Mobile Apps User Requirement Study

Version Management:

Version	Date	Editor	Changes
0.1	2017-07-16	Mohamed Arsar	First draft.
0.2	2017-07-16	Mohamed Arsar	Adding wireframe design
0.3	2017-07-21	Mohamed Arsar	Added Architecture

TABLE OF CONTENTS

1 Introduction

Video streaming has seen explosive growth over the last several years as high-speed data subscriptions soared to over 75% of households and many connected devices reached mass adoption Aflix is Video Streaming Web service for family entertainment, it is world's first ethical subscription based video service.

1.1 Objective

The goals of the project are to build mobile application that will enhance usability (by designing the user interface), security and maintainability. The main objectives of this project to build enrich mobile application on android platform. The mobile application that will attract user toward Aflix video service. The Android application which easy to use and handy to everyone

1.2 User Requirement Study

The Development will propose industry standard solution for the Apps Development. The Mobile Apps Development will be divided into 2 phases

- Android Apps Development
- IOS Apps Development

1.3 Android Version Supported

As of July 2017 here is the market share of Android Version

Android Name	Android Version	Usage Share
Marshmallow	6.0	31.8%↑
Lollipop	5.0, 5.1	30.1%↓
Kit Kat	4.4	17.1%↓
Nougat	7.0, 7.1	11.5%↑

The Applications the be developed will support all the above Android versions

2 Technologies

Android is an open source architecture that includes the Operating system, application framework, Linux kernel, middleware and application along with a set of API libraries for writing mobile applications that can give look, feel, and function of mobile handsets.

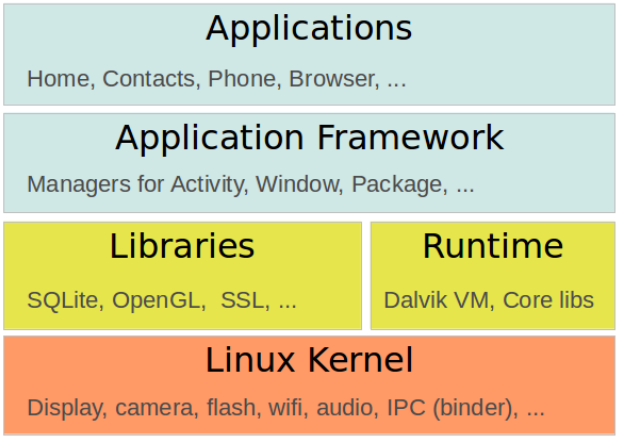


Figure 2-1 Overview of Android Architecture

2.1 Java

Java language codes are used to develop android mobile application which allows developers to write codes in the Java language. Using Google enabled Java libraries these code can control mobile devices. Android mobile Operating System also provides a flexible environment for development of Android Mobile Application as the developers can not only make use of Java IDEs but it is also possible to use Android Java Libraries

2.2 Android Studio

Android Studio is the official integrated development environment (IDE) for Android platform development. It was announced on May 16, 2013 at the Google I/O conference. Android Studio is freely available under the Apache License 2.0. Android Studio was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. Based on Jet Brains' IntelliJ IDEA software, Android Studio is designed specifically for Android development

3 Requirement Overview

3.1 Technical Architecture

The project will use the Android Studio platform to build native Android applications. The application itself will be built using Java and XML. The website for this service have built and working. Mobile App will encourage user to watch video on Aflix. The project will include server-side programming in PHP to enrich REST API of Aflix Website.

The current data fetch/sync architecture is shown below. In this existing model we simply send http request to standard web server using GET/POST method, Server receives request from client and server side script process request and prepare response and respond to user.

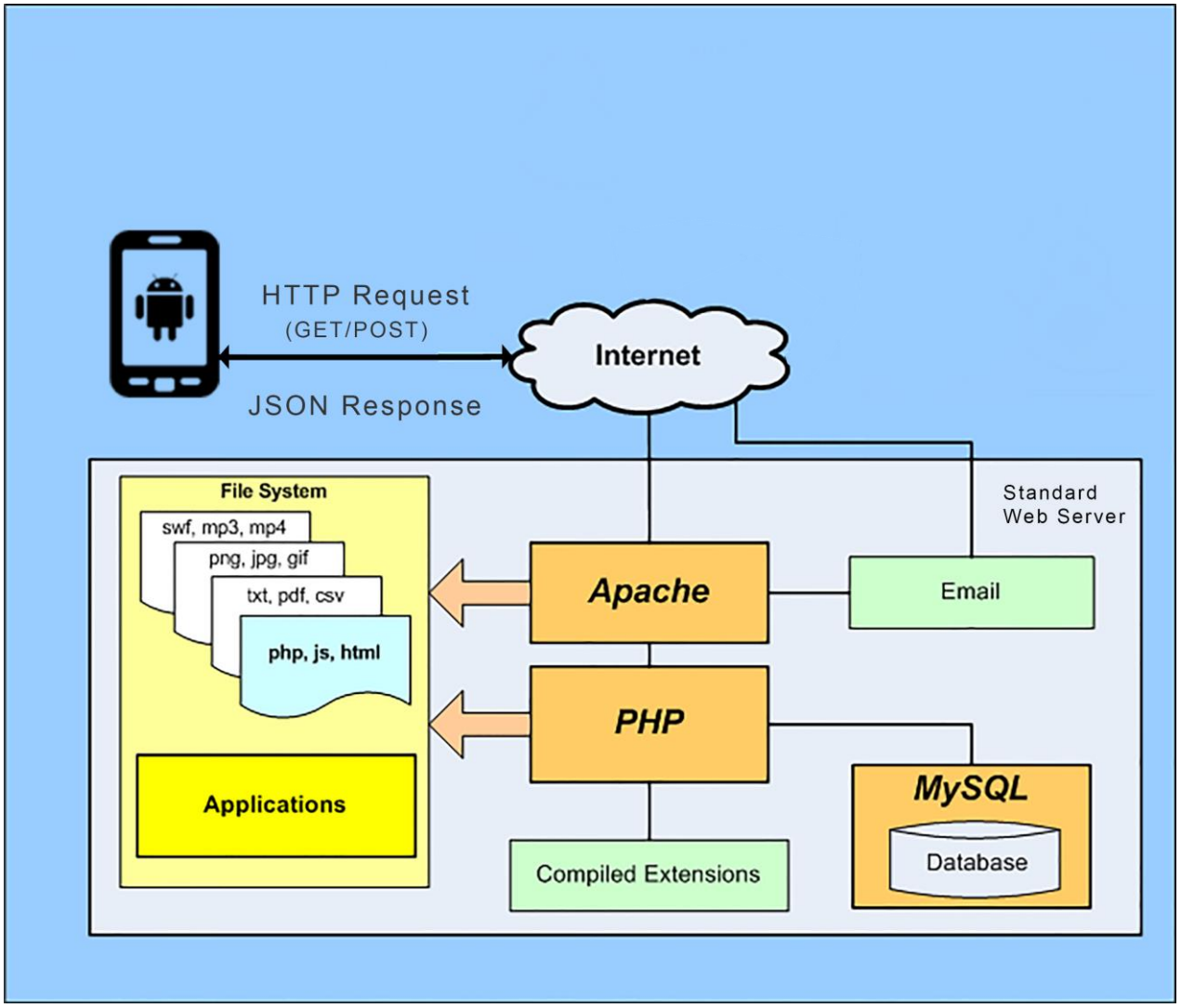


Diagram 3.1 Current data fetch/sync architecture

There are 2 types architecture proposed to the Aflix Team.

Mobile Engine	Pro	Cons
Dedicated Mobile Apps Engine	Easy Migration	Complex Development
Google Apps Engine	Powerful Engine, Easy Development	Difficult to migrate to other platform, may need some development when migrate

3.2 User Interface Flow

The below diagram shows user flow of Aflix app. When user click on screen ,System send call to java code (business logic) ,Business logic handle call and send request to Server/Cloud if required. Server/Cloud receives request and process it, may calls to database. Server/Cloud send back response .

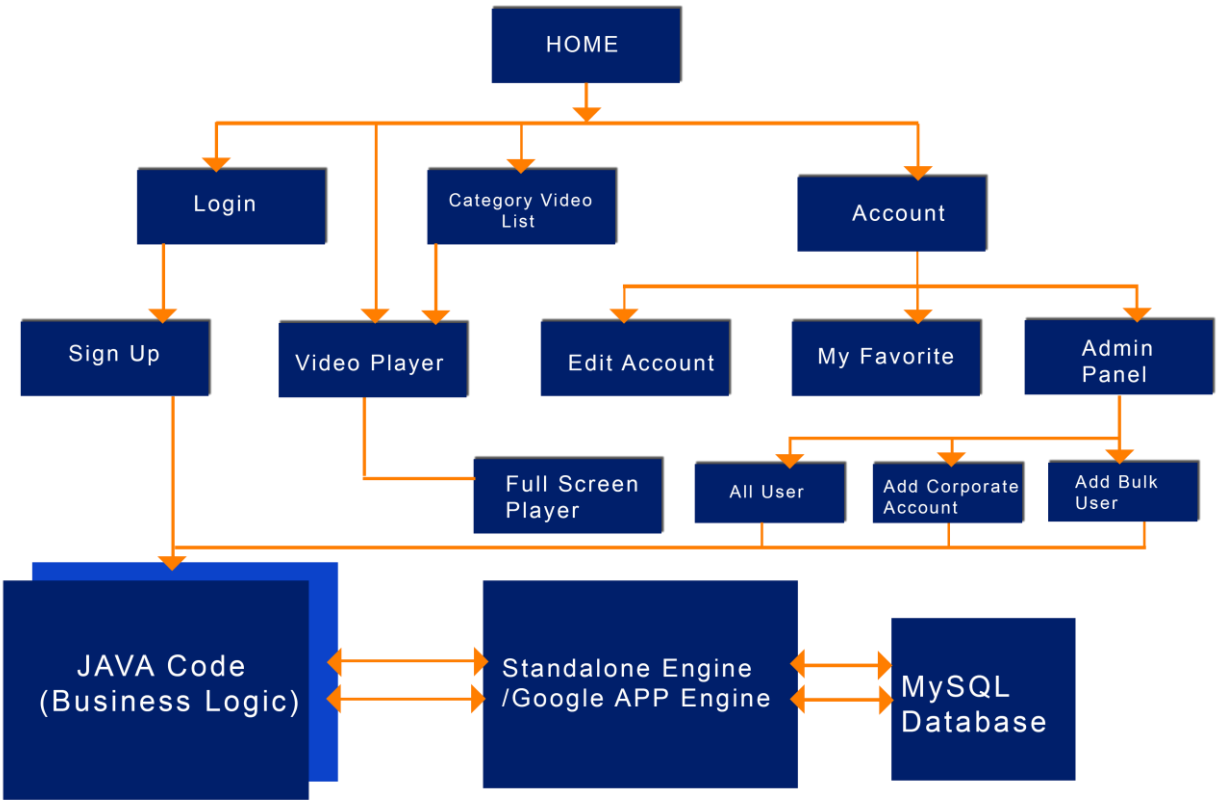


Diagram 3.2 User Flow

3.3 Option 1: Standalone Engine

The below diagram show standalone engine. In standalone engine developers have to do everything from scratch. They need to setup any server like Apache, IIS and configuring server correctly. In this architecture client can send request to server and server respond to client. The main advantage of this architecture , it can migrate easy than Google App engine.

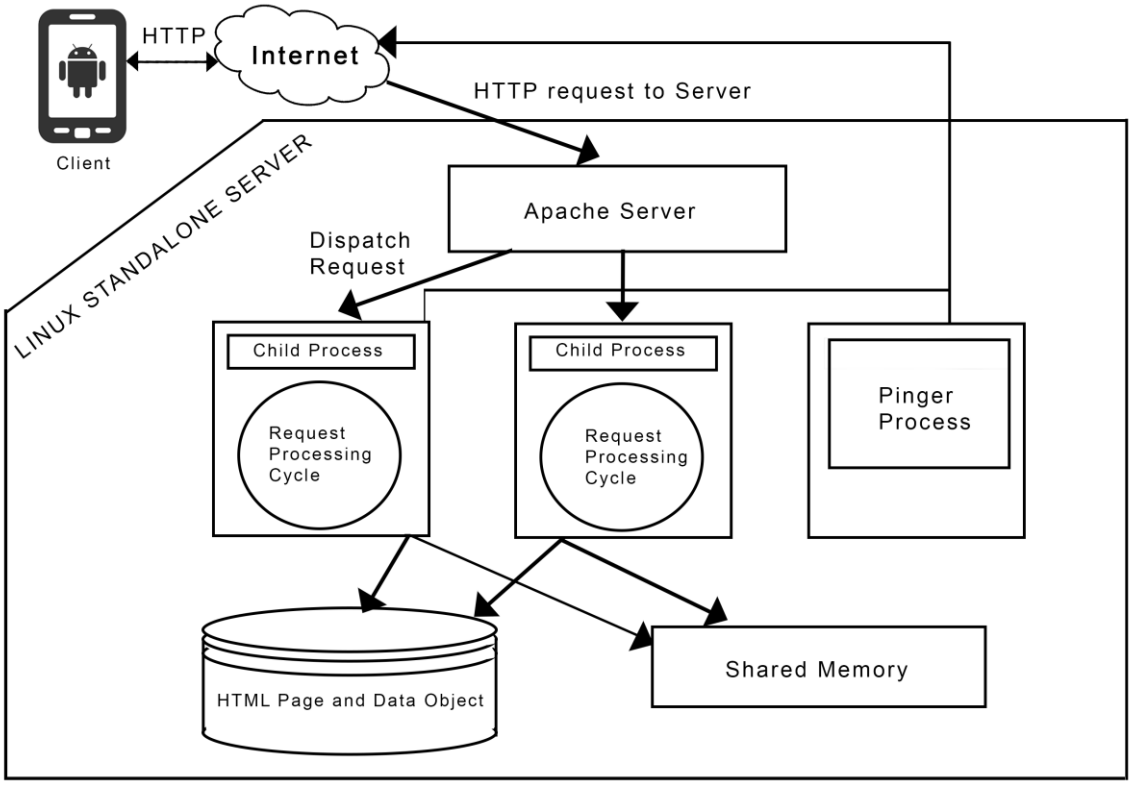


Diagram 3.3 Standalone Engine

3.4 Option 2: Google Cloud Engine

We are using Google Cloud Endpoints Frameworks which consist of tools, libraries and capabilities that allow you to generate APIs and client libraries from an App Engine application, referred to as an API backend, to simplify client access to data from other applications. Endpoints Frameworks make it easier to create a web backend on App Engine for web clients and mobile clients such as Android or Apple's iOS.

For mobile developers, Endpoints Frameworks provide a simple way to develop a shared web backend and also provides critical infrastructures, such as OAuth 2.0 authentication, eliminating a great deal of work that would otherwise be needed. Furthermore, because the API backend is an App Engine app, the mobile developer can use all of the services and features available in App Engine, such as Datastore, Google Cloud Storage, Mail, Url Fetch, Task Queues, and so forth. And finally, by using App Engine for the backend, developers are freed from system admin work, load balancing, scaling, and server maintenance.

It is possible to create mobile clients for App Engine backends without Endpoints Frameworks. However, using Endpoints Frameworks makes this process easier because it frees you from having to write wrappers to handle communication with App Engine. The client libraries generated by Endpoints allow you to simply make direct API calls.

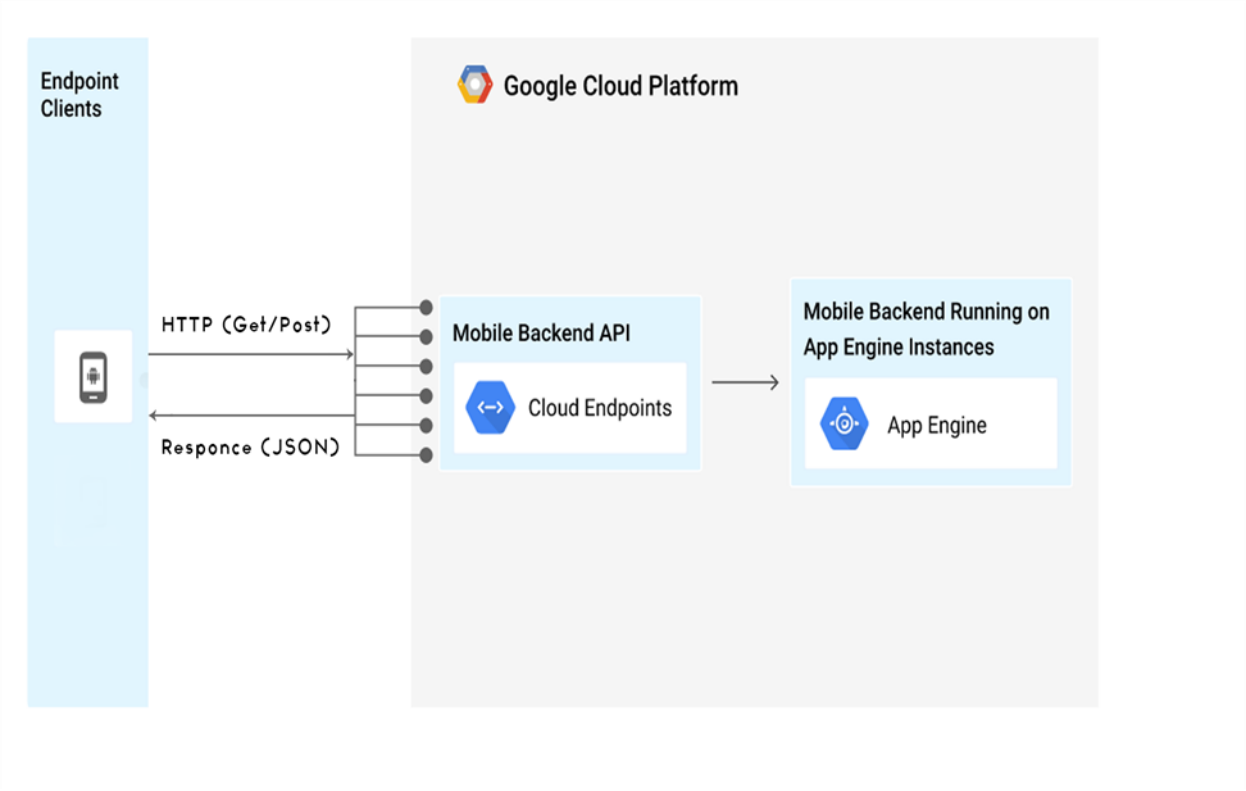


Figure 3-1 : System Architecture

4 Implementation

4.1 Prototyping

Prototyping is the important part of the app which decides how the app look like after implementation. Photoshop CS6 has been used for prototyping of the Android App UI design.

4.2 UI design in XML View of the app –Implementation of View

Material design will be been used with the latest ripple effect and flat UI with fragments in the app. The fragments gives more control of the element instead of simple activity alone.

4.3 Testing of UI in different versions of android using Genymotion

Maximum backwards compatibility of the app been set to KitKat (Android 4.4) and maximum up to Nougat (Android 7.1) and recently has been updated to

4.4 Implementation of Singleton class Of Volley Instantiation

Volley library has been used for transferring the data from the server and to the server asynchronously so that the app does not hang up working network task in the main thread. The singleton class keeps track of all the requests in the form of queue.

4.5 Parsing JSON data from the server – Implementation of Models

The JSON data from each call from the server is first parsed to store in the Saved Instances which is kind of local cache for the data to increase the performance of the app with long UI list.

4.6 Implementation of the Controller – The Adapter Class

The adapter class in the app does the exact work as the controller does. It transfers the request from the fragment or activity to the volley and then updates back the fragment using asynchronous calls from the model.

4.7 Handling the Images – Picasso Library

All the images in the app are loaded asynchronously using Picasso library. The library uses local caching to persist the large image sets in the app itself.

4.8 Using Git for Version Control

All the code revisions are saved in the form of version so that in case of any disaster like accidently overriding the working code.

5 Wireframe Designs

5.1 Splash Screen

Figure 5-1: Splash Screen

5.2 Login Screen

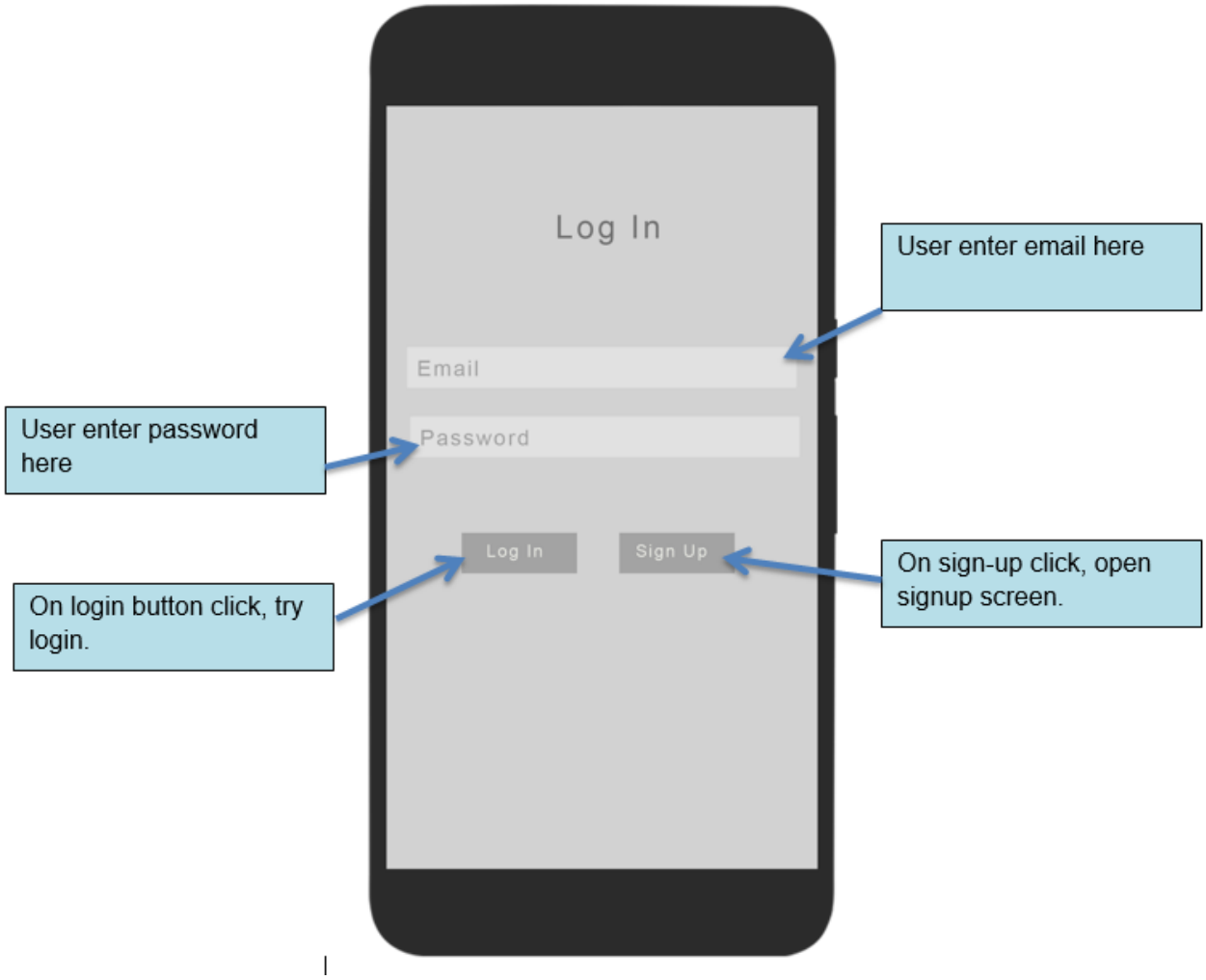


Figure 5-2 : Login Screen

5.3 Signup Screen

Figure 5-3 : Signup Screen

5.4 Permission to access contact - example

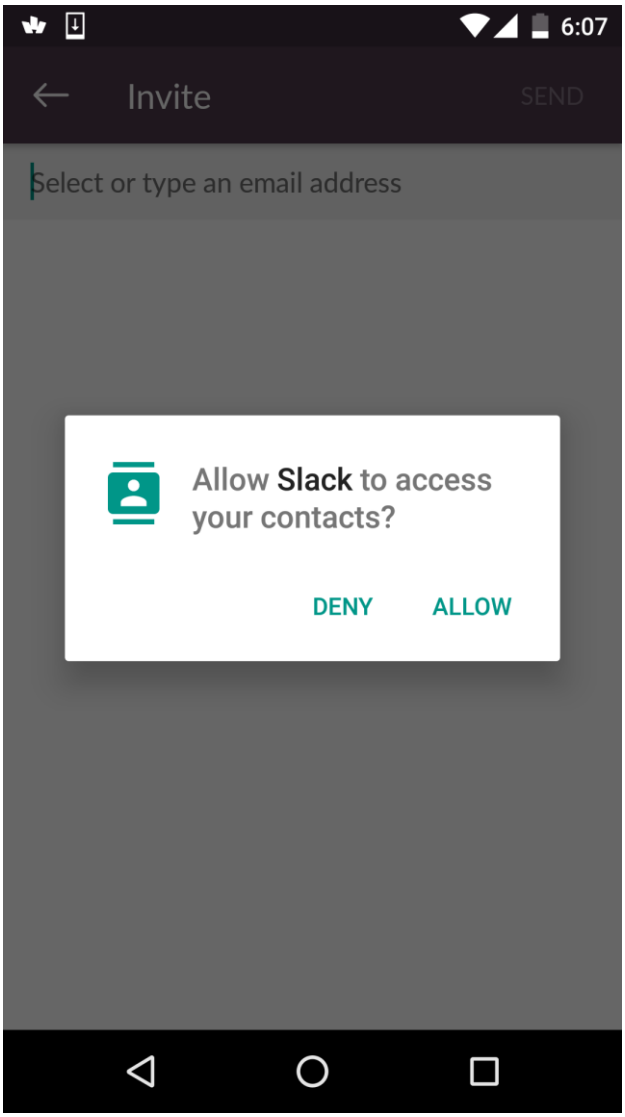


Figure 5-4 : Contact Permission

5.5 Home Screen

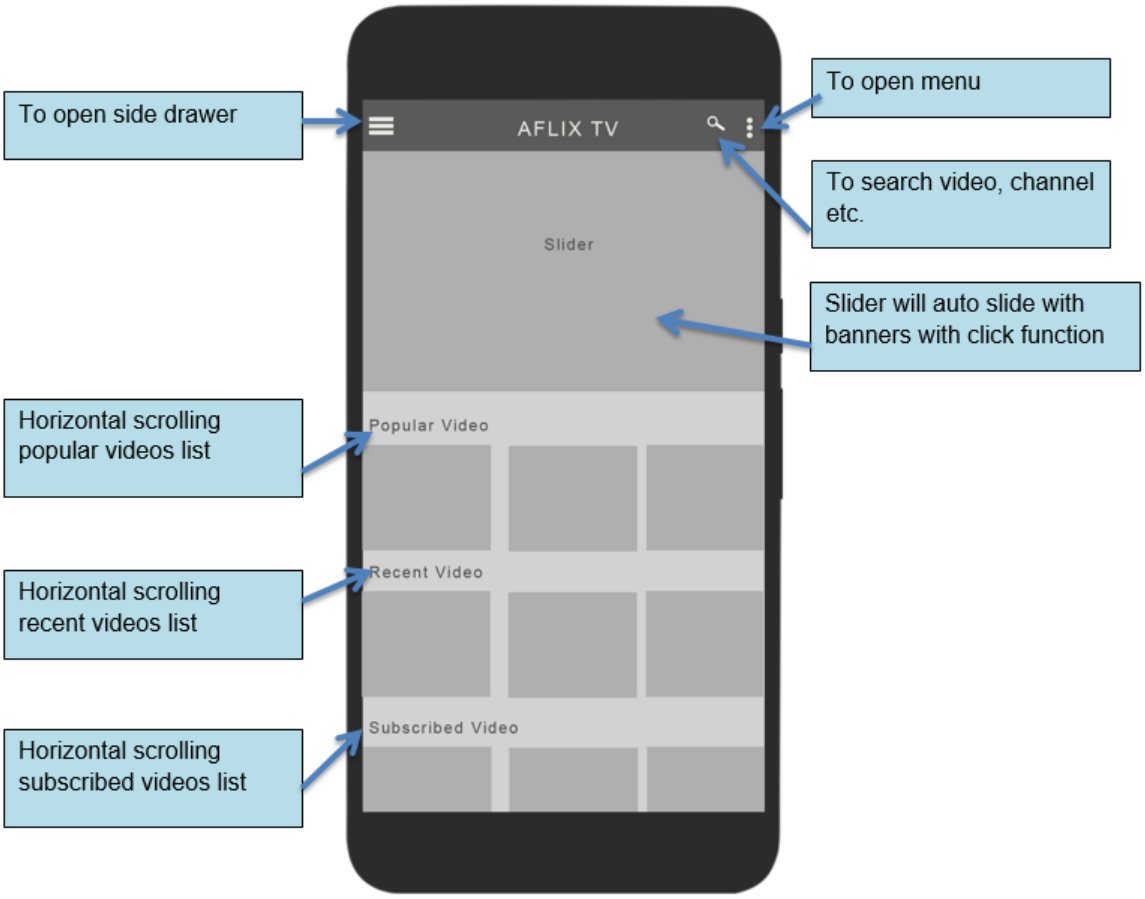


Figure 5-5 : Home Screen

5.6 Side Drawer Screen

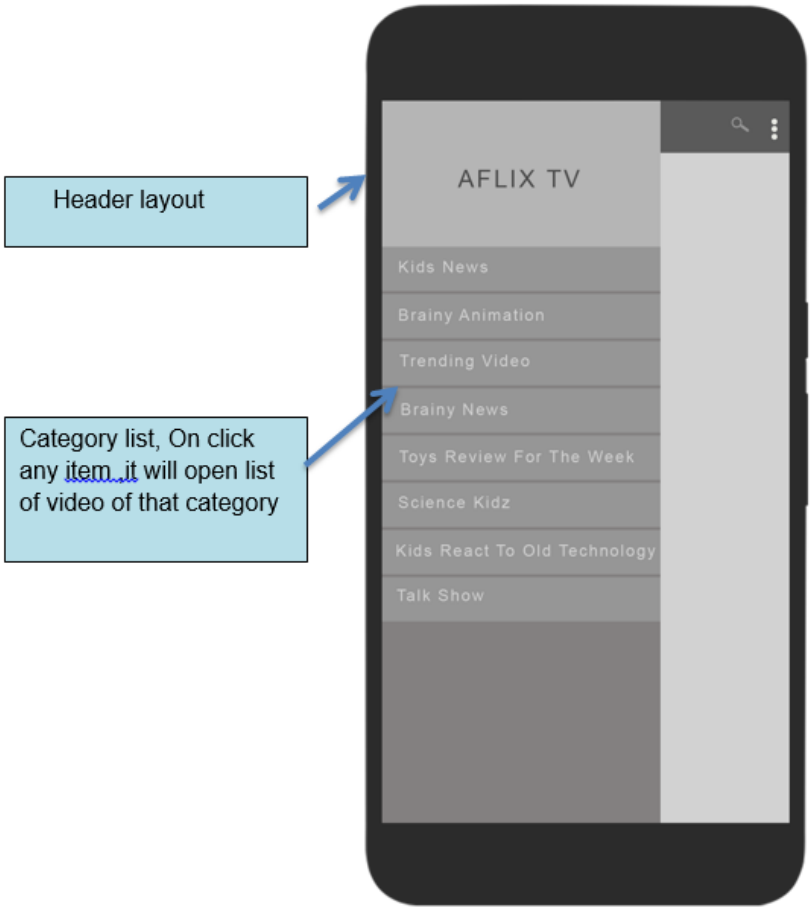


Figure 5-6 : Side Drawer Screen

5.7 Category Video List Screen

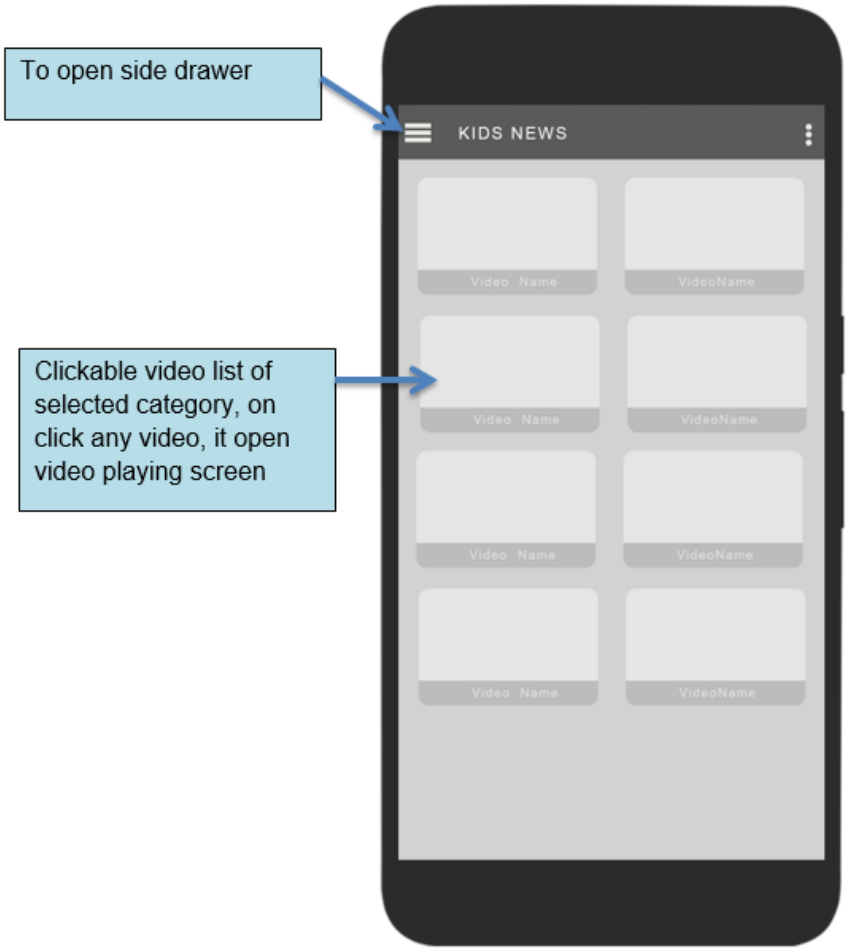


Figure 5-7 : Category Video List Screen

5.8 Video Player Screen

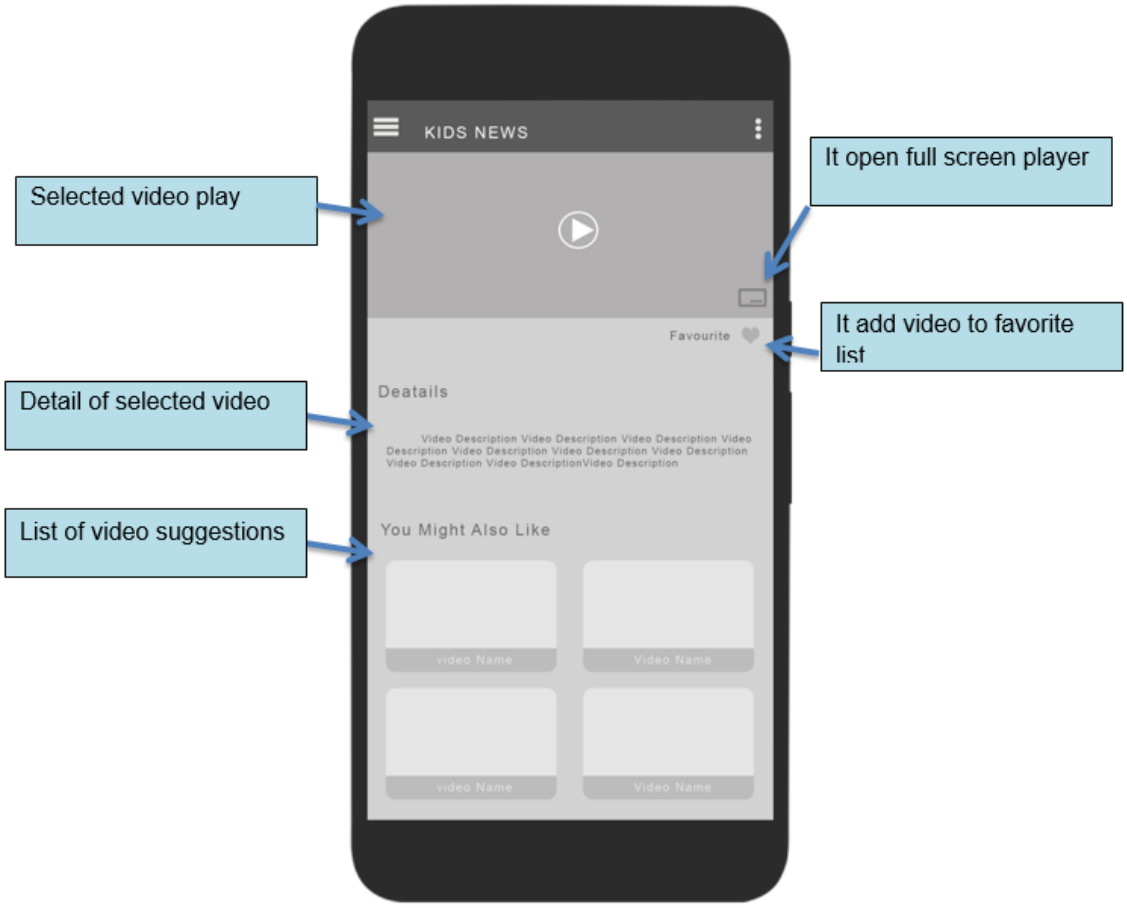


Figure 5-8 : Video Player Screen

5.9 Full Screen Video Player

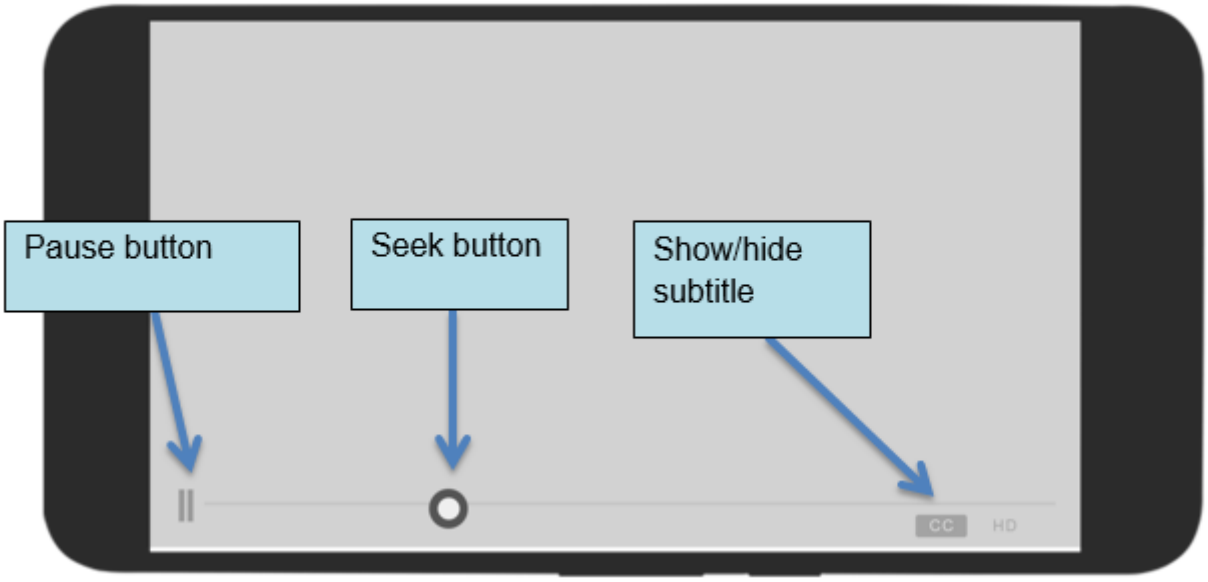


Figure 5-9 : Full Screen Video Player

5.10 Contributor Account

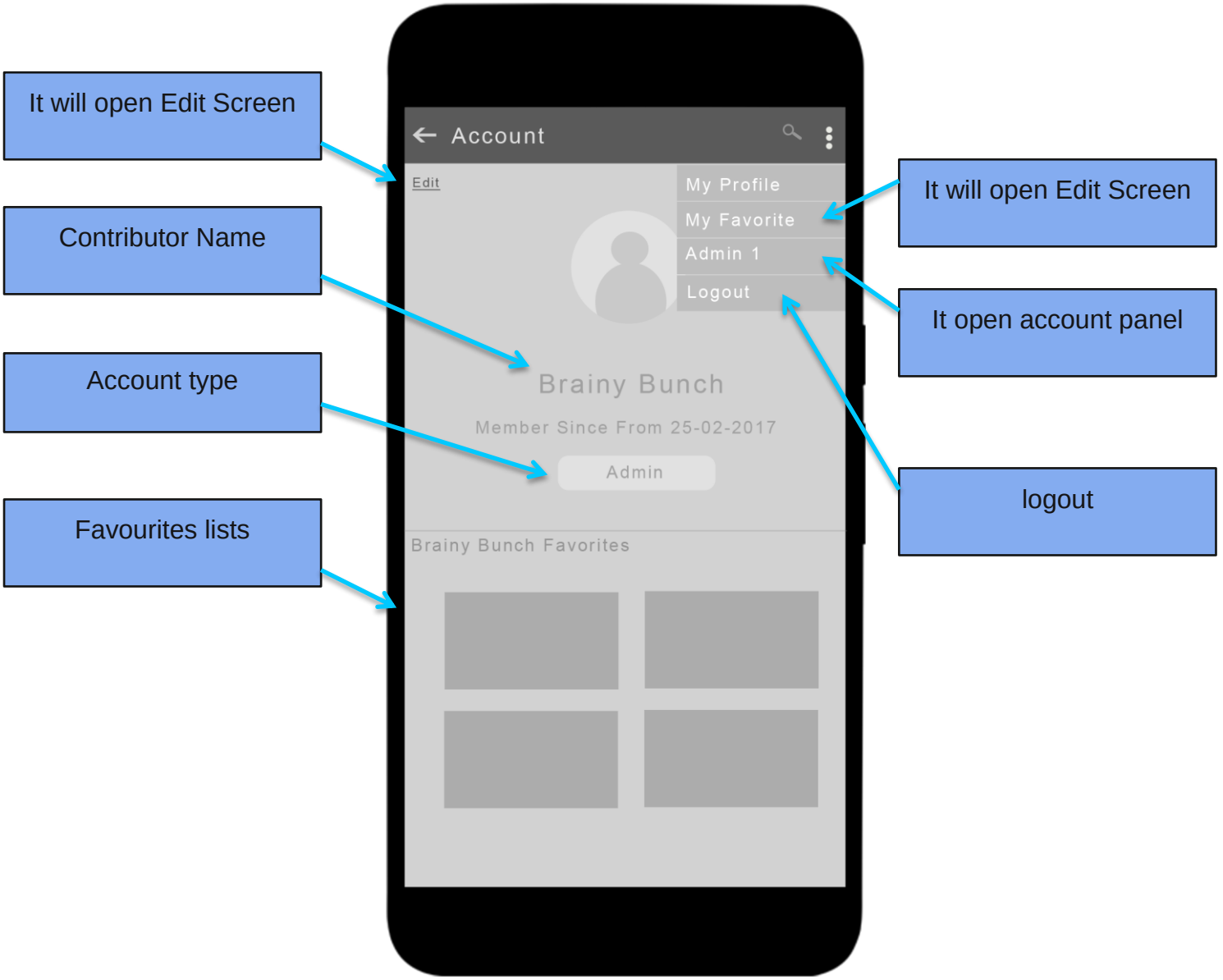


Figure 5-10: Contributor Account

5.11 Contributor Dashboard

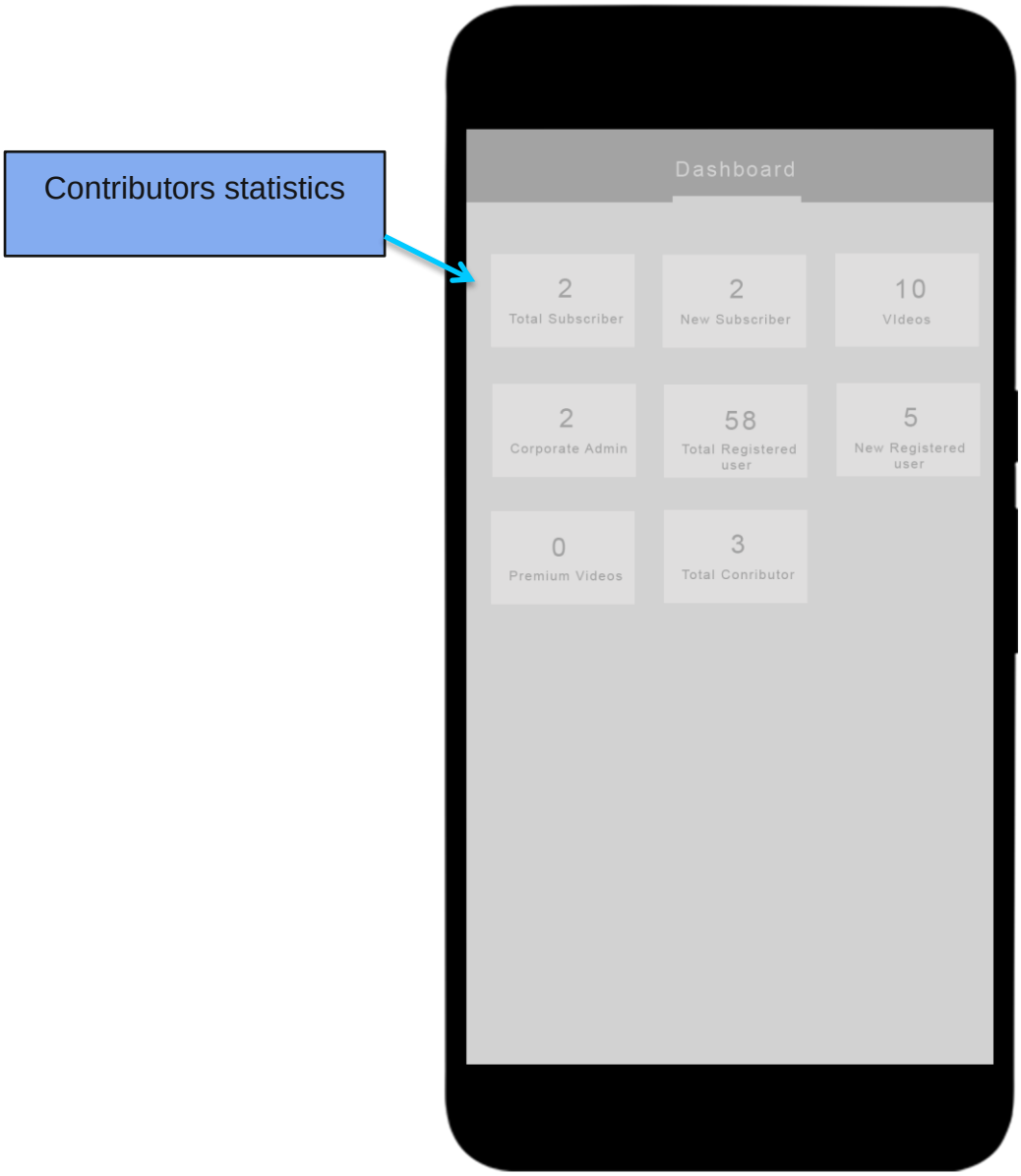


Figure 5-11: dashboard

5.12 Edit Account

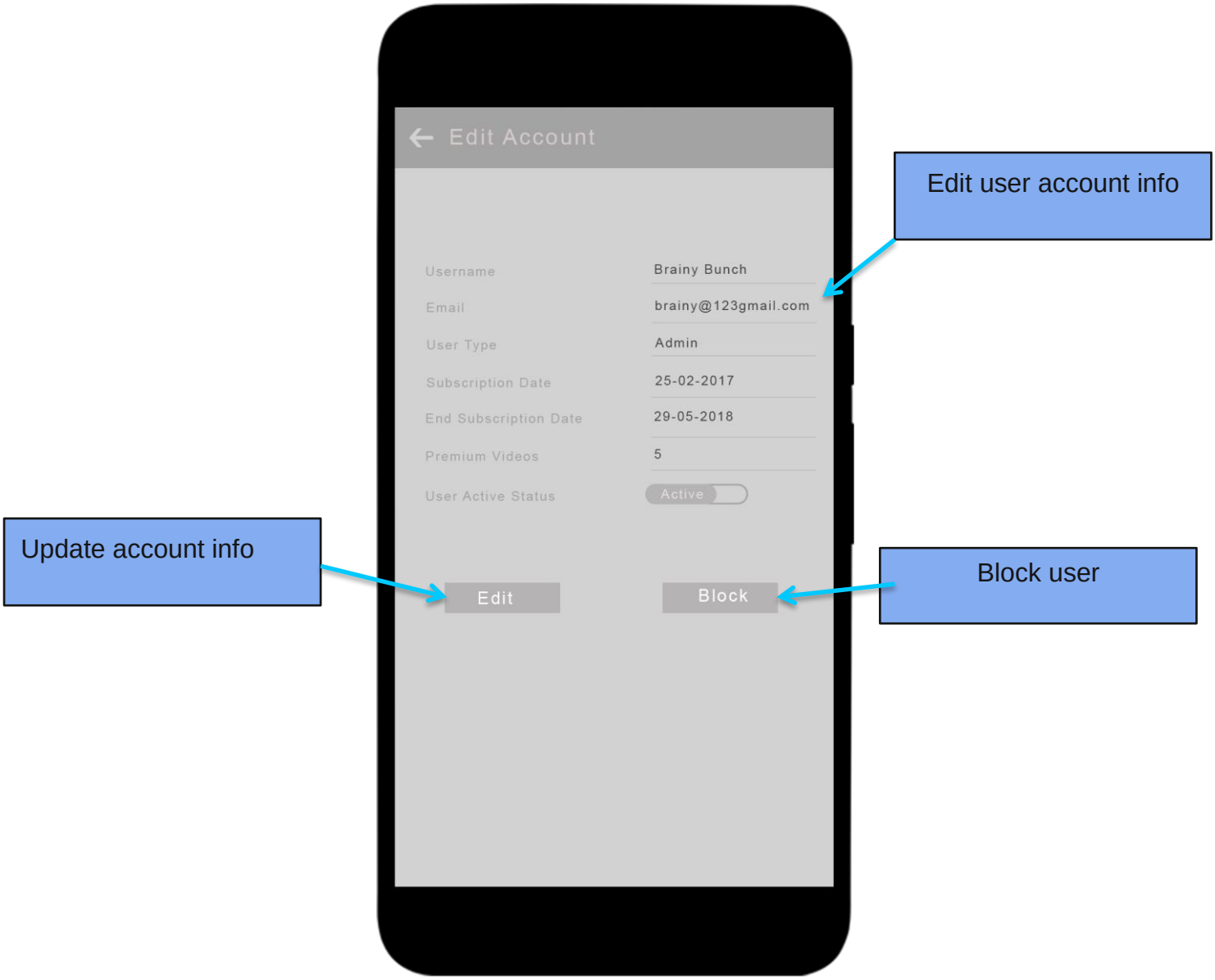
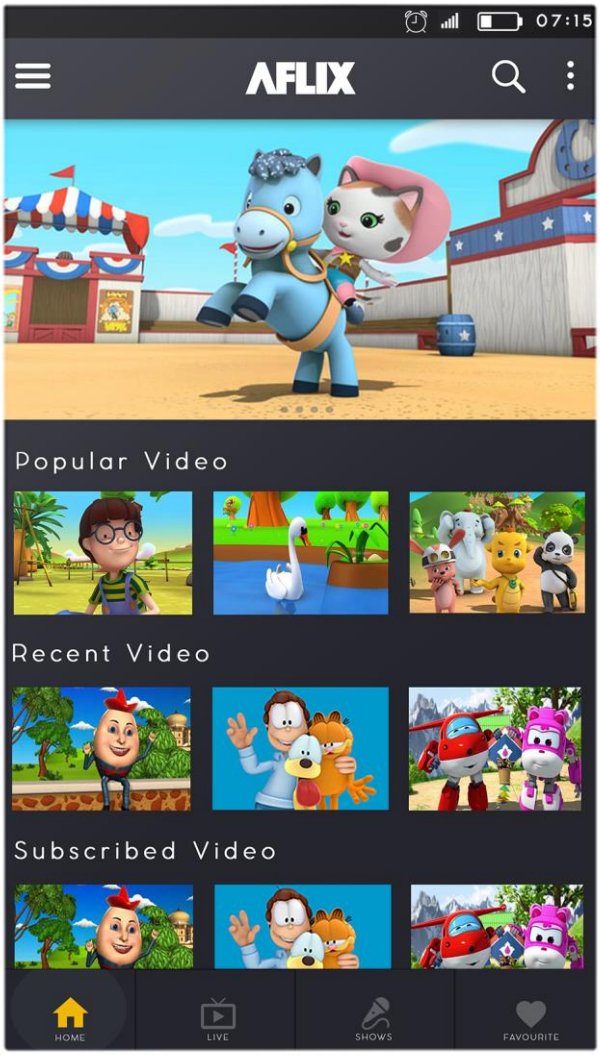


Figure 5-15: Edit account

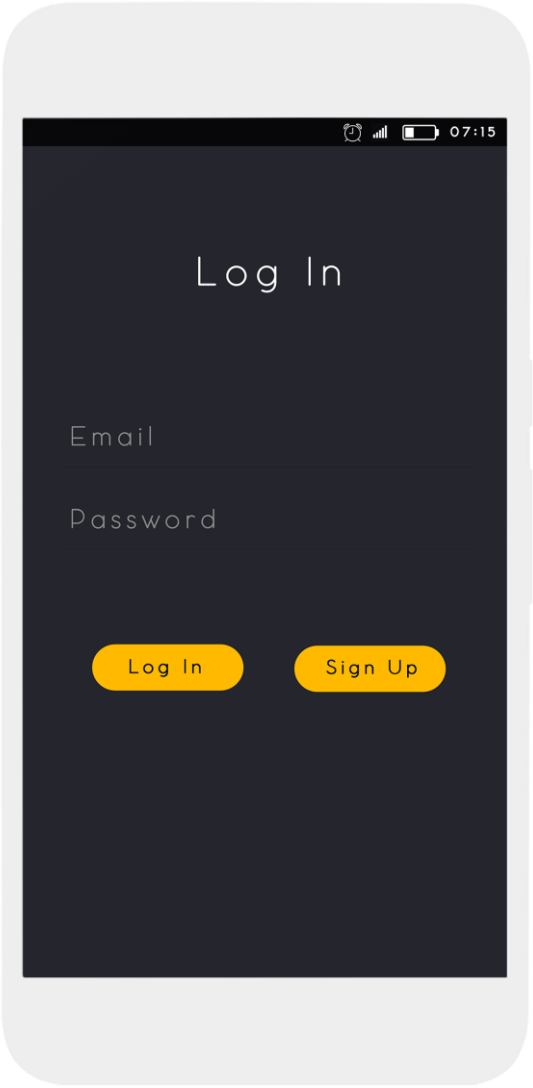
6 Proposed UI : Material Design

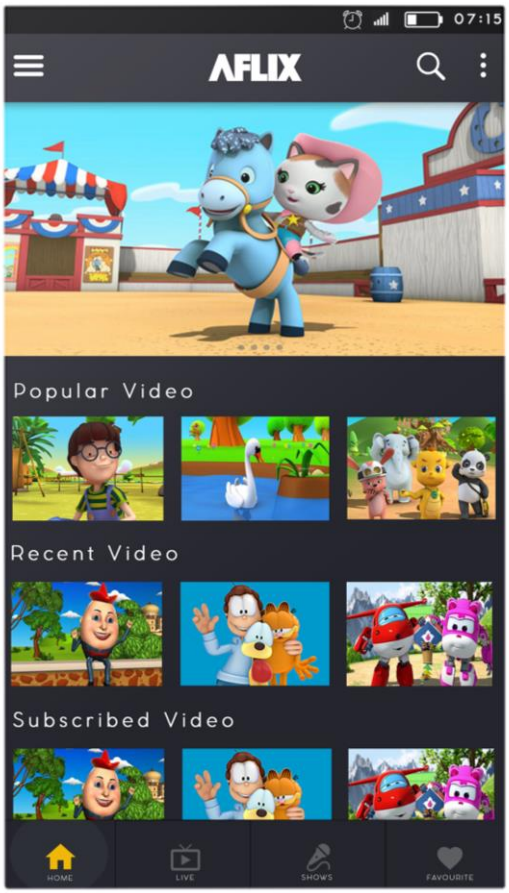
Material Design is a design language created for Google Android's new OS which was announced in Summer 2014. Material design is a comprehensive guide for visual, motion, and interaction design across platforms and devices.

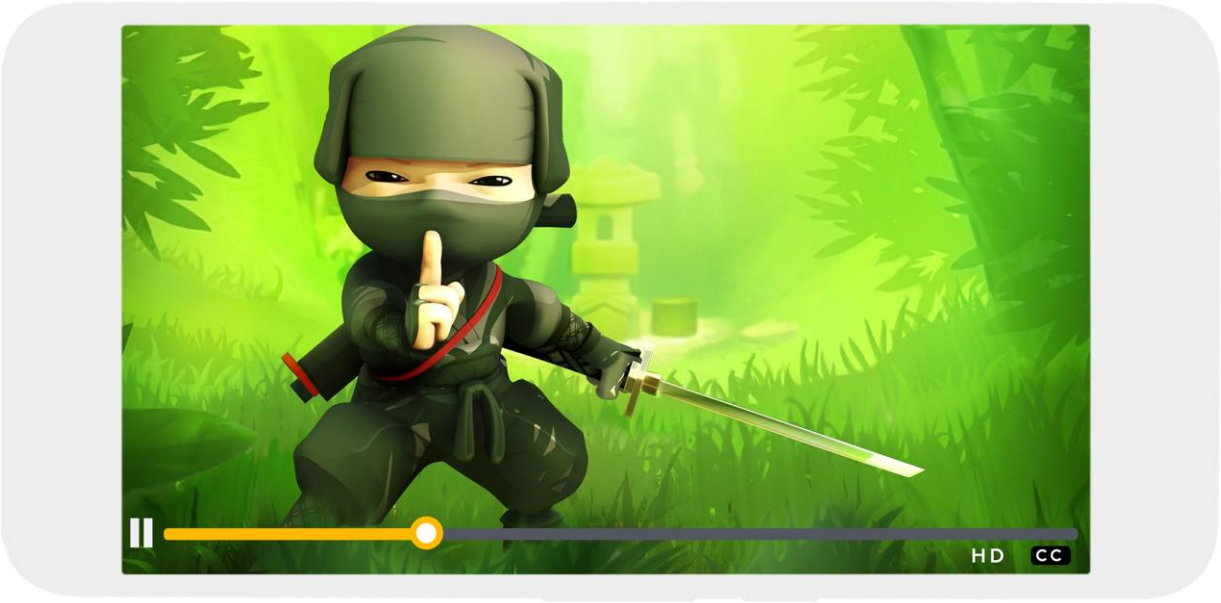
6.1 Option 1

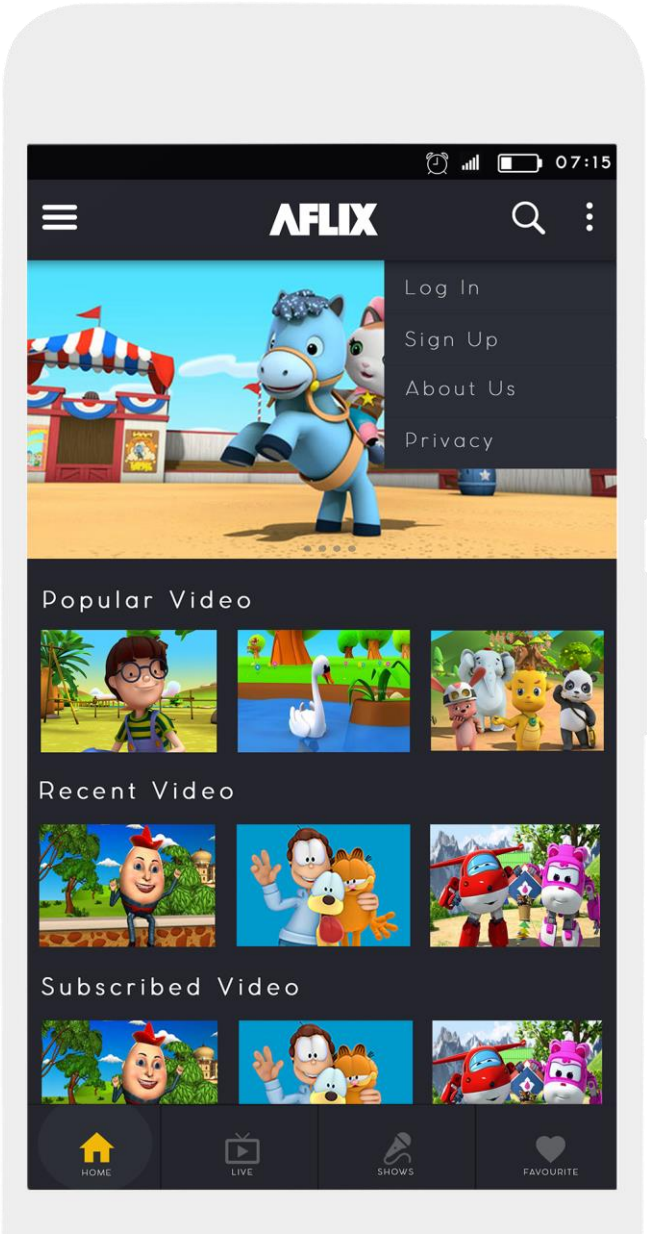


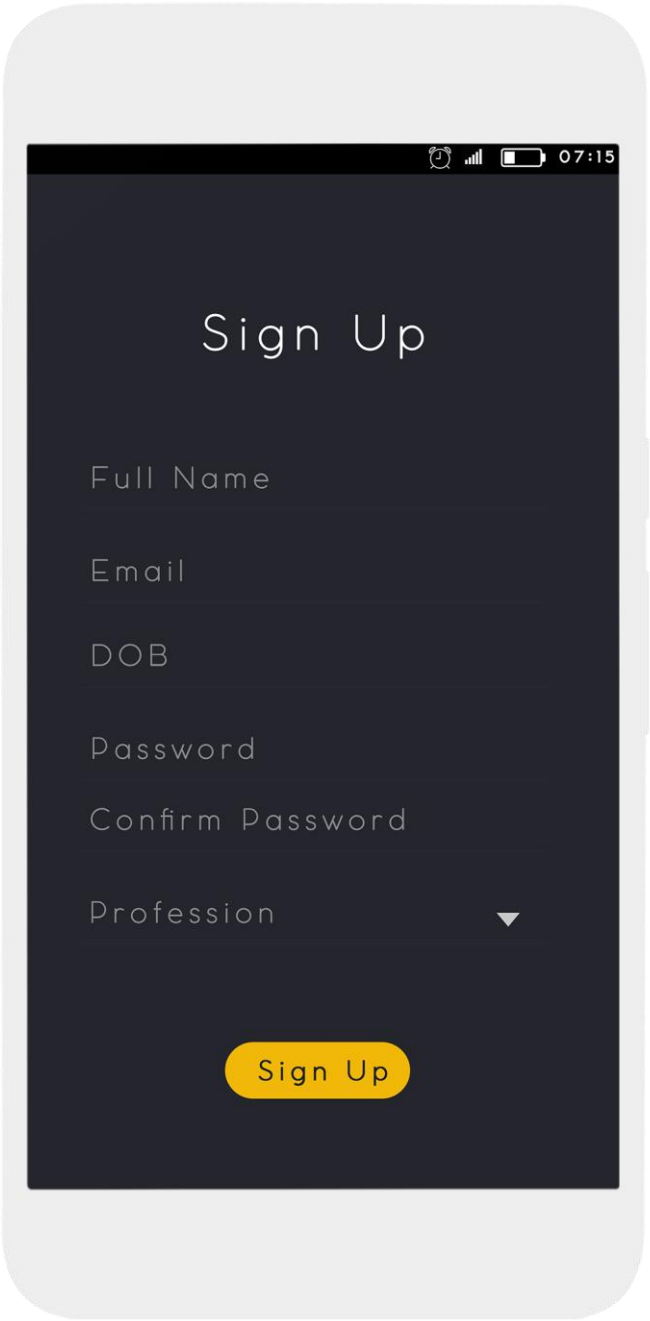
7 Individual Page Design

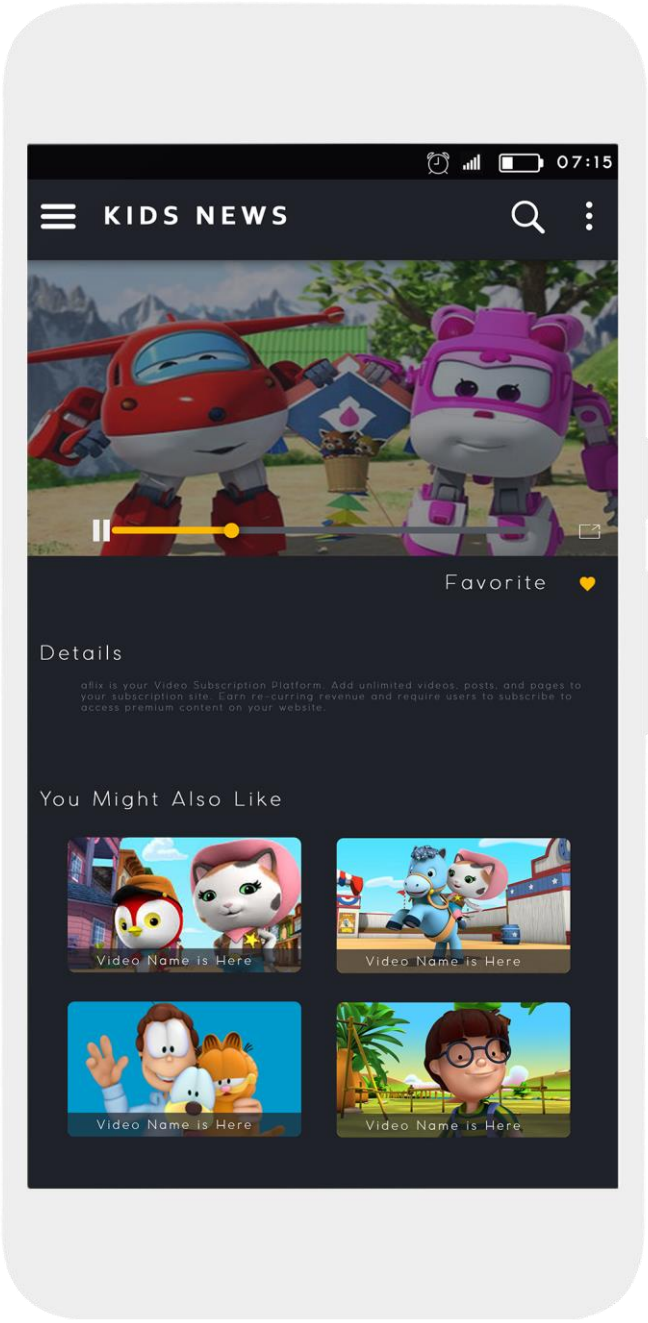


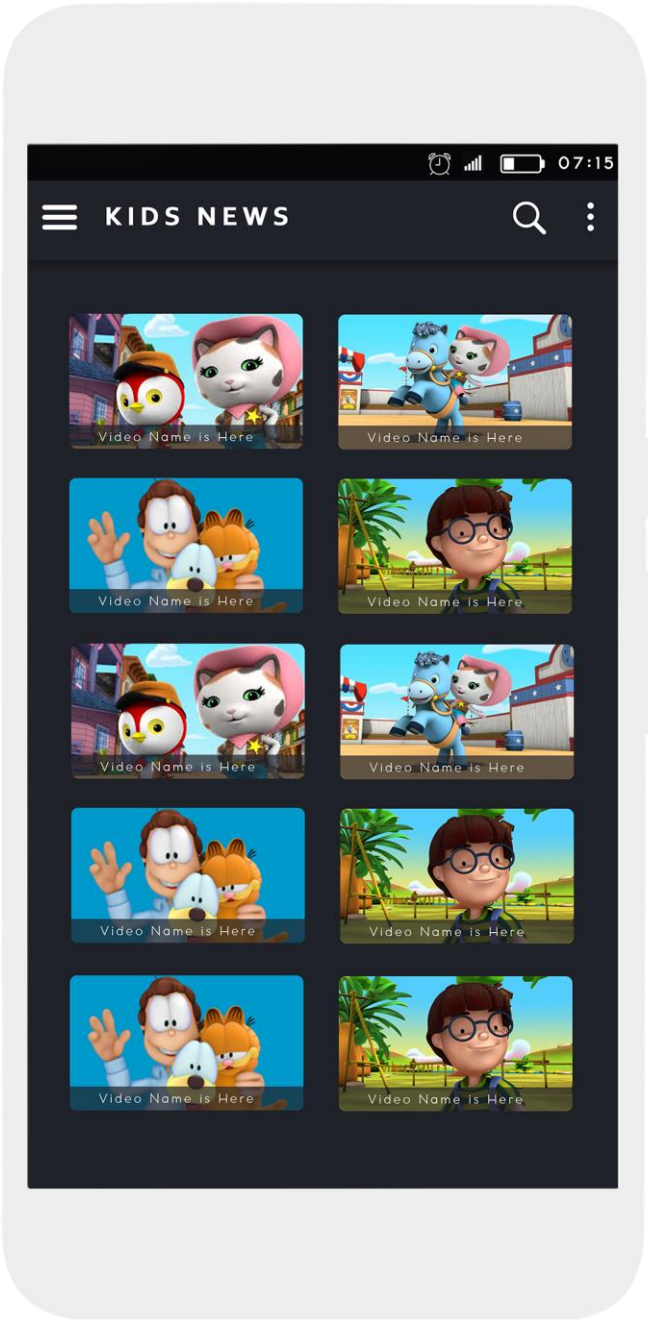

07:15
AFLIX
Popular Video
Recent Video
Subscribed Video
HOME
LIVE
SHOWS
FAVOURITE

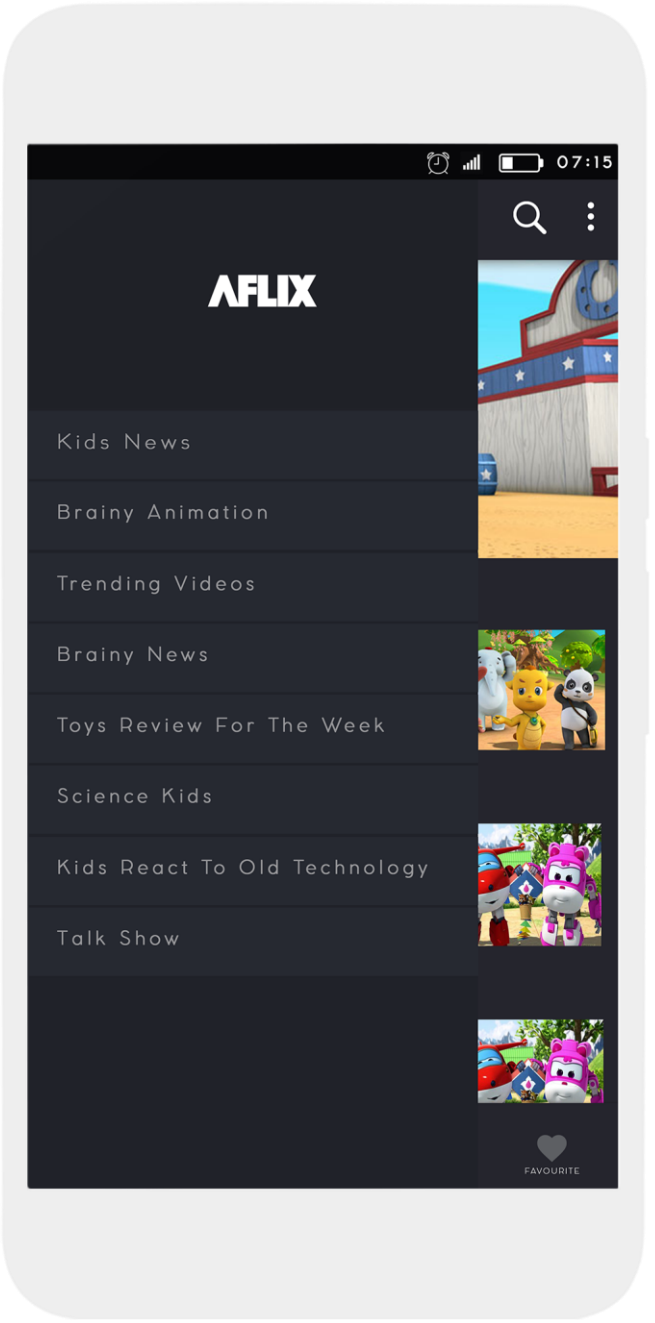

HD
CC


07:15
AFLIX
Log In
Sign Up
About Us
Privacy
Popular Video
Recent Video
Subscribed Video
HOME
LIVE
SHOWS
FAVOURITE


07:15
Sign Up
Full Name
Email
DOB
Password
Confirm Password
Profession
Sign Up


07:15
KIDS NEWS
Favorite
Details
aflix is your Video Subscription Platform. Add unlimited videos, posts, and pages to your subscription site. Earn re-curring revenue and require users to subscribe to access premium content on your website.
You Might Also Like
Video Name is Here
Video Name is Here
Video Name is Here
Video Name is Here


07:15
KIDS NEWS
Video Name is Here
Video Name is Here
Video Name is Here
Video Name is Here
Video Name is Here
Video Name is Here
Video Name is Here
Video Name is Here
Video Name is Here
Video Name is Here


07:15
AFLIX
Kids News
Brainy Animation
Trending Videos
Brainy News
Toys Review For The Week
Science Kids
Kids React To Old Technology
Talk Show
FAVOURITE