

High Res Audio Recorder

By: Benjamin Hoang, Angelica Olmedo, Jeremy Magana, Jarrett Koran

Abstract:

Our project is in the form of a high bitrate voice recorder. This was chosen because the default voice recorder on the Android platform leaves much to be desired. We set out to make a better version that can be better than sixteen bits per sample and better than 48kbps bitrate. These numbers are even lower than the VoIP of an application such as Discord, making the default Recorder by Google, less than usable let alone production quality. It was our goal to bring a better sound quality and a good user experience to a new voice recorder application for the Android platform.

The Importance:

High bitrate audio is important because it allows for more information to be captured and reproduced in the recording, resulting in a higher quality and more accurate representation of the original soundwave. This is particularly important for music, streaming, podcasting, and conversation recording where the nuances and details of the sound are crucial to the listening experience. A higher bitrate also allows for a broader dynamic range and a greater level of detail in the reproduction of the audio, which can greatly enhance the overall listening experience. It makes it more future-proof and allows for easier editing of the audio waveform. Our application facilitates this by letting users decide their own bitrate that they would like to use. The bitrate in Google's application was measured to be 48Kbps, which is insufficient.

High frequency audio recording is important because it allows for the capture and reproduction of a wider range of frequencies in the audio signal. High frequency audio is important for accurately reproducing sounds at the upper end of the human hearing range, which can be important for certain types of audio content, such as sound effects in movies or video games, or high quality recorded conversations. According to the Nyquist-Shannon sampling theorem, in order to accurately reproduce a continuous-time signal in a discrete-time system, the sampling frequency must be at least twice the highest frequency present in the original signal. This means that, in order to accurately reproduce a sound with a maximum frequency of 20 kHz, for example, the sampling frequency must be at least 40 kHz. This is because, if the sampling frequency is less than twice the highest frequency present in the signal, the reproduced signal will contain aliasing artifacts that will distort the original sound. Our application facilitates this by letting users decide their own frequency that they would like to use, up to what is supported by their device. The frequency in Google's application was measured to be 32Khz, which is insufficient.

Introduction:

Our objective is to provide a better way for users to record high quality audio anywhere they are at any time. Beyond that we want users to have the freedom to customize their experience with complete control over the options and configurations of each and every recording. Modern mobile phones have amazing capabilities and it is up to the developer to allow users to be able to bring those capabilities out. It is our goal to create a simple yet elegant improvement on the voice recorder medium, that is capable of showing our knowledge of Android concepts while also being an application we as a team feel comfortable publishing as a usable product for the Google Play Store.

App Description:

Our High resolution audio recorder application features are the ability to record your voice at a high bitrate. A voice recorder is a simple concept and application so its features are qualitative not quantitative. We have enabled the user the ability to sign in and create an account with the application. As well as making a bottom navigation bar that will take you from the recording menu, to the saved recordings, to the settings tab. These features are all that encompass the application.

Features:

- Login/Signup
- Voice Recorder
- List of recordings

Discussion (challenges, innovation, merit):

The biggest challenge was finding extra Kotlin examples outside of the ones provided in class by the professor. Some topics were things that we had gone over in class and had examples of in the homework, however not every challenge we faced was directly related to something that we did in class. With this we were frequently having to use our skills developed in our CSUSM career to search and find solutions and help online. This proved to be difficult since other languages such as Java were often what the solutions to problems were in, and while Java was close it was never exactly what we were looking for, leaving documentation manuals as the closest form of solution we could find. If anyone has ever read one, you know that they are not the easiest form of prose to comprehend, proving to be another challenge. To overcome this, we turned to more of a different angle finding similar projects that people had worked on in social media such as YouTube or even Instagram and GitHub. This allowed us to use similar work and ideas as examples and learn from them to be able to apply these concepts to our own work. A big challenge for us as a team was the RecyclerView, we struggled to get it implemented the way we wanted it to be done. In the end it is not what we had wanted it to be fully, but with outside life and other finals we had to keep it as it was. It was brought back to the point of it being really difficult to find proper documentation online in Kotlin. There were examples for this in Java and there were examples relating to pictures, but we needed help with audio files. This challenge still proved to be an insightful one and a learning experience for the future.

Limitations:

- A limitation of our project was our ability to fit ten different concepts proved to be challenging because we had to come up with different ways for the concepts to fit within the app. At its core, a voice recorder is a simple application that doesn't have many features other than its main functionality. Because of this it was difficult to include certain features because some just didn't make any sense to add.
- Another limitation is the ability to find Kotlin information online. Majority of android development research or examples tend to be in Java. Websites like StackOverflow typically have more responses to examples in more prevalent languages and that makes it harder to find examples in Kotlin.
- Making what we wanted to make also proved to be difficult. We have to adhere to the API's provided by Google, and they are very limited in scope and capability. The Android documentation does not provide good examples nor good snippets that can typically be found on other development platforms. We ended up using the MediaRecorder class, but the options that it provided are limited and seemingly deprecated.

List of implemented concepts in your project :

- **Firestore:** We utilized firestore for authentication of the login portion of the project, making it so users can sign up with a new account and login after. This enabled us to be able to tell the user if the login was successful or there was an error with the entered user or password. We are storing the user's name, email, and password in Firestore.
- **Bottom Navigation:** We included a bottom navigation bar to make it easier for users to navigate between the different pages of the application. After the login the bottom navigation will appear on each screen task. Out of all the navigating options, we felt that the bottom navigation was the most convenient and ergonomic on a handheld device
- **Fragments:** We utilized fragments to make our pages have more individual functionality of specific parts of the application such as the recording fragment, which is the most important part of the application as a whole.
- **Unit Testing:** We utilized unit tests to make sure that our application would run soundly and was robust enough to use. We decided to implement Unit Testing to find and fix any bugs that could be in our application.
- **MVVM Pattern:** The MVVM pattern was used to organize our code so that the front end ViewModel would handle button interactions while the backend Model would handle the more complicated code portions and calculations.
- **Permissions:** Since this is a voice recording application, we needed the phone's permission to use the microphone in order to record our voices. Permissions were also necessary for access to the device storage.
- **Material Designs:** Many material designs were used to create and style the application, we went for a modern simplistic design.
- **Notifications:** Once you record we created toast notifications to alert you to the successful saving of the new voice recording. This toast notification included the name of the file it was being saved as.
- **ViewModel LiveData:** The livedata we have in the application is the timer that shows a user how long their recording has been going for.
- **RecyclerView:** We have a recyclerview to show the list of recordings that have already been saved to the storage of the phone

A table of group members and their contributions:

Benjamin Hoang:	Firebase, Login, Fragments, ViewModel Live Data
Angelica Olmedo:	Fragments, RecyclerView, Bottom Navigation, Writing Report
Jeremy Magana:	Recording button functionality, Application Styling, Permissions
Jarrett Koran:	MVVM, Unit Test, Google Play, Writing Report

Conclusion:

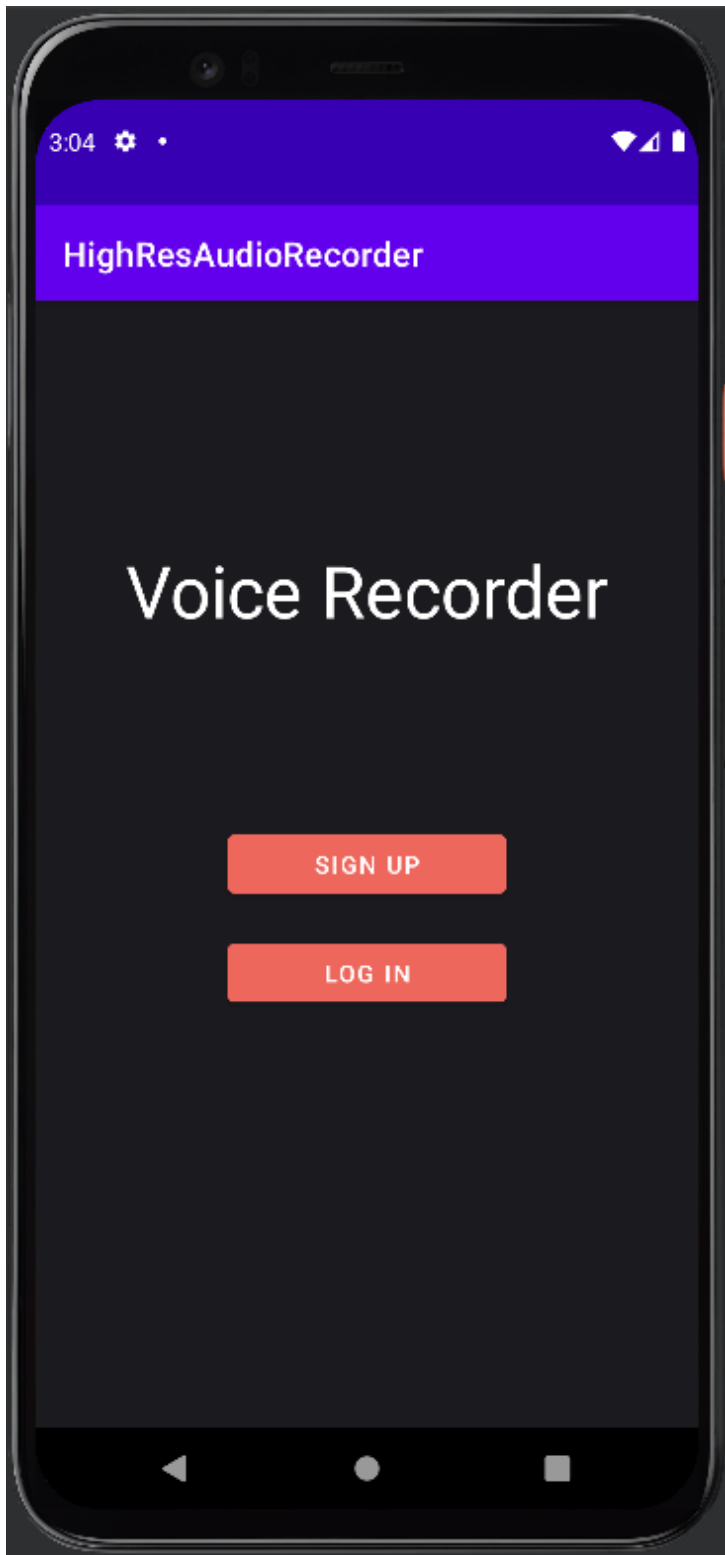
Our goal was to be able to create an application that could make recording on a mobile device sound better than default applications but remain easy and user friendly. Overall we believe that we accomplished this goal and were able to create a recording service that exceeds the quality of the default application and remains easy to operate. While the ease and user experience was a big part of our goal we also wanted to get more out of the potential that newer mobile devices provide because of the exceptional technology implemented in them. I believe as software developers we were able to accomplish this and get better quality out of the technology than the previous default applications were able to. Overall, this project was a great learning experience. We were able to build this audio recorder application with all the knowledge and tools from this course as well as use our background knowledge from previous courses to be able to learn on the go and adapt when we needed to. We are looking forward to users downloading and using the application.

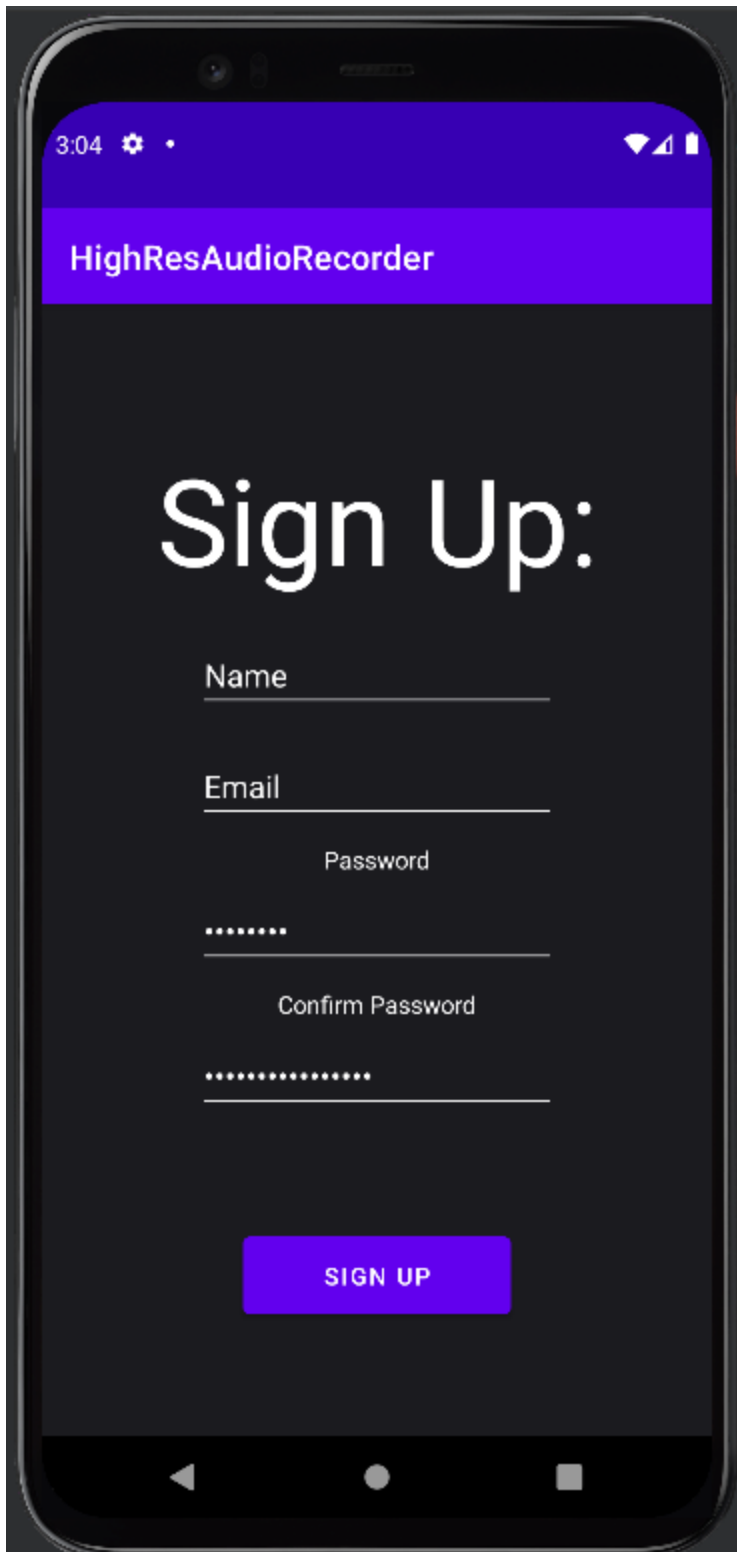
References:

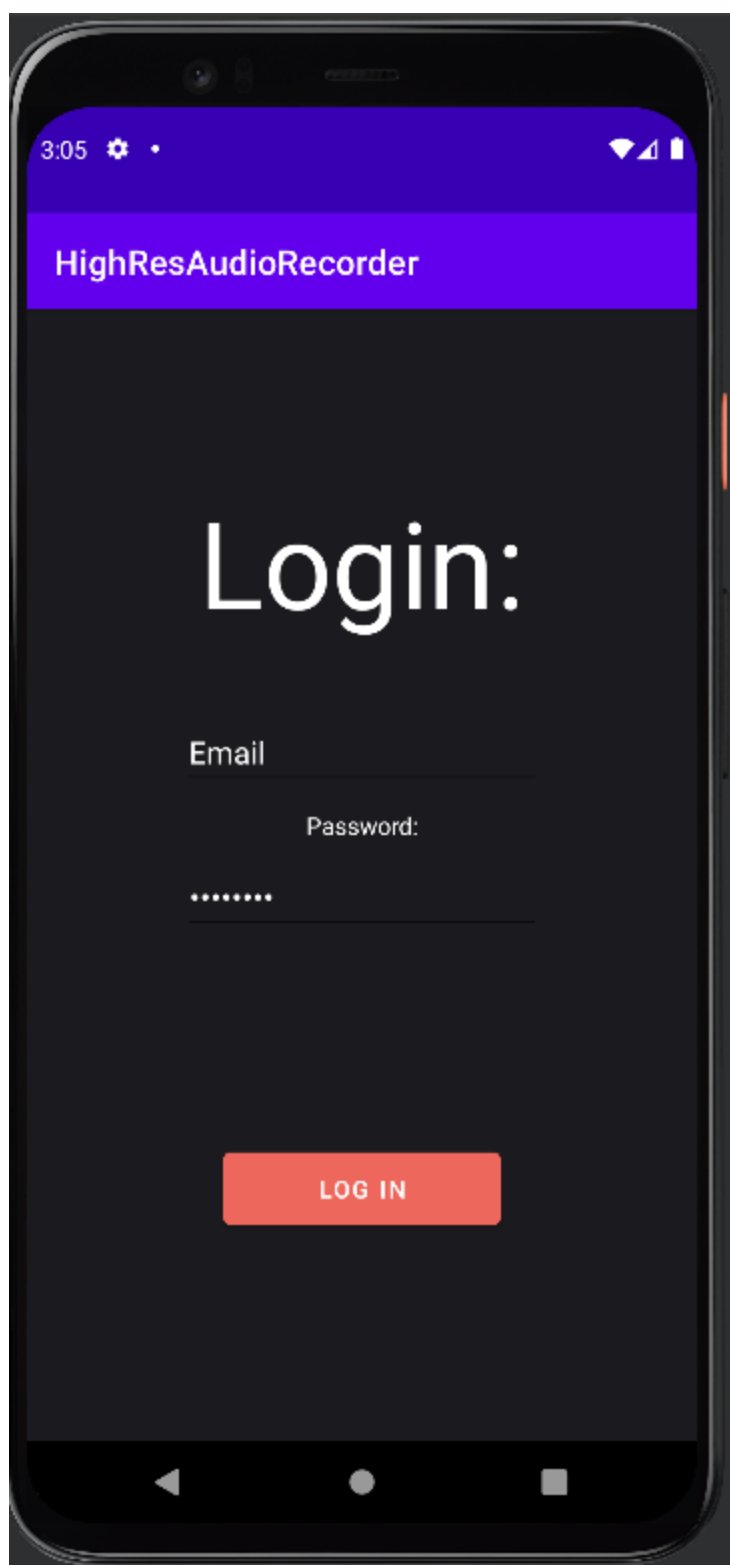
This project required a lot of research as we have not worked with audio recording implementation before. We had to collect information regarding collecting and saving audio. In addition to any course materials, we used a few online resources. References include the following.

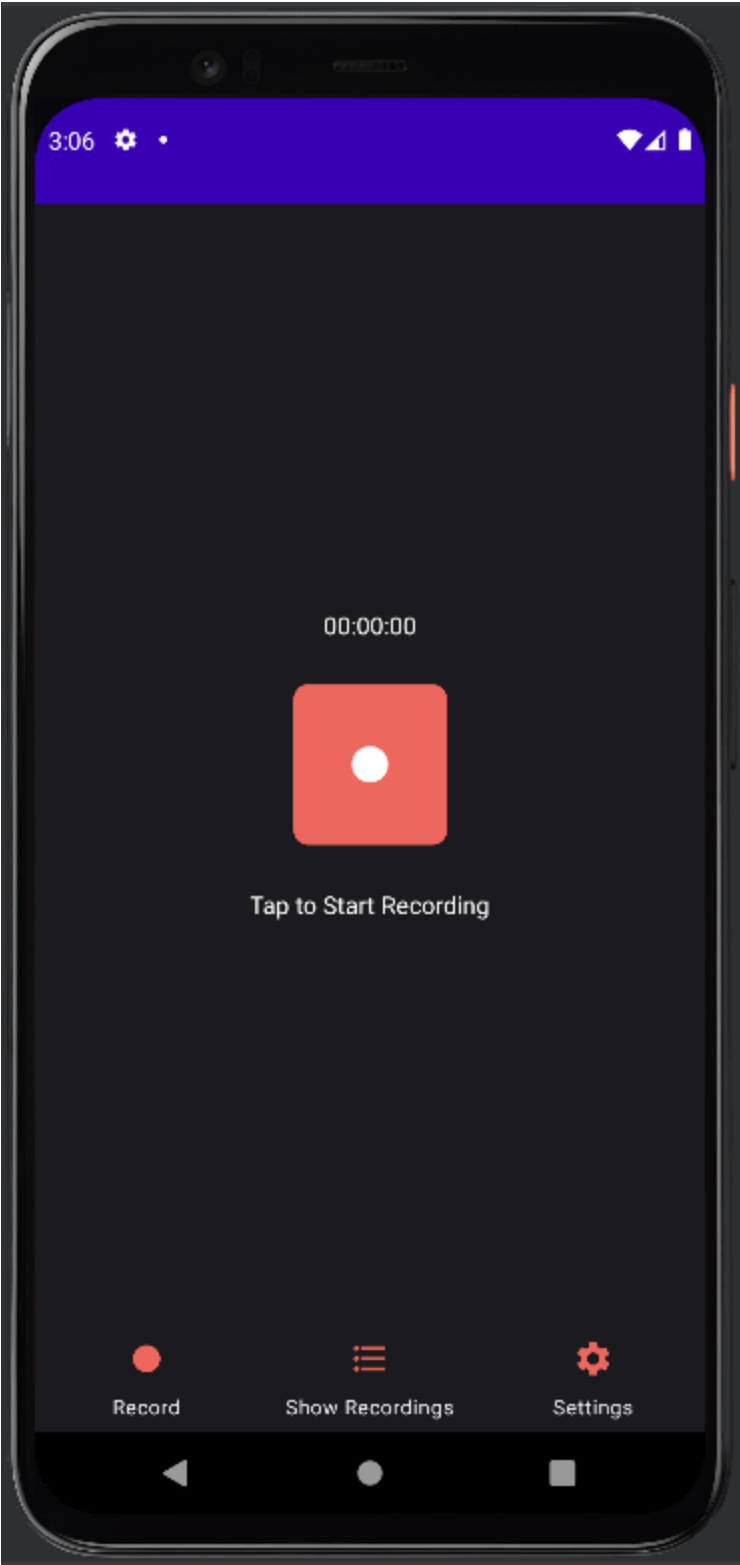
- developer.android.com/guide/topics/media/mediarecorder
- <https://stackoverflow.com>
- <https://www.youtube.com>

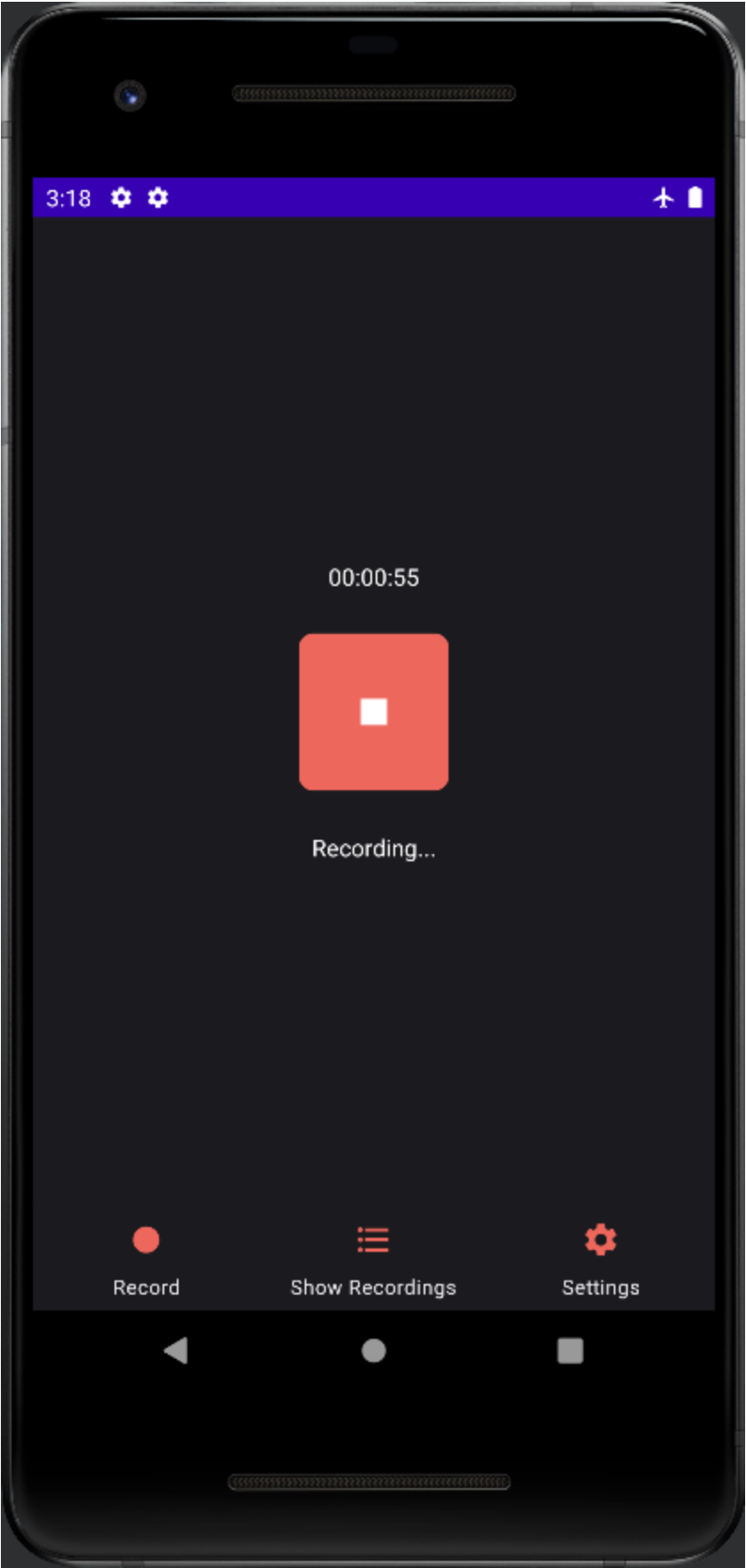
Appendix (All app screenshots)

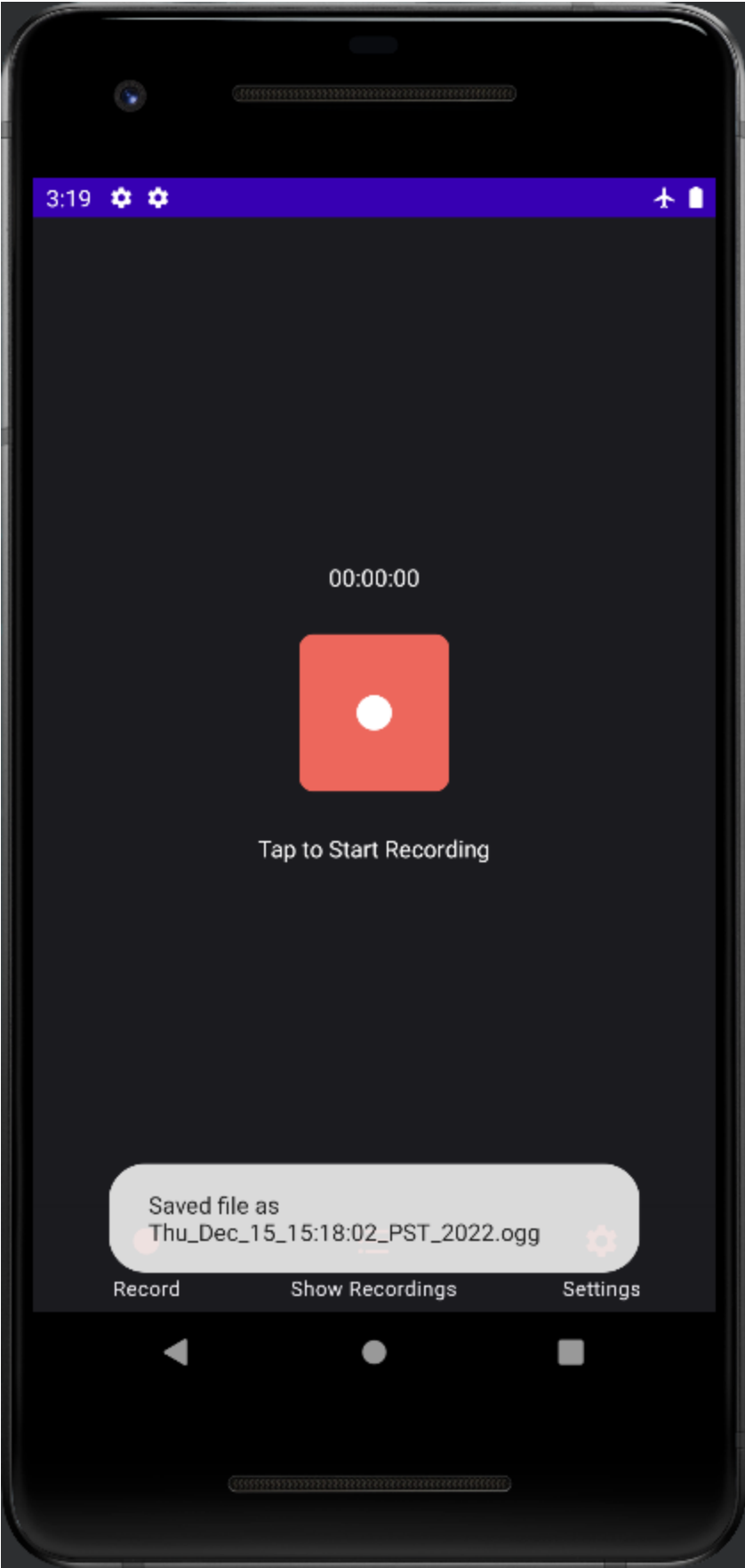












Google Play Store Submission:

Dashboard



High Res Voice Recorder

com.finalteam6.HighResAudioRecorder · [View on Google Play](#)

Update status

 In review · [Go to Publishing overview](#)

Production

Active · 0 active devices · 1 country / region