

## Description

Common Project Requirements

Intended User

Features

User Interface Mocks

Screen 1

Screen 2

Screen 3

Screen 4

Screen 5

Screen 6

Screen 7

Key Considerations

How will your app handle data persistence?

Describe any edge or 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 or other external services.

Next Steps: Required Tasks

Task 1: Project Setup

Task 2: Implement UI for Each Activity and Fragment

Task 3: Create layouts

Task 4: Widget design

Task 5: Enhancement of design

**GitHub Username:** [Vitiello Antonio](#)

# Teleprompter

## Description

Teleprompter is an autocue style app that transform your mobile in a display device that prompts the person speaking with an electronic visual text of a speech or script, similar to using cue cards but automatically, allowing the speaker to maintain eye contact with the public.

## Common Project Requirements

- Teleprompter App is written solely in the Java Programming Language ver. 1.8.
- Utilizes stable release versions of all libraries, Gradle plugin ver. 3.1.3, Gradle wrapper ver. 4.4 and Android Studio ver.3.1.3.
- Use Loaders with content provider.
- Implement Intent Service for Widget backend communication.
- App supports accessibility using: 'contentDescription' for static and dynamic elements, 'hint' for static and dynamic elements to indicate each element's purpose, 'labelFor' attribute when labeling View objects that serve as content labels for other View objects.
- All strings will be defined like resources in res/values/strings.xml.
- All images will be vector drawables in res/drawable, or when it is not possible will use WebP images.
- Colors will be defined like resources in res/values/colors.xml
- Styles and themes will be defined in res/values/styles.xml
- Will be enabled RTL (Right to Left) layout switching for languages written from Right to Left and adopted Start and End property instead of Left and Right equivalent. For example android:paddingLeft will be replaced with android:paddingStart.

## Intended User

Teleprompter is intended for presentations, lectures, broadcasters, filmmakers, musicians, singers, business professionals, politicians, students, or any public speaking!

## Features

App is written solely in the Java Programming Language

- Unlimited scripts
- Full screen mode
- Mirror mode (for use in teleprompter rigs)
- Background color selection
- Adjust the text size, color and font family
- Script import from clipboard (Copy & Paste)
- Adjust the speed of the teleprompter
- Delete scripts

## User Interface Mocks

Edited with Wire Flow Wireframe Design app from Google Play:

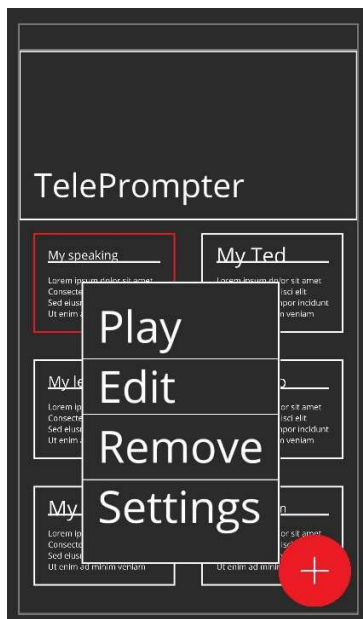
<https://play.google.com/store/apps/details?id=com.ludomade.blueprint>

### Screen 1



Teleprompter starts with this layout that presents a list of cards where every card represent a saved script. There is a floating button at bottom end of the screen to add a script or long press on a list item for more actions on the card.

### Screen 2



A long press on a card opens a menu with the possible actions on that card:

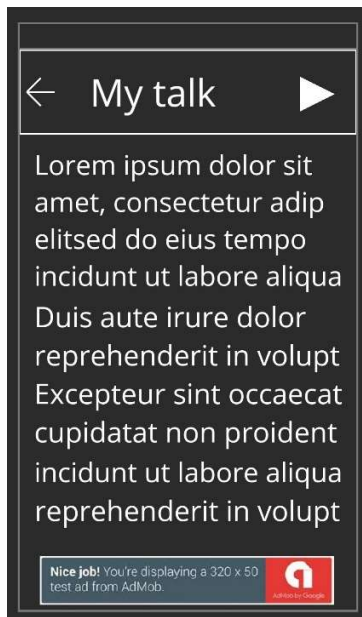
- Play, to start script autocue
- Edit, for editing this script
- Remove, delete this script
- Settings, for script playing

### Screen 3

The image shows a mobile app screen titled "New script". At the top left is a back arrow icon. Below the title are two text input fields: "Title" and "Body". The "Body" field is larger and occupies more space. At the bottom right corner, there is a red circular button with a white checkmark, representing the "Add script action (FAB)".

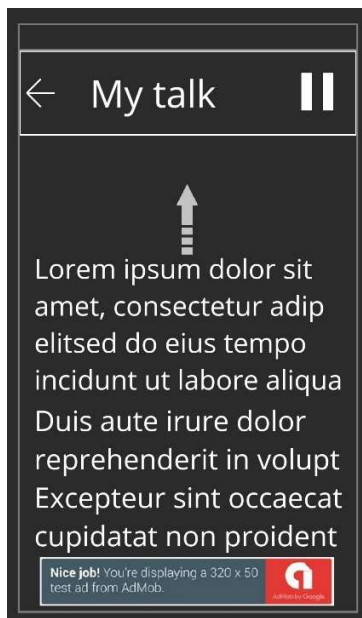
Add script action (FAB) opens the edit mode layout, where user inserts script title and body. A FAB at bottom end of the screen let user save the new script.

## Screen 4



A single tap on a card in main page opens the autocue page where the user can play the script using the button icon in appbar. For free flavor of Teleprompter app an ad banner will be presented at bottom of the screen just in this page.

## Screen 5



While playing autocue, the user can pause it using the button icon in appbar or swipe to fast advance or rewind the script.

## Screen 6



The Settings page intended for customizing autocue presentation.

## Screen 7



User can add a widget for Teleprompter on phone launcher, it provides the script title and a small description of a last played script.

## Key Considerations

### How will your app handle data persistence?

Teleprompter will use SQLite database to store scripts and ContentProvider to fetch stored scripts.

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

While playing the script user can pause or play autocue.

If user hit the back or UP button it behave naturally and takes the user back to the previous screen, if user tap on the same script the new play starts from beginning of the script.

Teleprompter app will provide different layouts for the main page: for large screens or landscape orientation will set automatically number of columns to fit the screen.

Widget can show only the last played script and if user add more than one widget on launcher they all contains the same script.

Teleprompter does not provide a search feature in this first release.

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

Google AppCompat support library, to provide support for low level API's.

Google Preference support library, to support settings page.

Data Binding support Library, to bind views to data and interact with them.

Google Services Ads, to provide Ads banner in app.

Google Services Analytics, to track mobile traffic.

Google Design support Library, to provide cards creation.

Google RecyclerView support Library, to provide RecyclerView.

JakeWharton Timber, for logging any kind of verbose or errors in debug build but not in production free or paid flavor.

Any library to provide FAB (Floating Action Button) Animations if useful.

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

- AdMob, to show ads in free version.
- Analytics API, for analysis of app usage.

## 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

- Update Android Studio and create new project, with a name, Teleprompter, and a primary domain path, target SDK 27 and min SDK 19 to targets over 80% of devices. Project will have an empty Activity with his layout.
- Update Gradle wrapper and Gradle plugin to latest stable version.
- Adding project dependencies and third-party libraries.
- Choosing color palette for app.
- Configure free and paid flavors for application.

### Task 2: Implement UI for Each Activity and Fragment

- Planning UI on paper and make a rough diagram.
- Implementation of the plan starting from basic design.
- Starting from designing main activity, go through fragments, other activities and then implement adapters.
- Create shared preferences for settings.

### Task 3: Create layouts

- Create a layout for every Activity and Fragment using CoordinatorLayout AppBarLayout CollapsingToolbarLayout in main page.
- Create layout for RecyclerView items.
- Adding layouts and views for dialog, error messages and progress bars.

### Task 4: Widget design

- Designing interface of widget.



## Task 5: Enhancement of design

- Enhancement of design by implementing standard and simple transitions between activities.

Add as many tasks as you need to complete your app.

---

### Submission Instructions

- After you've completed all the sections, download this document as a PDF [ File → Download as PDF ]
  - Make sure the PDF is named "**Capstone\_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
- Add this document to your repo. Make sure it's named "**Capstone\_Stage1.pdf**"