GitHub Username: TBS99

Randomness

Description

It’s an app that will get you out of your comfort zone giving you random songs or cast to hear which might interest you in another way

Intended User

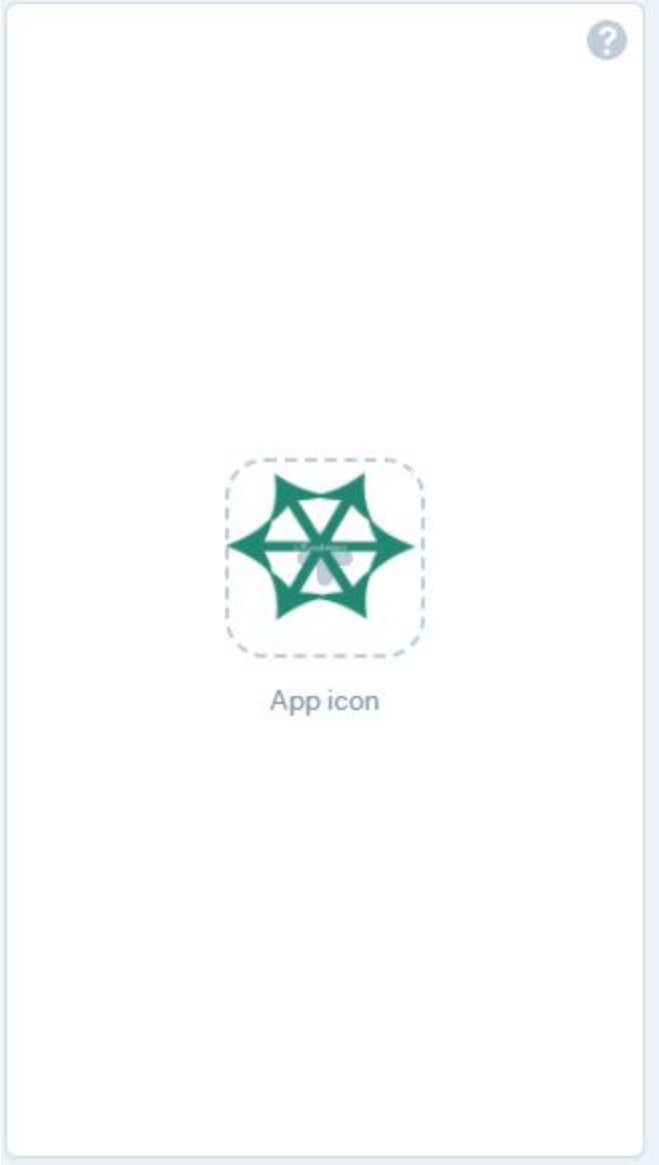
All Sorts of people and specifically targeting 16-25 years old for as it’s the most random years of all

Features

- Speech Recognition
- Media Playing
- Random Shuffling
- Getting People out of Their Comfort Zone

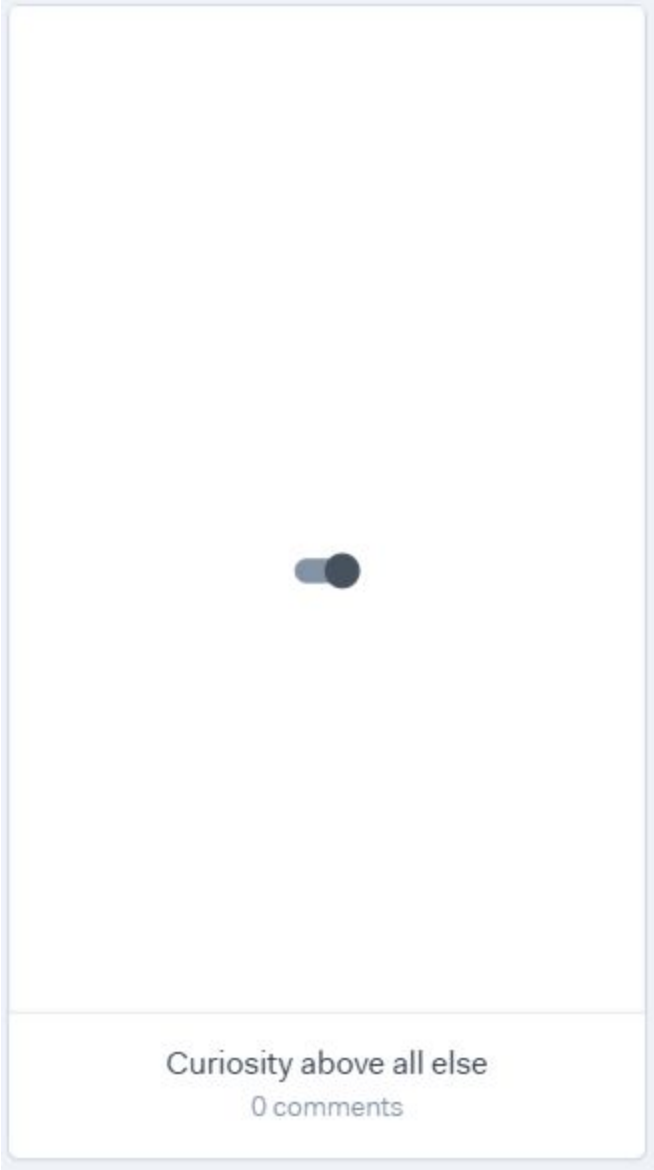
User Interface Mocks

Screen 1



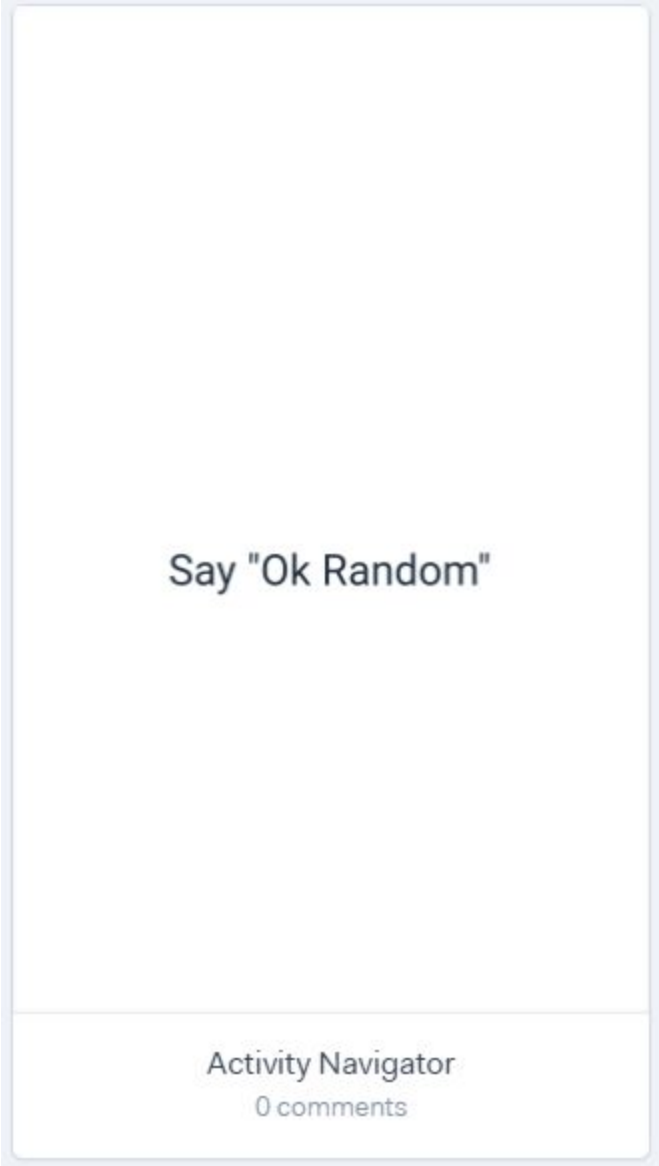
This is the Splash Screen which will have only the logo

Screen 2



Adding just a button is always curious and every single person will wish to press it and by that it starts the speech feature for the next screen

Screen 3



Little Guidance in the speech recognition is the Random Word which in second stage will be Secrecy or Custom which is a subscribed feature that will be added later on
However Random allows the user to enter the app and actually enter

Key Considerations

How will your app handle data persistence?

Firebase Realtime Database will handle the users Device ID
As well as the Metadata of all the songs or music that is uploaded to Firebase Storage due to another interface of the Admin to Firebase Data which has its own other implementation

Describe any edge or corner cases in the UX.

For example, how does the user return to a Now Playing screen in a media player if they hit the back button?
Really interesting as i am actually using music for my app , the Now playing music is Broadcasted or in other words saved for the whole app so Now Playing will always be showing the current or the most recent one unless the user leaves this random list and goes out of the cast only then they will go to another random list

Describe any libraries you'll be using and share your reasoning for including them.

For example, Picasso or Glide to handle the loading and caching of images.
Glide for Now for songs image
ExoPlayer for media Player
Firebase Analytics Realtime database , Storage

Describe how you will implement Google Play Services or other external services.

Firebase Analytics will be used to monitor the current reach
While Firebase Realtime database , Storage will maintain the data of the users as well as the metadata of the music

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

- Configure libraries
- Start Modules , sub parts and implementing the mvp to see what's happening during the fetching of the media from the storage of firebase while always watching internet connection

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for MainActivity
- Build UI for Speech Recognition
- Build UI for the Recycler View which will have 10-20 items from the Firebase Storage
- Build UI for the detail of each item

Task 3: Speech Recognition

- Create layout Screen 2-3
- Module for the Speech recognition
- Test it
- See if my POC(Proof Of Concept)is alright to go on to the next step

Task 4: Songs or music from Firebase

- Create layout Screen 4
- Create the RealTime Database Structure
- Create the Classes in android studio
- Implement the firebase
- Upload data to firebase storage
- Test if the uploading is correct
- Build fetching module
- Test if its working correctly

Task 5: Song Details

- Create layout screen 5
- Check if the metadata of the song is also on Realtime Firebase Database
- Check if Glide or Picasso is getting the image correctly