

No description, website, or topics provided.

9 commits

4 branches

0 releases

1 contributor

Branch: videoPosts

View #7

Create new file


Upload files

Find file







Clone or download

This branch is 2 commits ahead, 2 commits behind master.

#19 Compare

 SpencerCurtis Update README.md

Latest commit ff97e3b on Oct 17, 2018

 LambdaTimeline.xcodeproj	Add starter project	4 months ago
 LambdaTimeline	Add starter project	4 months ago
 .gitignore	Create .gitignore	4 months ago
 Podfile	Add starter project	4 months ago
 Podfile.lock	Add starter project	4 months ago
 README.md	Update README.md	4 months ago

README.md

Lambda Timeline

Introduction

The goal of this project is to take an existing project called LambdaTimeline and add features to it throughout this sprint.

Today you will be adding video posts.

Instructions

Create a new branch in the repository called `videoPosts` and work off of it from where you left off yesterday.

You're welcome to fulfill these instructions however you want. There are a few suggestions to help you along if you need them.

- Create UI that allows the user to create a video post. The UI should allow the user to record a video. Once the video has been recorded, it should play back the video (the playback can be looped if you want), allow the user to add a title just like in an image post, then post it.

▼ Recording UI Suggestions

- You may take the `CameraViewController` used in the guided project as a base. You will need to modify it so the video doesn't get stored to the user's photo library, but instead you can use the url that the `didFinishRecordingTo outPutFileURL: URL` method gives you back to send the video data to Firebase
- For information on how to play back the video, refer to `AVPlayer` and `AVPlayerLayer` in the documentation. Of

course you're welcome to google for more information, but familiarize yourself with these objects first.

2. Add a new case to the `MediaType` enum in the `Post.swift` file for videos. Take a look at the memberwise initializer for the `Post`. Make sure that it takes in a `MediaType` and sets `mediaType` correctly.
3. Create a new collection view cell in the `PostsCollectionViewController`. It should display the video, as well as the post's title and author. It's up to you if you want the video to play automatically or have it play when you tap the cell, or a button, etc.
4. Create a detail view controller for video posts similar to the `ImagePostDetailViewController`. It should display the post's video, title, artist, and its comments. It should also allow the user to add their own text and audio comments.

Go Further

- Add the ability to record audio with the video. When the video plays on a cell or in the video post view controller, the audio should play as well.
- Add the ability to record from the front camera and let the user flip the cameras.
- Add the option to save the video to the user's photo library