GitHub Username: Priyanka Nandiraju

Teleprompter

Description

Teleprompter is a display device that prompts a person speaking with an electronic visual text of speech or script. The idea is to help users to give presentations without memorizing the script. The teleprompter app builds on a classic teleprompter and exemplifies an android app following Android fundamentals and project specifications. In this app, a user can type and save the script which can then be teleprompted with a custom settings for font size, scroll speed, etc.

Intended User

- Presenters

- Youtubers
- Students
- News readers
- Singers, etc.

Features

- Save short scripts
- Detail view of all saved files
- Ability to change teleprompting script's size, color.
- Adjust script's scroll speed based on preference.

User Interface Mocks

Click this interactive link here to experience how the app's wireframe looks like.
(*Developed using Adobe Experience Design CC (Beta)*)

Screen 1

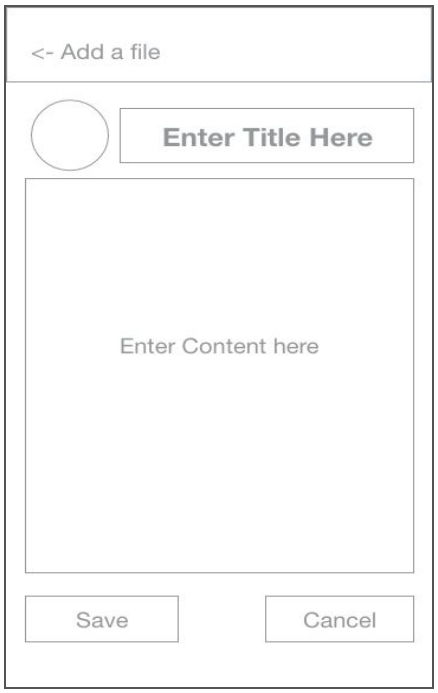
Phone View

Tablet View

The Main activity consists a toolbar which will collapse when the user scrolls through a list of saved files. Users can 1. add a new file by clicking on the floating button; 2. navigate to settings by clicking on the menu icon to the right of toolbar; and 3. go to specific file and start teleprompting by clicking on individual cards. 4. When user swipes the item to the left, user can download the file.

Admob will be displayed on the bottom of the screen.

Screen 2

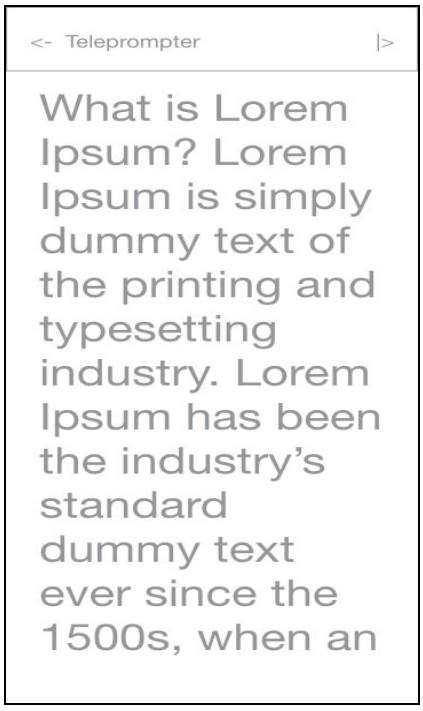


Phone View

Tablet View

Add a file" screen provides the interface for users to save a file on the device after entering the title and content. User can also upload an image as reference for each file.

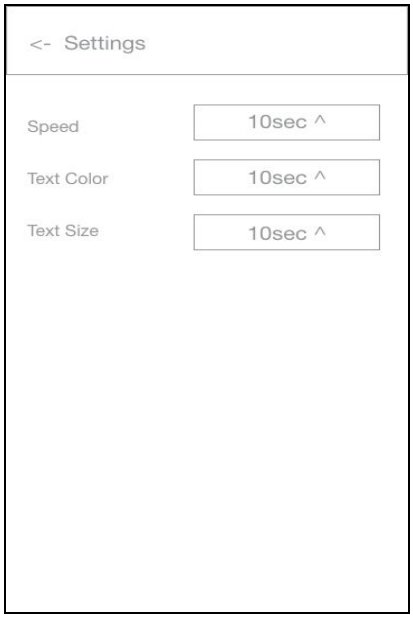
Screen 3



Phone View & Tablet View (In landscape)

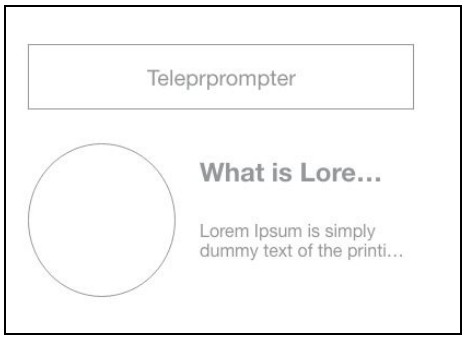
Users can start the "teleprompter effect" by clicking the play button to the right of the toolbar.

Screen 4



Phone View & Tablet View (In landscape)
Users can set their prefered scroll speed, text color and text size in the settings screen.

Screen 5



Widget UI

Key Considerations

How will your app handle data persistence?

Teleprompter app will use Content Provider to save user's script file and Sharedpreferences to store simple user preferences data from settings.

Describe any edge or corner cases in the UX.
N/A

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

App will use Butter Knife for field and method binding.

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

Admobs and Google Analytics will be implemented.

Next Steps: Required Tasks

Task 1: Project Setup

- Import and build on the latest version of Android Studio (Eclipse users take note)
- Configure butterknife library
- Create packages for Content Provider, Utils, etc.

Task 2: Data persistence

- Build Framework for Content Provider to save files
- Use shared preference to save user's setting preferences
- App uses Loaders to move data stored in Content Providers to views.
- AsyncTask will be used to download the saved content onto the device as a file.

Task 3: Implement UI for Each Activity and Fragment

- Build UI for MainActivity by implementing Collapsible Toolbar, floating button, card views, etc.
- Build UI for Add files Activity
- Build UI for Settings Activity
- Build UI for Teleprompter Activity

Task 4: Implement Google Play Services

- Create UI for Admob
- Implement Admob
- Analyze possible use cases.
- Implement Analytics

Task 5: Accessibility

- Implement navigation using a D-pad
- Verify all content descriptions

Task 6: Widget

- Framework for widget
- Build widget UI
- Implement widget by providing relevant information to the user on the home screen.

Task 7: Core platform Development

- Cross check Error handling scenarios such as no data, no internet connection, etc
- All strings in strings.xml
- Enable RTL Layout switching on all layouts

Task 8: Building

- Signing configuration, and the keystore and passwords are included in the repository.
- Keystore is referred to by a relative path.
