

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Setup data module](#)

[Task 3: Setup dependency injection with dagger](#)

[Task 4: Implement UI for Each Activity and Fragment](#)

[Task 5: Connect UI to data module using viewmodels](#)

**GitHub Username:** tjohnn

# Teleprompter

## Description

The app is a professional teleprompter system for android phones and tablet. It prompts speakers to read text from it as if they are actually saying it offhand.

## Intended User

This app is for news casters or speakers at any gathering.

## Features

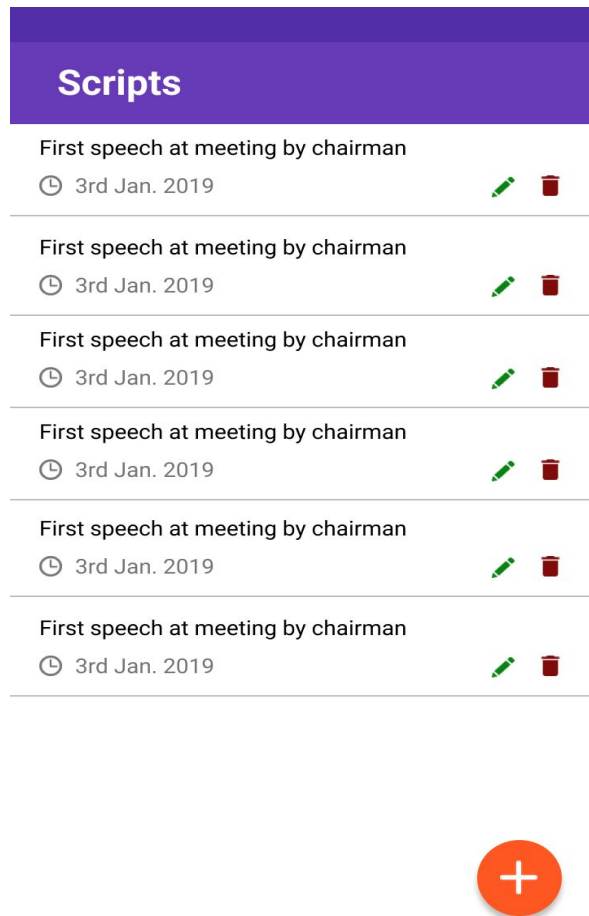
Main features of the app are as follows:

- Saves scripts to be read
- Allow importing of text files as script

- Prompts the script to be read to the speaker.
- Allows control over the speed at which the text being prompted is scrolling.
- Ads for free version

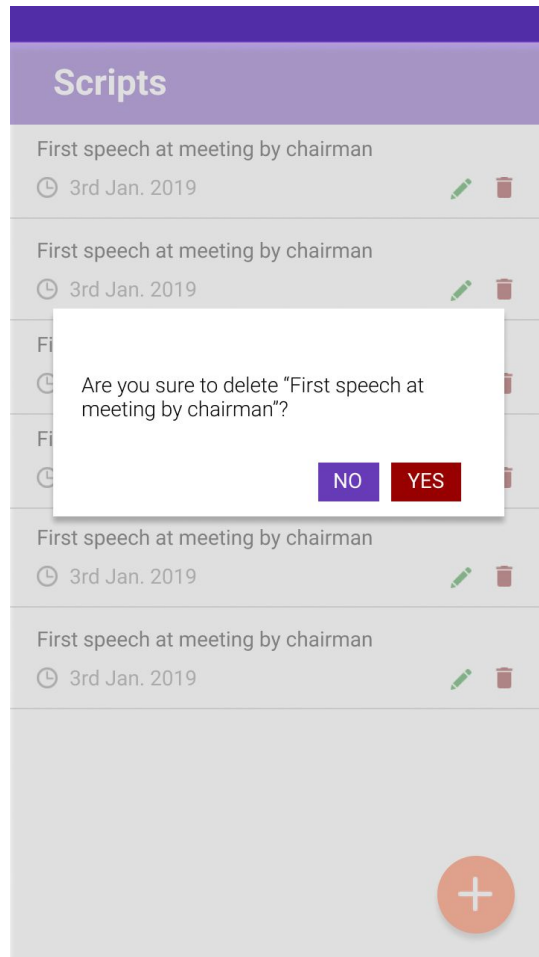
## User Interface Mocks

### Scripts list



The first screen of the app which shows list of scripts added by user and allows navigating to create script screen. Clicking on the list item starts the teleprompting screen for the item clicked

### Delete Script



Delete script popup confirmation screen

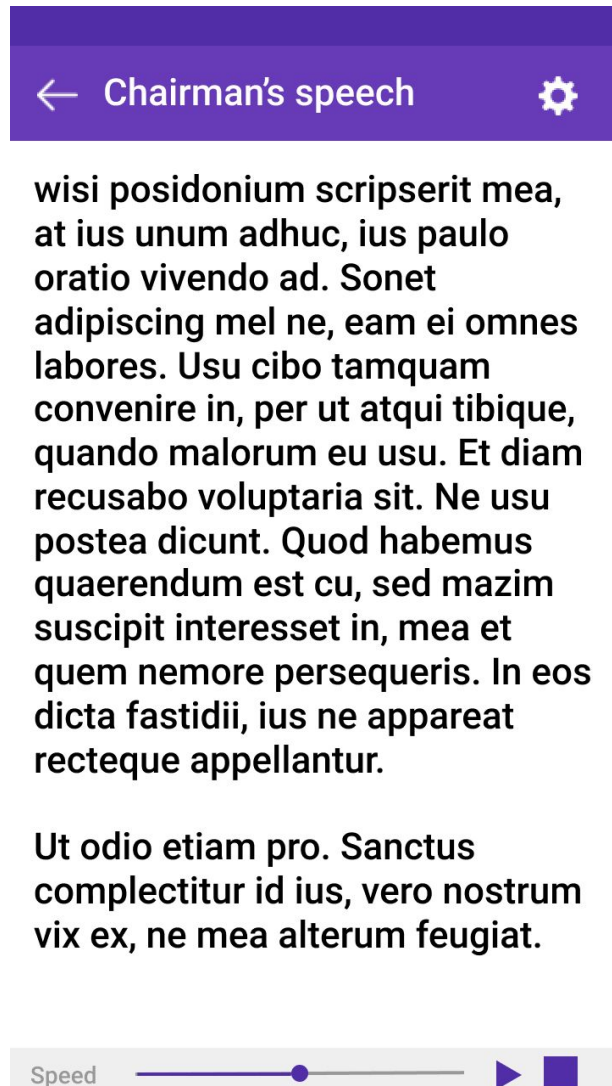
## Add/Edit Script Screen

 **Add Script** 

**Submit**

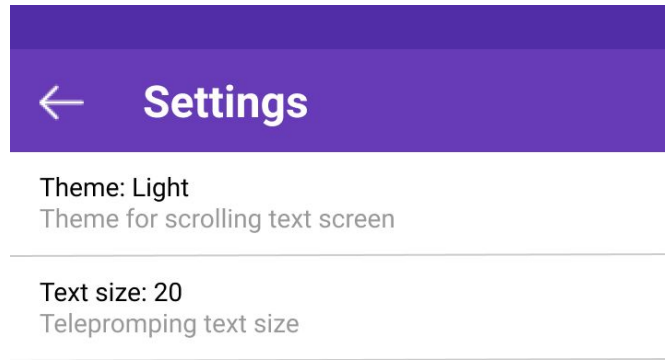
The screen that is used for adding or editing a script detail, it also allows importing of a text file when user clicks on the import icon in options menu

## Teleprompting Screen



Teleprompting screen that shows the scrolling text to user, with controls to play, pause, stop and to change the scroll speed of text. It also allows navigation to teleprompting settings screen.

## Settings Screen



Teleprompting settings screen

## Key Considerations

## How will your app handle data persistence?

Data will be saved in sqlite database and operations on the database will be carried on with the help of the Room ORM

## Describe any edge or corner cases in the UX.

Opening the teleprompting screen of the app continues the scrolling of the script text from where it stopped the last time it was opened, except the reading was completed where it restarts teleprompting. To take care of this, the last scroll position of each scripts are being saved in the database in the onDestroyed() callback of the Fragment displaying it. Another edge case is handling of screen rotation while a script is being teleprompted, this is handled by saving the scroll position in onSaveInstanceState bundle and resuming the text scroll at exact position.

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

Room ORM for handling database transactions.

Butterknife for binding views.

Dagger for dependency injection.

Timber for logging.

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

Google play service will be used for showing adverts on free version of the app

## Required Tasks

Tasks to build the app is divided into 5 sub tasks below:

### Task 1: Project Setup

Steps to setup project:

- Create a new Android Studio project
- Setup libraries and their respective versions
- Create keystore and setup gradle to run release version of the app from android studio
- Setup google ads services

- Setup gradle for free and paid version

## **Task 2: Set up dependency injection with dagger**

DI sub-tasks:

- Create dagger app and integrate with AppComponent
- Setup activity injection module
- Setup injection of viewmodels with viewmodel factory

## **Task 3: Set up data module**

DI sub-tasks:

- Setup database
- Create data repository module to communicate with data access objects
- Integrate data setup with dependency injection

## **Task 4: Implement UI for Each Activity and Fragment**

UI sub-tasks:

- Build UI for script list screen
- Build UI for adding new script
- Build UI for editing script
- Build UI for teleprompting scripts
- Build UI for teleprompter settings
- Integrate google ads with Scripts screen for free app version

## **Task 5: Connect UI to data module using viewmodels**

Implement viewmodels and make them bridge between data and screens

Sub-tasks:

- Create viewmodels and inject them into respective activities/fragments
- Implement data synchronization with viewmodel livedata
- Observe livedata from viewmodels in the activities/fragments and bind data to ui