Introduction
Hi everyone, we are NotJustCode, this is Azeezat, and I am with my colleagues MJ and Shafiq today. We are here to speak on the progress of our recording app.

The agenda today is to speak about the direction we've taken in implementing tags into our prototype, we'll speak on the platform we are using, flutter, and finally our MVPs for the project.

Tags
I will begin by discussing the tags implementation.

Since our last meeting, we have re-invented our search functionality. **Page1:** The issue that was previously pointed out to us was that, for example, if a user has 2 folders which a recording can fit into, for example chor and music, how do we help the user categorize that so they don't run into issues if the recording can go into multiple folders.

Page5: Our initial design was to name the recording and allow the recording to be saved into a folder or category. Since then, we have thought through how we want to implement tags, YAAAY! So now we have tags in our categorization. The dialogue box will allow the user to save the recording, put it into a folder, and have as many tags as is necessary for the recording.

This helps to eliminate the confusion as they can really choose any folder, as the search is powerful enough to pick up on the tags they have added to the recording. **Page4**: The example here is, there are 2 scenarios for a user to search. The first is the homepage, once the user clicks on the search button, it will conduct a global search which will search the entire app for that recording. The second scenario is to go inside a folder, for example, choir. If we hit the search button, we expect the results to be limited to that specific folder before getting results from other folders. The search functionality is layers deep.

Page5: How we have implemented the tags to do this is that the search functionality searches for both name and tags. The name is what the user saves *here* and the tag is what they save *here*.

Page6: We are implementing both but the name takes precedence as this most likely what the user wants, and then the tags come after, that is the sequence we want to code for.

Scroll down page 6: For example, we have names of recordings here, *read the recordings*, and then we have the names of their corresponding tags here, *read the tags*.

Page7: Here, we demo the user typing "B", in the choir folder, which is interesting because there is a recording name that starts with "B" as well as multiple tag names which begins with "B". Books is the name of an actual recording, so that populates our search first, however, it would also go through the tags to looks for any tags that start with "B". In this case, we have "bible", "books". The user can swipe left or right to see more tags. But for now we are focussing on the recordings themselves. The user will first see names of recordings that begin with "B", followed by recordings that have TAGS which begin with “B”. The sequence will be alphabetical, of course. If the search can’t pull up any recordings with that, at this line of demarcation here, it will look into other folder to provide suggestions within the application.

The next thing we want to try with this is to give users the ability to search for a specific tag by typing a hashtag before the name of the tag. And if they type just a string, we will gear towards bringing out results based on the name of the recording.

With that said, I will now pass on to my colleague MJ to speak on the platform we have chosen to move forward with, Flutter.

What platform did we choose? - Flutter

Why did we choose Flutter, what research have we done that showed that Flutter is the best way forward for us?
Before making a decision, the most important criteria for us was a platform that has good audio capabilities. We looked at examples of apps which have been created using different platforms, such as Kotlin, Xamarin, Flutter, ReactNative and more. We chose Flutter because we believe that it has the best options for audio and also, we feel like it is supported, just because it is owned by Google and we might be able to get more help when we have questions or any issues.

So far, we have installed Flutter and have started playing with it. We followed along another app which was created using Flutter. Here is what we have found about the app so far while we were using it:

- Everything in Flutter is a widget - that is, you basically build your UI out of widgets. “Widgets describe what their view should look like given their current configuration and state. When a widget’s state changes, the widget rebuilds its description, which the framework diffs against the previous description in order to determine the minimal changes needed in the underlying render tree to transition from one state to the next.”
- Stateful (for user interaction) and stateless (unchanging) widgets
- Audio packages:
    1. Flutter: https://pub.dev/packages/audio_recorder
    2. Flutter: https://pub.dev/packages/flutter_sound
    3. Medium - Flutter sound plug ins: https://medium.com/codechai/flutter-sound-plugin-audio-recorder-player-e5a455a8beaf

Then MJ can go through a bit of the features we have found and some features.

Flutter + Revision Control

MVPs
Moving forward, we know that we can't do all the things we would like to achieve with our app, so the first milestone we are hoping to complete is: Getting the app to actually record and its organizational features. That is the most important thing we need to do before we even begin implementing any other feature.

We have discussed and these are the extended milestones for us:

- Get the app to record
- Naming and saving of the recordings
- Folders and inserting the recordings into folders

MVPs

- Record
  - Play with the platform some more
  - Figure out what the sound packages do, and the features of the package
- Folders and tags
  - After we have figured out the features of the package, we will know what we need to know more about. We can now implement the categorization
- Getting the geolocation/metadata of the recording
- Figuring out how to get permissions to the phone so we can do more