

Branch: audioComments ▾

[ios-lambda-timeline](#) / [README.md](#)

Find file

Copy path



SpencerCurtis Update README.md

f15b3fe on Nov 5, 2018

1 contributor

64 lines (38 sloc) 3.29 KB

# 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 audio comments.

## Instructions

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

You're welcome to fulfill these instructions however you want. If you'd like suggestions on how to implement something, open the disclosure triangle and there are some suggestions for most of the instructions.

1. Create UI that allows the user to create an audio comment. The UI should allow the user to record, stop, cancel, and send the recording.
  - ▼ Recording UI Suggestions
    - In the `ImagePostDetailViewController`, change the `createComment` action to allow the user select whether they want to make a text comment or an audio comment, then create a new view controller with the required UI. The view controller could be presented modally or as a popover.
    - ~~Alternatively, you could modify the `ImagePostDetailViewController` to hold the audio recording UI.~~
2. Create a new table view cell that displays at least the author of the audio comment, and a button to play the comment.
3. Change the `Comment` to be either a text comment or an audio comment.
  - ▼ Comment Suggestions
    - In the `Comment` object, change the `text`'s type to be an optional string, and create a new `audioURL: URL?` variable as well. Modify the `dictionaryRepresentation` and the `init?(dictionary: ...)` to accomodate the `audioURL` and the now optional `text` string.
4. In the `PostController`, add the ability to create a comment with the audio data that the user records, and save it to Firebase Storage, add the comment to its post, then save the post to the Firebase Database.

▼ Post Controller Suggestions

- Create a separate function to create a comment with the audio data.
- You can very easily change the `store` method to instead take in data and a `StorageReference` to accomodate for storing both Post media data and now the audio data as well.

5. In the `ImagePostDetailViewController`, make sure that the audio is being fetched for the audio comments. You are welcome to fetch the audio for each audio comment however you want.

#### ▼ Audio Fetching Suggestions

- You can implement the audio fetching similar to the way images are fetched on the `PostsCollectionViewController` by using operations, an operation queue, and a new cache. Make a new subclass of `ConcurrentOperation` that fetches audio using the comment's `audioURL` and a `URLSessionDataTask`.

6. Implement the ability to play a comment's audio from the new audio comment cell from step 2. As you implement the `AVAudioRecorder`, remember to add a microphone usage description in the `Info.plist`.

## Go Further

---

- Add a label (if you don't have one already) to your recording UI that will show the recording time as the user is recording.
- Change the audio comment cell to display the duration of the audio, as well as show the current time the audio is at when playing.