GitHub Username: the-cybersapien

PromptoDroid

Description

Type or import text and easily teleprompt it.
Access your Text Stories across all your devices automatically. Save/Delete/ Run the Teleprompt and make videos with ease..

- Font and Background colour selection
- Delete scripts
- Adjust the speed of the teleprompter
- Adjust the text size
- In app brightness control

Intended User

The App can be of use to anyone who wants an easy way for text prompts when working on a Video or Audio.

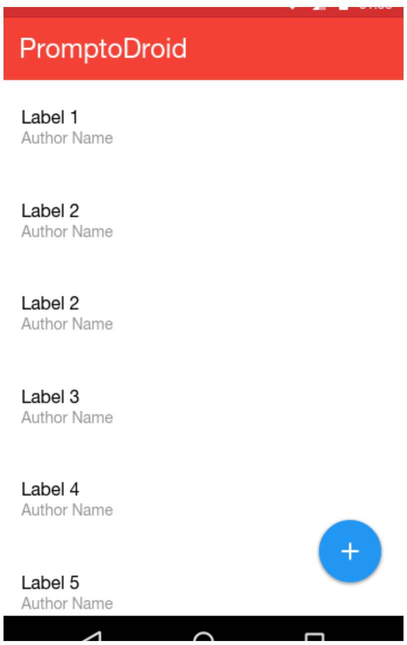
Features

- Save information across devices with Google and Facebook Login
- Adjust Text Size
- Customizable prompter Speed
- In - app brightness control.
- Font and Background color selection

User Interface Mocks

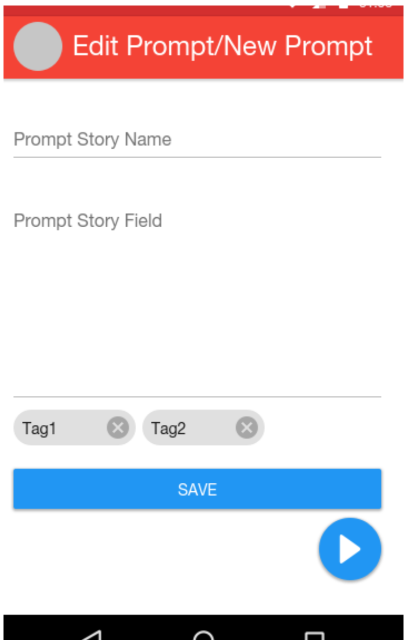
These have been created with the Proto.io tool.
See the interactive version here: https://pr.to/4U43K6/

Screen 1



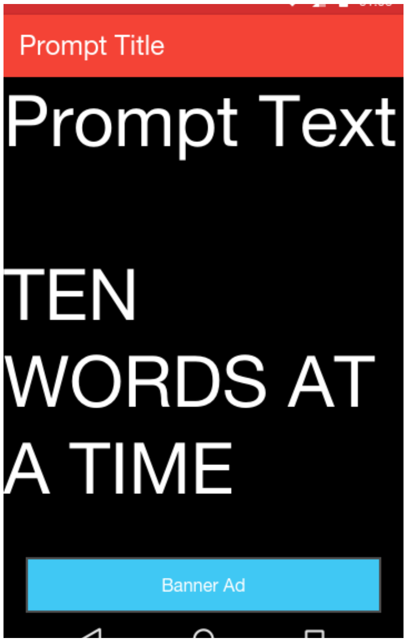
The main screen contains the list of Saved Prompts for the user.

Screen 2



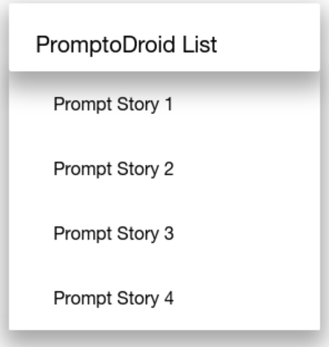
This screen provides an interface to add a new story, view the story and to edit.

Screen 3



The Main Teleprompter Screen automatically scrolls the text slowly, showing 10 to 15 words at a time.

Widget Mock



This is the most basic mock up of the widget, as I couldn't find a better way to make the design. It shows a list of Prompts saved in the app by the user.

Key Considerations

How will your app handle data persistence?

Data will be handled using Firebase Real Time database to support across-device synchronization

Describe any edge or corner cases in the UX.

The main part of UX creation is to automatically change the Teleprompter text at user customizable intervals. This can be handled with an ASyncTask with a Thread Handler.

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

The App will utilize the following Libraries:

1. FirebaseUI- Android
   To handle the Login flow for the activity and to show lists using FirebaseRecyclerAdapter
2. Picasso
   To handle Image loading and caching wherever necessary.

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

Firebase will be heavily utilized in the App in Login Functionality, Realtime Database and Crashlytics for Analytics

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 the basic structure for the App with Activities and design the Database schema for the data.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for the Activities
- Build UI for Fragment
- Build UI for the Widget

Task 3: Implement App Login

- Use Firebase UI and design a login screen

Task 4: Implement Data handling

- WIth the UI, implement a structured way to store data in-app
- Use the FirebaseRecyclerView to show the lists in app.

Task 5: Implement Teleprompter Screen

- Design a custom TextView with multiple Threads to show Text in app.

Task 6: Implement App Widget

- Attach Data to App Widget.