

Easy-Teleprompter

GitHub Username: [Emmaunel](#)

Description

With the Easy-Teleprompter, there is no need for flash cards or notes for your speeches. Easy-Teleprompter App is the perfect pocket size teleprompter to practice your speeches.

The app provides the user with a easy to understand interface that can be customize to the user's preference.

Read your speeches or script with confidence with Easy-Teleprompter!!!

Easy-Teleprompter is designed for presentation, lectures, broadcasters, filmmaker, business professionals or any public speakings!

Intended User

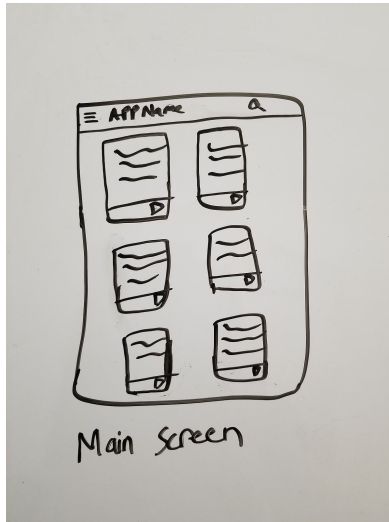
This app is designed for students, lecturers, broadcasters, youtubers, singers and presentations in general.

Features

- App is written solely in the java programming language
- Saves speeches
- Sync speeches to database incase user loses files
- Record video while reading speeches
- Adjust script's scroll speed based on user preferences
- Adjust script's text size, color
- Ability to change how the scripts are being displayed(grid or list)

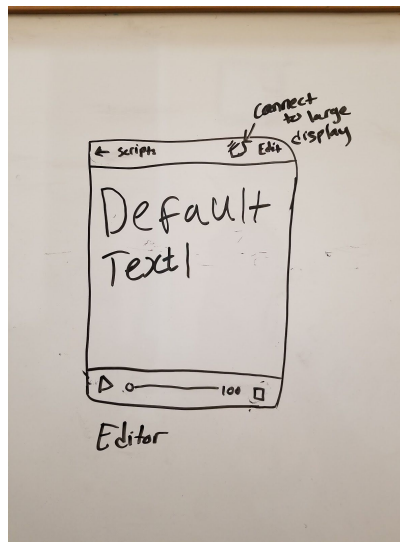
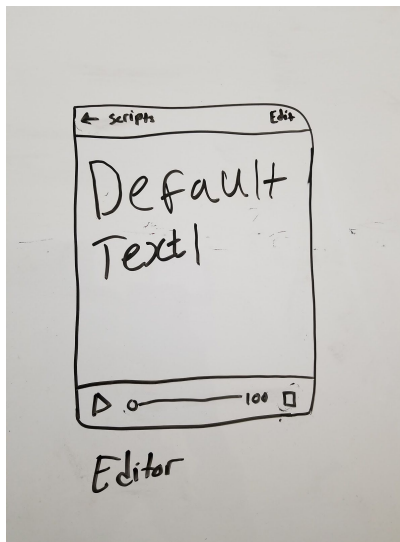
User Interface Mocks

Main Screen



The screen has all the user's saved scripts and speeches. The play icon will start the scrolling of the scripts on another screen. The user also has the opportunity to search through the scripts.

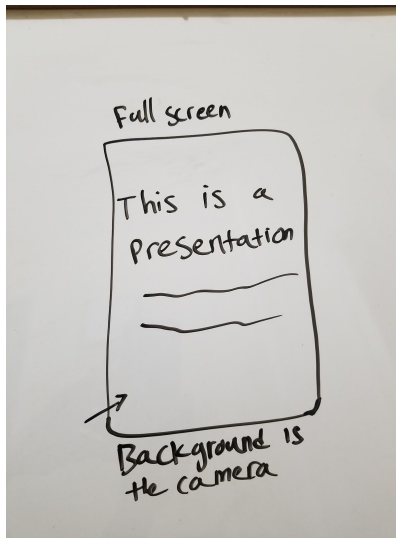
Screen 2



This is the screen users will spend most of their time. The first picture will be able to edit their scripts and do other things like change the font or color of the texts.

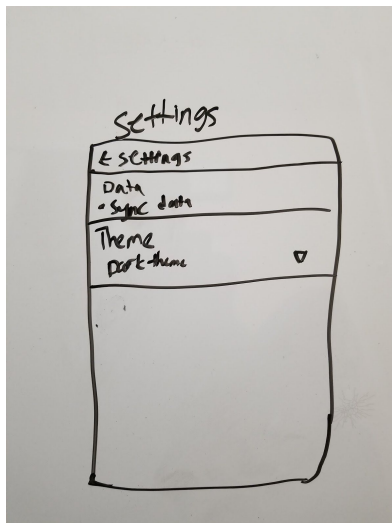
The second picture show an alternate version where there is "connect to TV" button. This will allow users not only to read the script off their phone but any smart tv.

Screen 3



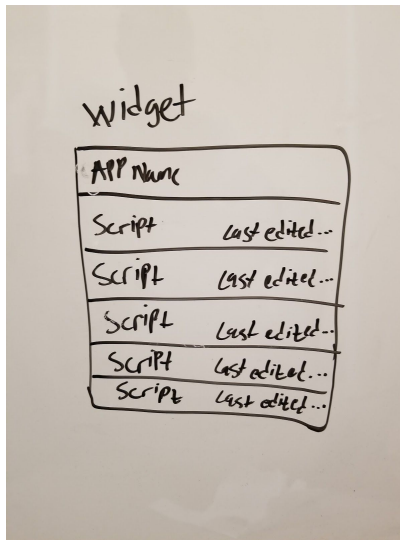
After the user writes their script, they have the choice of recording themselves while reading. And to achieve that, there will be two fragments overlapping each other, one for the text and the other for the camera.

Screen 4



Users are able to change the theme of the app, how the scripts are displayed (grid or list), sync their scripts to external database and change the default scrolling speed.

Widget



The widget will display all the scripts in a list and also tell when the it was last edited. When an item is clicked, it opens that specific item in the app.

Key Considerations

How will your app handle data persistence?

Room Database for local storage

Firebase for external storage

Shared Preference for user's settings

Describe any edge or corner cases in the UX.

While recording, if the user presses the back button, the recording will be stopped.

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

Library	Version
Ahoy! Onboarding to introduce the app to the user after install	1.0.4
Butterknife for view binding	8.8.1

Room Database for local storage	1.1.1
Android design library for material design	27.1.1

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

Admobs and Analytics will be used

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

- Create a new project on Android Studio
- Configure the app with Firebase database
- Implement Butterknife and Onboarding
- Implement Room Database
- Implement Espresso testing

Task 2: Implement UI for Each Activity and Fragment

- Dashboard Screen - Activity
- Onboarding
- Login Screen
- Editor - Activity
 - Editing Screen - Fragment
 - Recording screen - Fragment
- Setting Screen - Activity
- About Screen - Activity

Task 3: Setup Firebase

Design a data schema so that it is easy and fast for the app to get data from the database. Also Implement email and google login.

Task 4: Add data persistence

Add firebase database and room database to gradle file.
Create a Dao, Viewholder and live data for the room database.
Create an Application class to initialize firebase

When the user sync the local database will firebase, Async task will be used to perform that process. And Room Database will use JobDispatcher to update the database every certain hour(depending on the user's preference)

Task 5: Create paid flavor

The paid version will remove ads from the main app.

Task 6: Write UI Test

Create UI tests to automatically test the app's UI

Task 7: Clean up code

Enable RTL layout switching on all layouts
Put all hardcoded string in the string.xml
Put all the common attributes(for instance, for all text) in style.xml

Task 7: Prepare app for launch

- Create an icon
- Prepare play store listing
- Create a video presentation