Mobile App Engineering Final Project Report

App name: MayShow
Team name: Mayday
Team member: Xuewei Li, Shi Wang, Junjie Feng, Ruotian Zhang

Contributors

Xuewei Li	Sign in & sign up activity related to create a new user and login by username, by implementing KickflipApiClient. Basic UI structure building. Testing transmitting quality. Write report, slide, readme.
Shi Wang	Finding SDK. CameraActivity related to start broadcasting by implementing Kickflip SDK. StreamListFragment related to receive live videostream by implementing Vitamio SDK. Write report, slide, readme.
Junjie Feng	Managing user's information by implementing Firebase. MyProfile fragment related to listview adapter by implementing Kickflip SDK. Testing and debugging the Searching module and the LogIn and SignUp module. Adding indication to code. Write report, slide, readme.
Ruotian Zhang	StreamListFragment related to listview adapter and sort stream by keyword and username by implementing Kickflip SDK. Testing and debugging the LogIn and SignUp module. Write report, slide, readme.

App Description

- ### Library we use

Basically, MayShow is based on camera API, Kickflip SDK and Vitamio SDK. Firebase was included to help us manage the user's information. What we want to achieve is establishing a connection between users, allowing them to start a live video stream and watch others' recording/recorded streams. KickflipApiClient is the core of Kickflip SDK. Once a user enter the App, a KickflipApiClient class instance was created automatically according to the username and password which the user typing in. Then we can use startStream method to stream a m3u8 playlist(HLS). It's a pity that KickflipApiClient class does not have method related to stream parsing, so we still need a tool to process the stream for video displaying. For that reason, we find a strong tool named Vitamio SDK. It can process different protocol stream including m3u8 playlists, RTSP streams and MMS(Microsoft Media Stream), which the m3u8 is exactly what we need.

Originally, our goal is sharing our phone screen to others and we learned a lot of projects already set before online and found teamviewer API. However, teamviewer is a distant-assist application and is aiming for helping user fix their phones. By this way, it is not allowed to watch by multi-people. In another word, it is an one-on-one technique and does not satisfy our requirement. We need to utilize a one-to-many technology stream communication. Later, one group member found a way to capture the screen and hoped we could combine it with Kickflip SDK by sending the view of the screen instead of the camera view. The problem is that getting video source module and coding module are highly complicate. It would be difficult for us to rewrite them since we have scarce knowledge about certain transmitting protocol(m3u8). For all the reasons we mentioned above, we decided to do this instead.

- **Main functionalities**

First, we let users register an account by username and password. In the next time,the user could directly login to the application using certain account which are already stored both in our Firebase and Kickflip database. By this function, the user who has already set the account up in our database, can be able to enter his/her account and broadcast a new live stream as many times as wish. Once the user logged into the application, they could check out the list of live streams or videos captured by ALL THE USERS in the "Square" fragment. If they click on any video or live stream, it could be played by entering MediaActivity class. Users could also search the live streams or videos by typing the specific username. Users could also check out the stream list anytime by clicking the "Square" button at bottom of the screen. The "Square" button could also work as a refresh button, which means users could refresh the list by clicking it. In this module, our application offer a platform for showing the list of streams based on the Kickflip database.

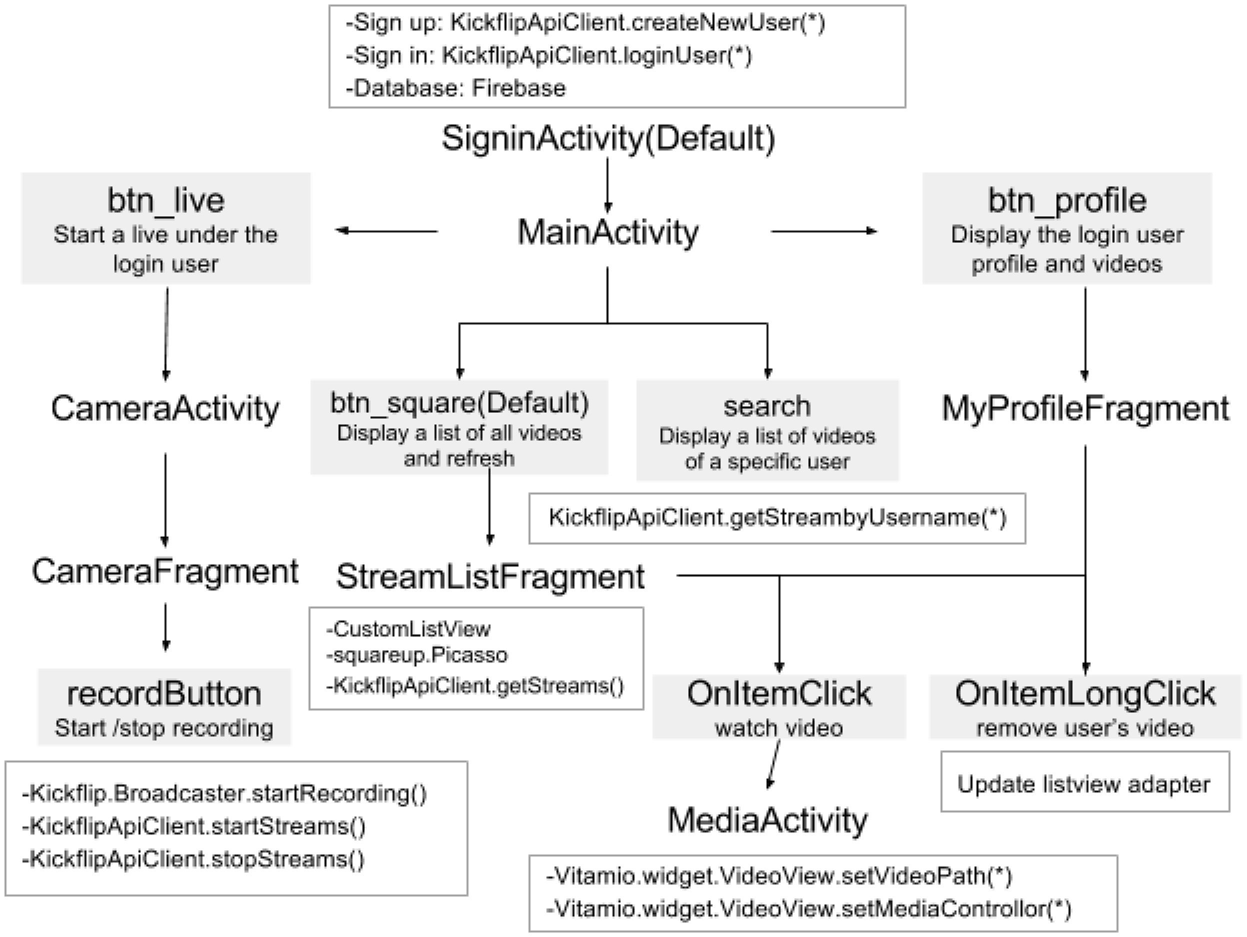
Second, when users want to start a live stream with his/her camera, they could click the "Live" button. Once they click it, application will enter CameraActivity, then the user could start or stop broadcasting by clicking the red button on the right side of the screen which looks like the recording button in the system camera application. In this module, our application implements the function of live stream broadcasting from camera recording.

Finally, we also provide a Myprofile fragment to show the user information and the video captured by himself/herself.It They could delete the the previous video by long click on it. In this part, the logged in user can be able to view the stream list owned by himself and also delete the video as he want. Deletion function is also implemented in the "Square" module, while the user is only authorized to delete the streams that are belong to himself.

For the listview adapter, we write our custom stream adapter based on the kickflip SDK. The element of the list is a structure called "Stream". The Stream structure contains the whole information about a video or live stream, like the URL of it and the owner name. By using the functions in KickflipApiClient, we could update the list of stream from the the online Kickfilp database.

App Architecture

The whole project has 4 activities (SigninActivity, MainActivity, CameraActivity and MediaActivity) and 3 fragments (StreamListFragment, CameraFragment and MyProfileFragment) . The whole structure is shown below.



-Sign up: KickflipApiClient.createNewUser(*)
-Sign in: KickflipApiClient.loginUser(*)
-Database: Firebase

SigninActivity(Default)

MainActivity

btn_live
Start a live under the login user

CameraActivity

CameraFragment

recordButton
Start /stop recording

-Kickflip.Broadcaster.startRecording()
-KickflipApiClient.startStreams()
-KickflipApiClient.stopStreams()

btn_square(Default)
Display a list of all videos and refresh

StreamListFragment

-CustomListView
-squareup.Picasso
-KickflipApiClient.getStreams()

search
Display a list of videos of a specific user

KickflipApiClient.getStreambyUsername(*)

btn_profile
Display the login user profile and videos

MyProfileFragment

OnItemClick
watch video

MediaActivity

-Vitamio.widget.VideoView.setVideoPath(*)
-Vitamio.widget.VideoView.setMediaControllor(*)

OnItemLongClick
remove user's video

Update listview adapter