
GitHub Username: [SuZanElSaMahy](#)

VidViewer

Description

Write a brief summary of what app does.

VidViewer media player is a free app that play videos from the Youtube channels according to your choice. All what you need to do is enter the channel id that you want to watch and then you can find all of its videos in front of you.

Intended User

Who is the intended user?

This app is for anyone who loves watching videos from Youtube and keep a track on what it's watched locally on his device.

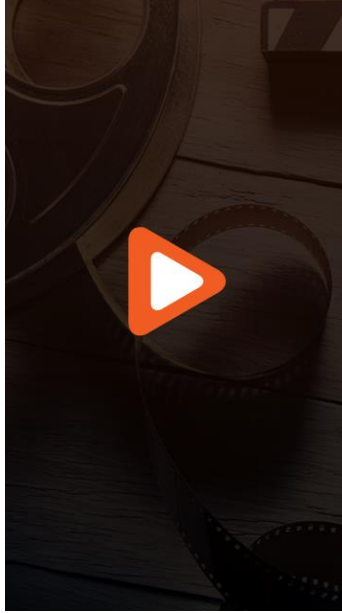
Features

List the main features of your app :

- Uses youtube Api to play videos/playlists.
- It also includes a widget to display channel name that the user is playing its videos.
- Saves a history of videos played by the user.
- App supports Arabic and English languages.

User Interface Mocks

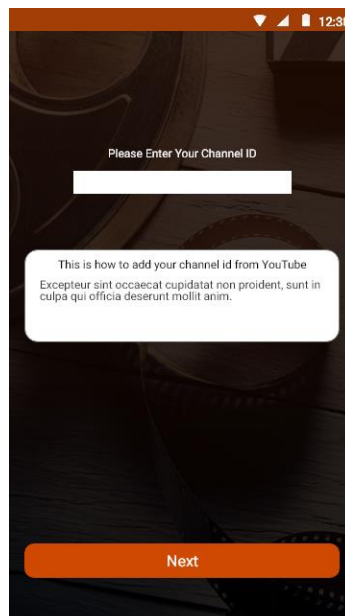
Screen 1 : Splash Screen



-This screen will be presented for the user for 5 seconds.

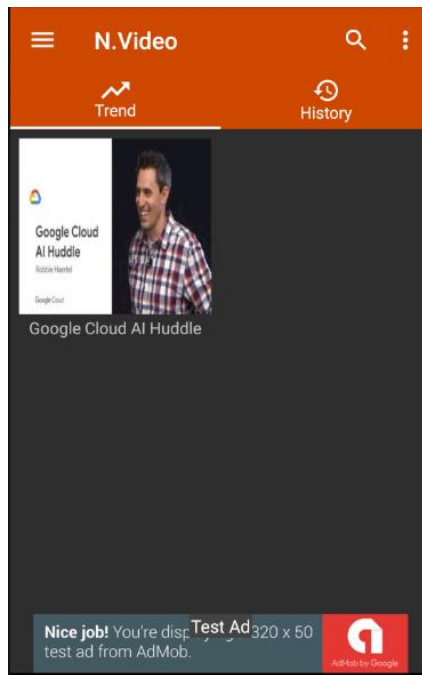
- User will be directed to Add Channel Id Screen.

Screen 2 : Add Channel Id Screen



- This screen asks the user to enter the channel id that he wants to watch its videos.
- Instructions about how to get the id will be illustrated.
- After entering id and on pressing next , user will be directed to the Home Screen.

Screen 3: Home Screen



- This screen shows list of channel's videos.
- There is a navigation drawer that displays list of playlists in the channel and by pressing any of them it will display the contained list of videos
- There are two tabs : 1) to display current videos of a playlist. 2) History to display watched videos by the user.
- User can search for any videos he wants.
- On pressing on 2 dotted icon, 2 options will appear:
 - Change language : from Arabic to English and vice versa.
 - Change channel : the user navigate to the Add channel id to enter a new id.
- Content descriptions will be used with image views illustrating the title of videos.
- When user clicks on any video, he will be navigated to Video detail screen and also will be save to the local database.

Screen 4: Video Detail Screen



- This screen will be presented when a user selects a video from the home screen.
- It will show some info about the video
- User can play the video he chose in the youtube player fragment.
- User can share the video.

Widget:



The widget will present the channel name that the user is currently playing its videos.

Key Considerations

How will your app handle data persistence?

- App will depend on Android Architecture component ViewModel and Room to save the videos played data.

Room database model will have Video played id, title, description and thumbnail url. When the user enters Video detail activity , the video will be saved to the DB. History fragment view model will then handle calling the DB to retrieve saved data if exist.

- The channel id will be saved in shared preference.

Describe any edge or corner cases in the UX.

- There is a back button in the video detail screen to return to home screen.
- In case that there is no network, video data will be cached for the user but without playing the video and a no network dialog will show up to ask him to reconnect.
- There is a channel id verification to check if the user has entered a correct id.
- There will be an option for the user in the settings to change the channel id if he wants to change the current channel and then he will be returned to the select channel screen.
- A loading bar/circle will be shown while loading.
- App keeps all strings in a `strings.xml` file and enables RTL layout switching on all layouts.
- App widget will display current played channel name.

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

- Picasso or Glide to handle the loading and caching of images.
- Youtube api library to load and play YouTube videos and to customize and control the video playback experience.
- 'com.jakewharton:butterknife:8.8.1' to inject views into Android components.
- 'com.squareup.retrofit2:retrofit:2.0.2' to handle API calls.
- Material design libraries for better UI views.

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

- 'com.google.android.gms:play-services-ads' to display ads at the bottom of the screen in a rectangular/ banner format.
- 'com.google.apis:google-api-services-youtube:v3' to enable using youtube Api's and youtube player fragment.

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.

App is written solely in the Java Programming Language.

Environment used:

Android Studio : 3.1.3

Libraries of version :27.1.1

Gradle : classpath 'com.android.tools.build:gradle:3.1.3'

Task 1: Project Setup

- Create Android Studio project.
- Configure libraries and needed dependencies.

Task 2: Implement Splash Screen

- Build UI for splash screen
- Handle navigation to Add Channel Id Screen.

Task 3: Implement Add Channel Id Screen

- Build Screen UI.
- Add Input validations for checking channel id.
- Save channel id in shared preference.
- Handle navigation to Home Screen.

Task 4: Home Screen

- Build Screen UI.
- Add youtube search api. (Asynctask will be used to call the search api and get its results)
- Retrieve channel and video data from api.
- Add localization setting in dropdown menu of the toolbar and handle adding strings in Strings.xml.
- Add history setting dropdown menu of the toolbar and handle navigation to history screen.
- Add change channel dropdown menu of the toolbar and handle navigation to Add Channel Id Screen.
- Implement Room database to cache video's watched data.
- Handle navigation to Video Details Screen.
- Add google ads.

Task 5: Video Details Screen

- Build Screen UI.
- Handle getting video data from home screen.
- Add youtube player fragment.
- Add Floating button to share a video.

Task 6: History Screen

- Build Screen UI.
- Handle retrieving watched videos from database.
- Handle navigating to video detail screen.

Task 7: Widget

- Build Widget UI.
- [Use intent service to handle calling it from the main activity.](#)
- Handle navigation to home screen when pressing on the widget.