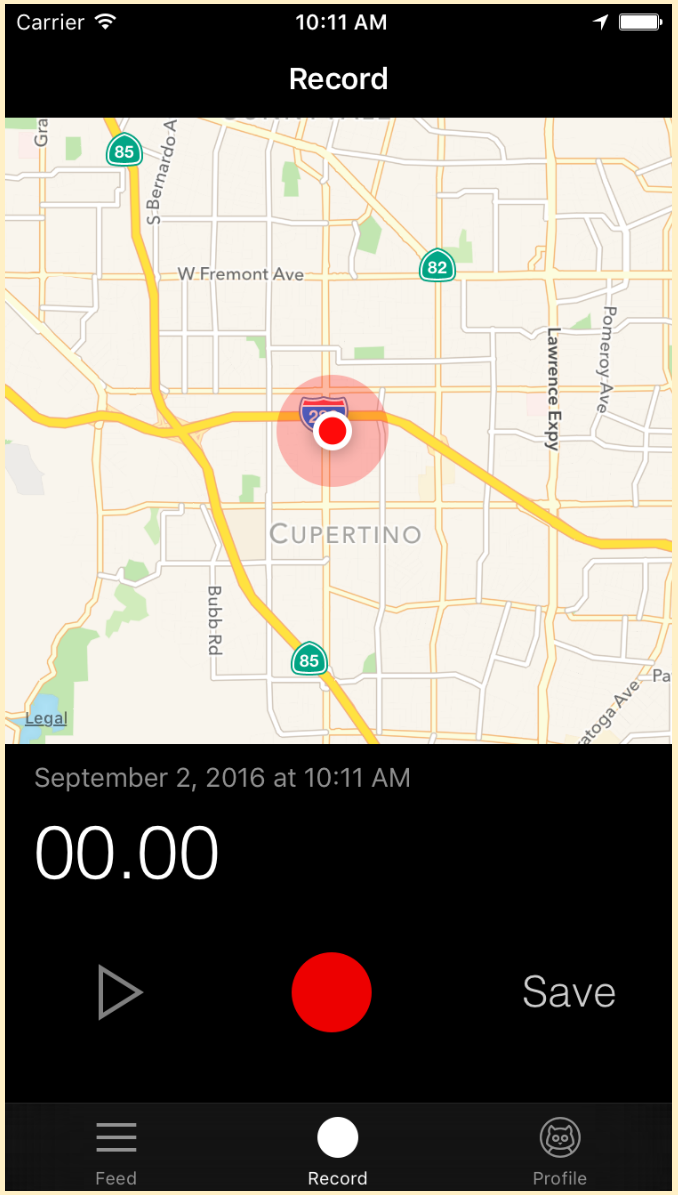
Allow "Orelo" to access your location while you use the app?

Orelo needs your location to be useful!

Don't Allow | Allow

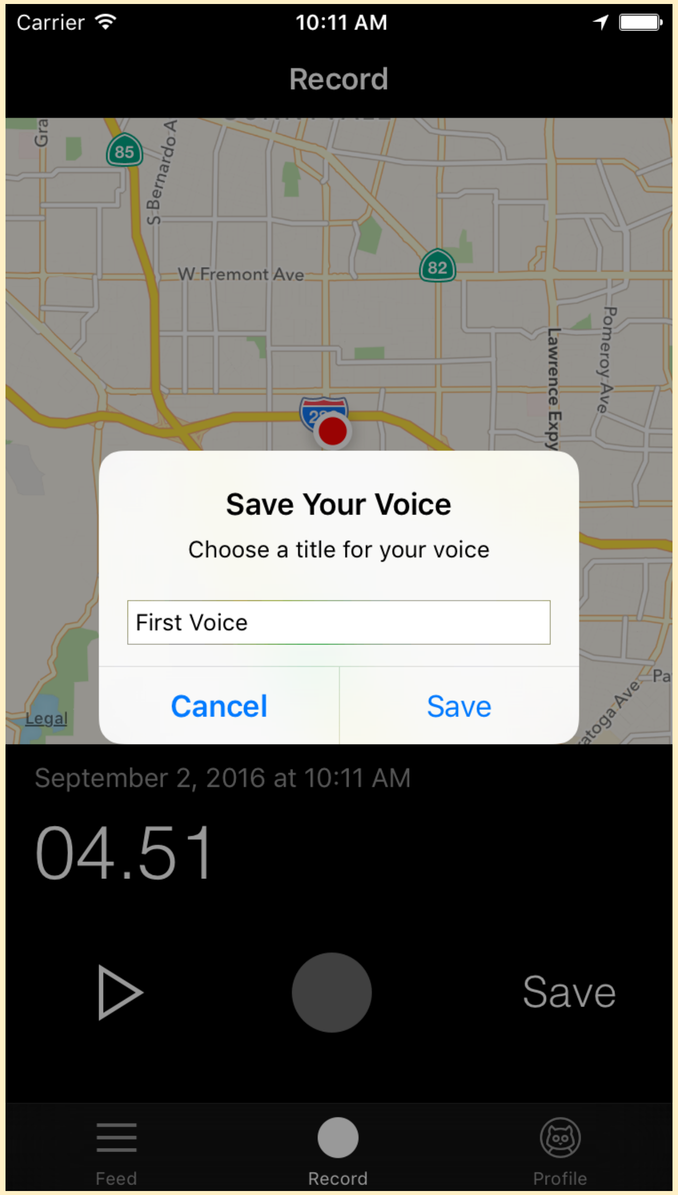
September 2, 2016 at 10:10 AM

00.00



September 2, 2016 at 10:11 AM

00.00



Save Your Voice

Choose a title for your voice

First Voice

Cancel | Save

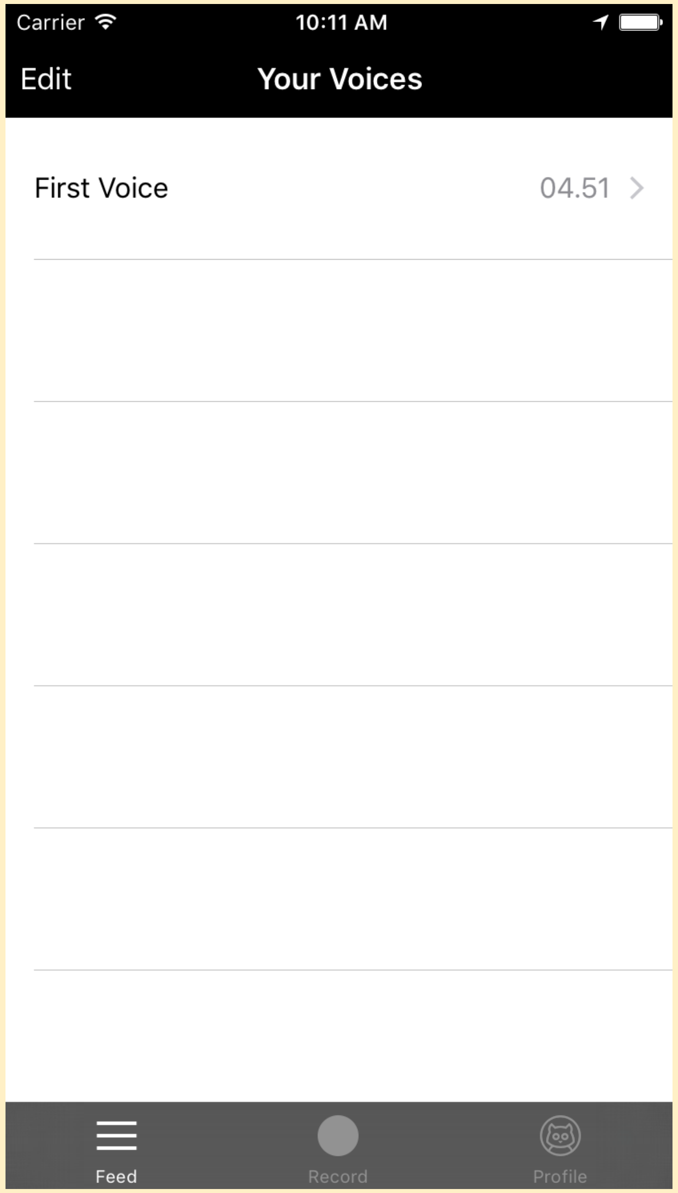
September 2, 2016 at 10:11 AM

04.51

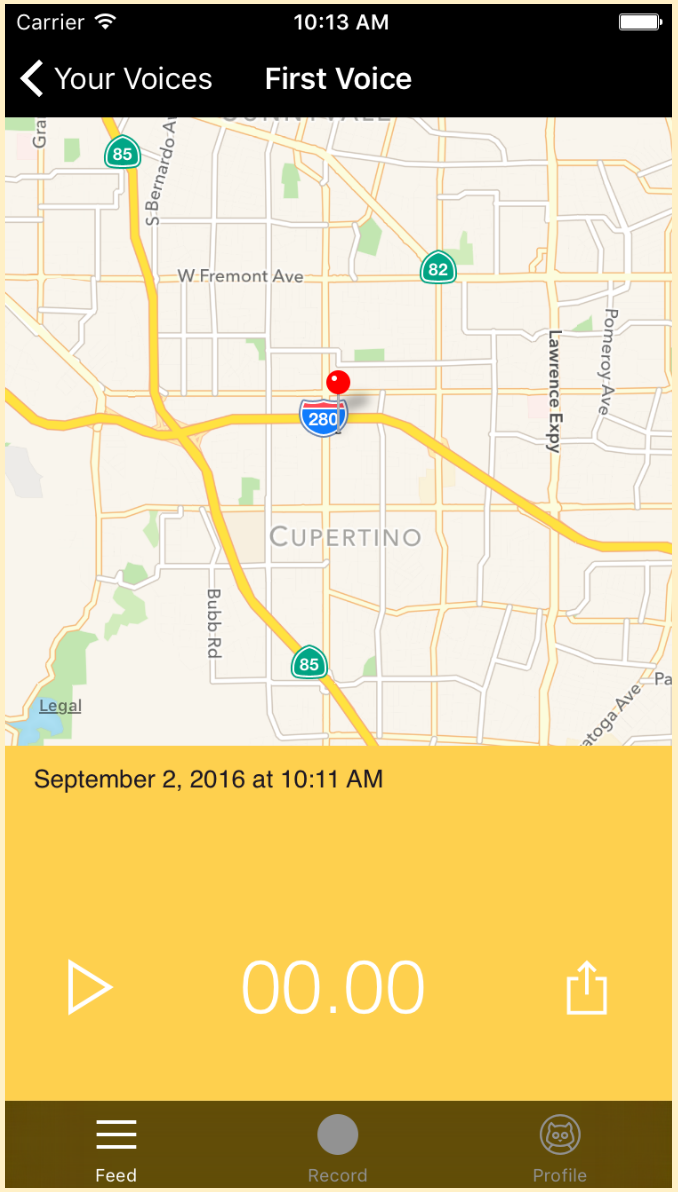
Listen To Your Voices

- Save will redirect user to first tab which is a table loaded with all recorded voices
- Loading data is handled by NSFetchedResultsController
- You may edit table and voices by tapping on **Edit**
- Tapping each voice will **perform a segue to a detail view** where user can listen to the voice, check location and of course, share it

Listen To Your Voices

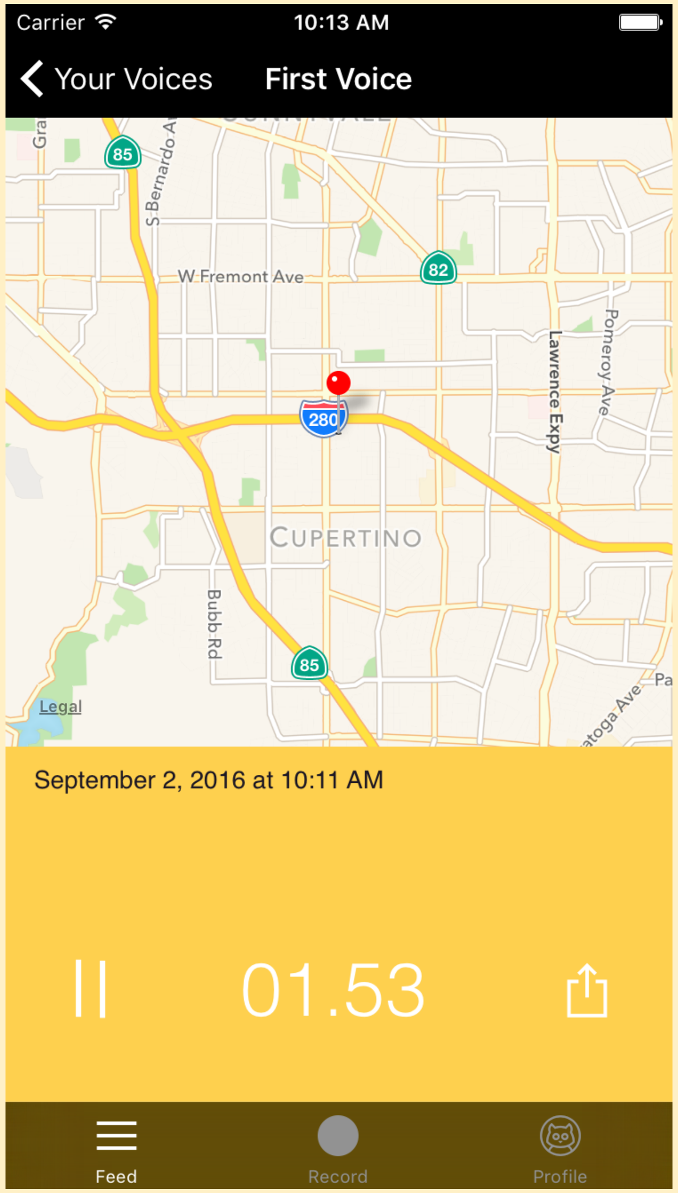


First Voice 04.51



September 2, 2016 at 10:11 AM

00.00



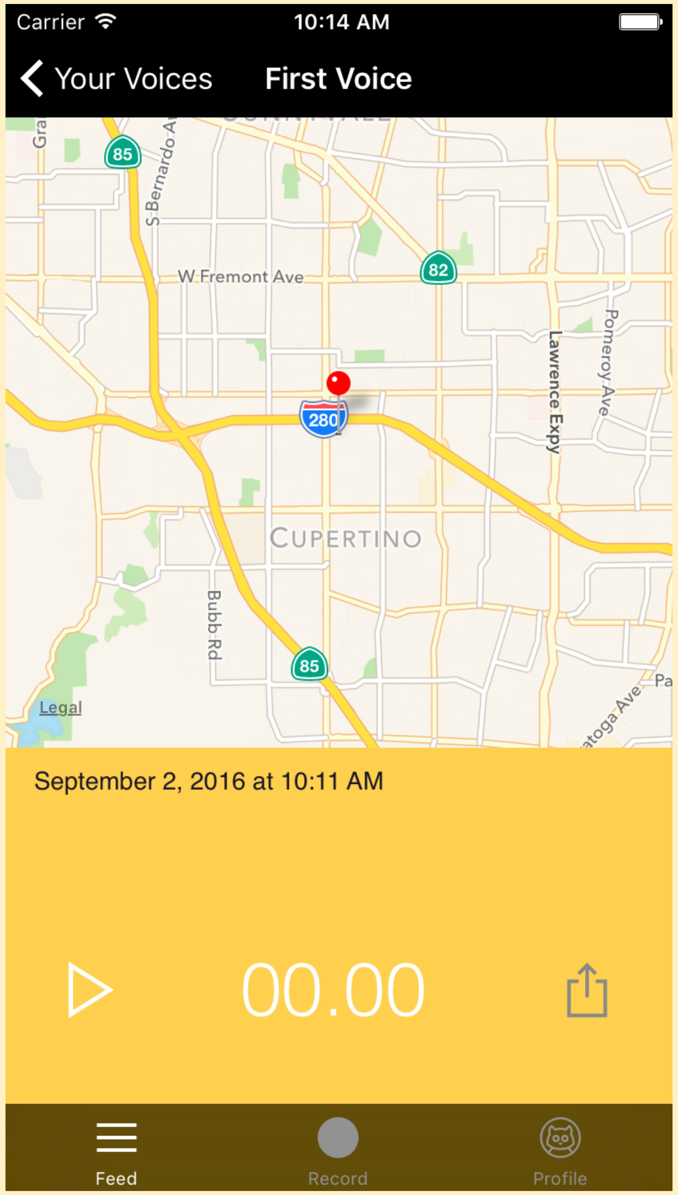
September 2, 2016 at 10:11 AM

01.53

Share Your Voice

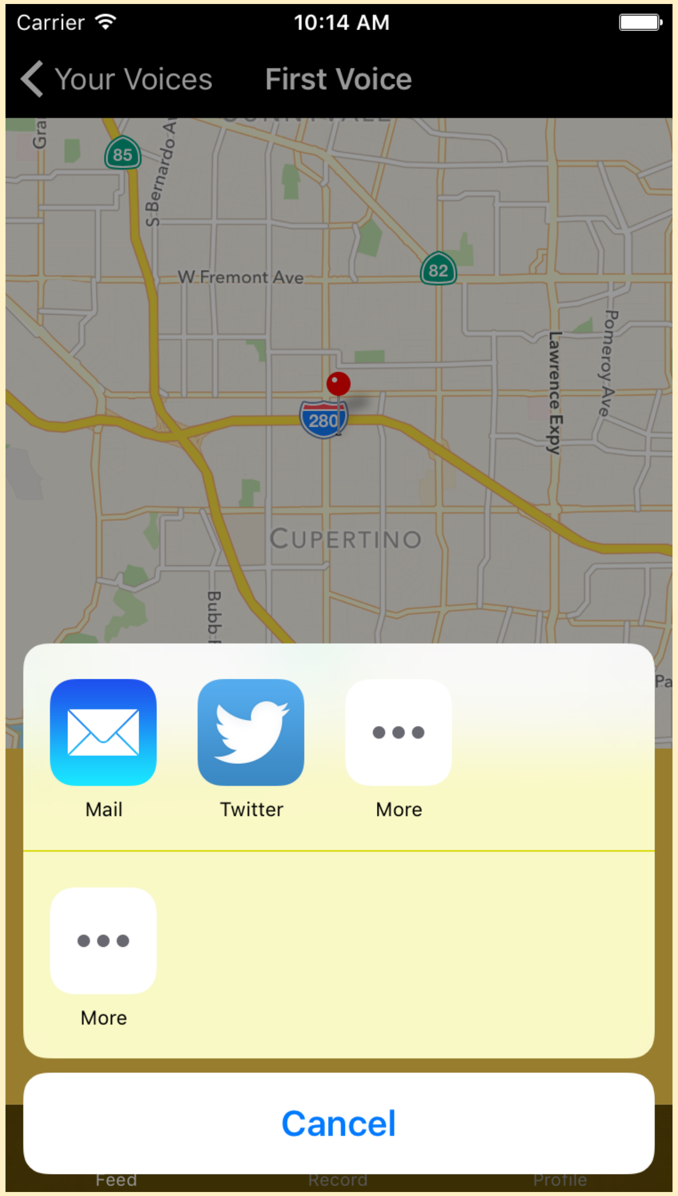
- By clicking **share button**, a default UIActivityService will come up and let you to share your voice
- Your voice title will appear in your message automatically. And a human readable location name of voice location will be there as well
- This address is provided by a **CLGeocoder's** method called **reverseGeocodeLocation()**

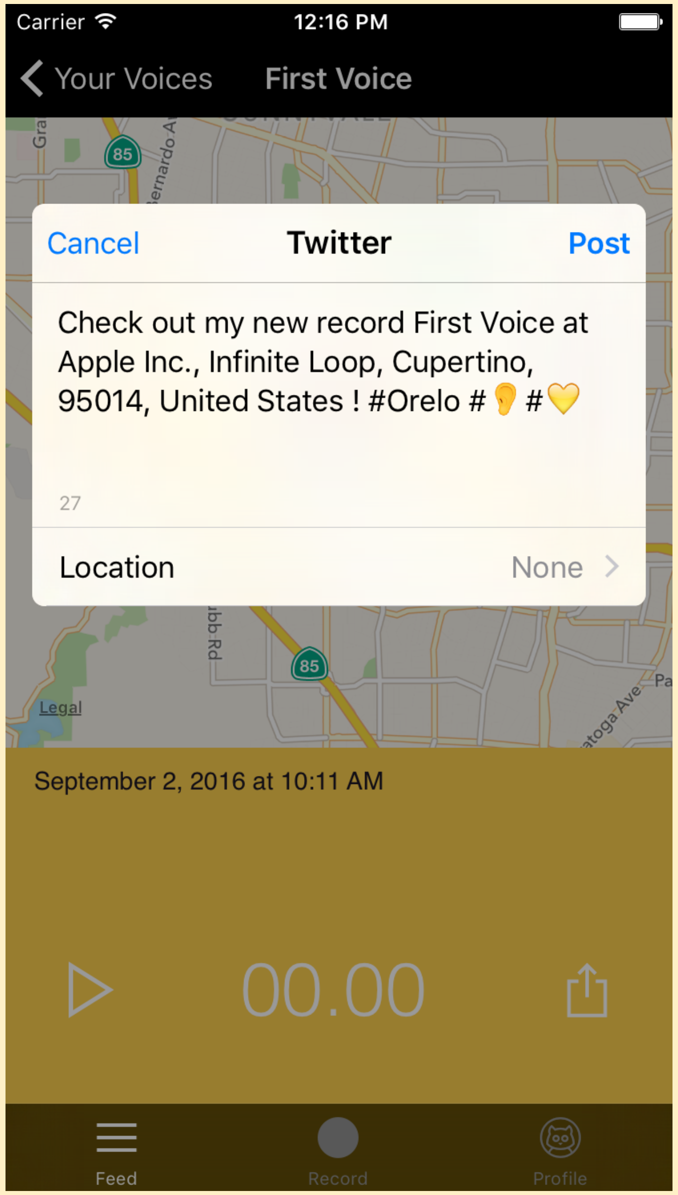
Share Your Voice



September 2, 2016 at 10:11 AM

00.00





Check out my new record First Voice at Apple Inc., Infinite Loop, Cupertino, 95014, United States ! #Orelo #👂 #💛

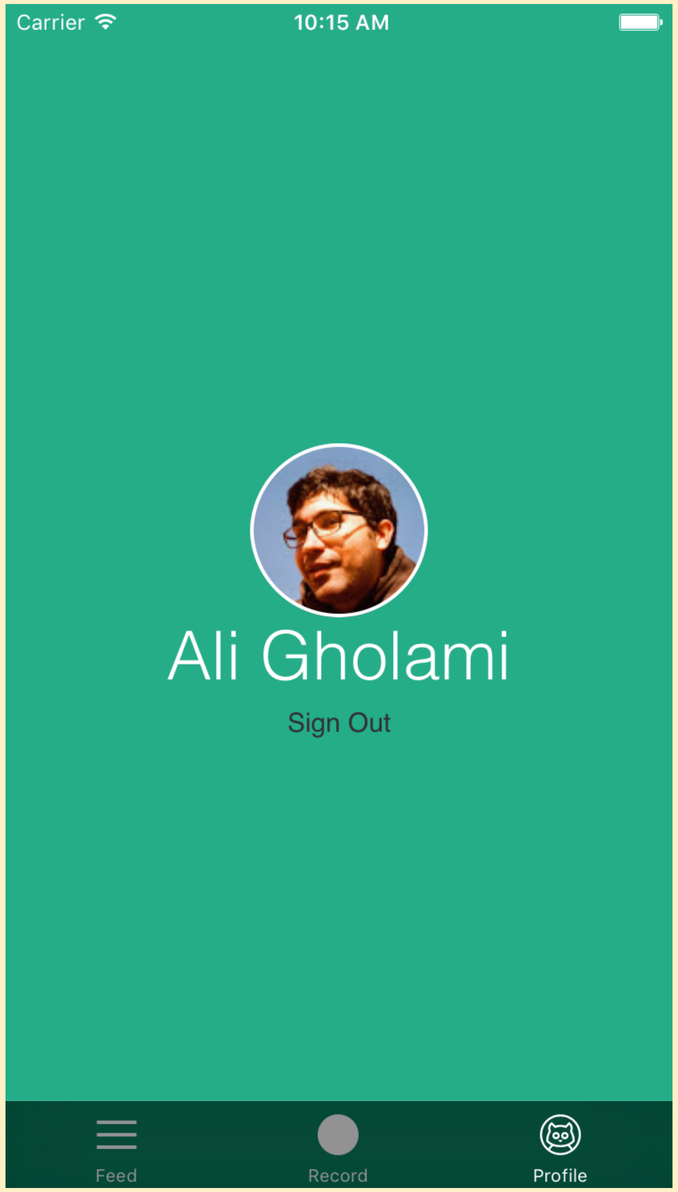
27

Location: None

September 2, 2016 at 10:11 AM

00.00

Profile Tab



- User's profile image and display name are **synced** from the google account
- A **CALayer()** overlay gives the image its rounded shape
- Sign Out button will logout the user and perform a segue to the login view controller

Which tools have been used?

UITableView AVFoundation AVAudioRecorder

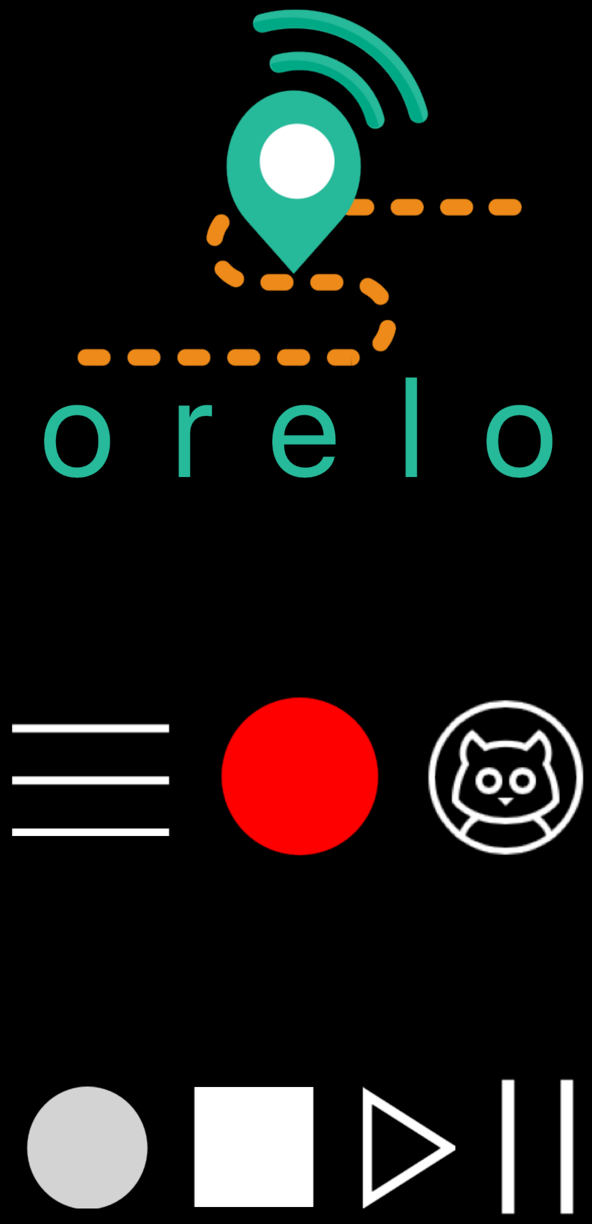
Xcode 7.3 NSProcessInfo CoreData

Firebase UIKit CoreLocation Swift 2.2 CALayer

AVAudioPlayer NSTimer UIActivityView MapKit

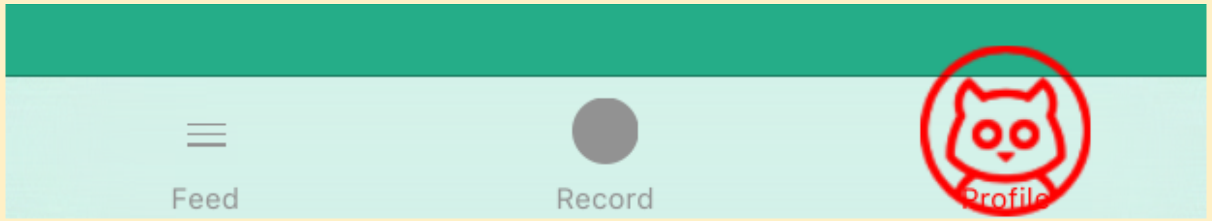
Sketch CLGeoCoder NSManagedObjects

Buttons, Tabs, icons and launching logo have been designed using **Sketch 3** in SVG format and saved in three different sizes PNG files

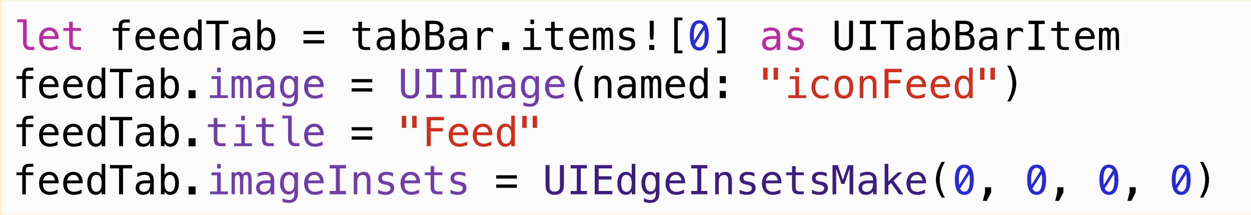


Challenges I

- At one point, after setting images for tab bars, clicking (tapping) on tabs caused weird behaviour:



- It has been solved by connecting control of these tabs to the code instead of setting them from Attributes inspector



```
let feedTab = tabBar.items![0] as UITabBarItem
feedTab.image = UIImage(named: "iconFeed")
feedTab.title = "Feed"
feedTab.imageInsets = UIEdgeInsetsMake(0, 0, 0, 0)
```

Challenges II

- Firebase did not redirect user after logging in to Google account.

- It turns out that **GIDSignInDelegate** should have been implemented in AppDelegate so Firebase could recognise where it should redirect after login

- Then **GIDSignInUIDelegate** could be implemented in different View

Challenges III

- One major issue was to keep table view immediately updated after saving a new voice.

- I was trying to figure it out myself, by fetching data from CoreData database and using **reloadData()** method on the table view

- But final and best solution was using **NSFetchedResultsController** which keeps track of all changes in database and immediately update table view

Possible Improvements

- A real social network needs **a database connected to the network**, so voices could be accessible by everyone
- **Design matters!** I’m quite sure that it can have better UI to provide better user experience
- Better **error handling** is needed
- More advanced auto layout to make **Orelo** also available on **iPads**
- I imagine that using Orelo on **appleWatch** could be even more fun

Thank You!