**Report**

**Application Description**

Application name: CloudPlay

Descriptions: CloudPlay is a Youtube API based music streaming app aimed to be simple and easy to use with minimalistic design which found to be lacking in popular apps such as Spotify or SoundCloud. The app contain minimal, but essential features which focus on allowing user to be able to listen to online music with relative ease and speed without having to be distracted or overwhelmed by multiple features or obvious advertisements system.

**API and library used**

API used:

* Youtube data API
* Youtube-dl streaming server
* Google Sign in API and Google Play services

Library used:

* Youtube API library
* Google Sign in and credentials library
* Easy Permissions
* GSON and GSON convertors
* Card view
* Recycler view
* Picasso image caching library
* P1Droid gif drawable
* Palette library
* Retrofit
* SlidingUpPanel library
* Pymongo
* Flask

Database used:

* MongoDB

Cloud services used:

* Mlab
* Uptimer Robot
* Heroku
* Github

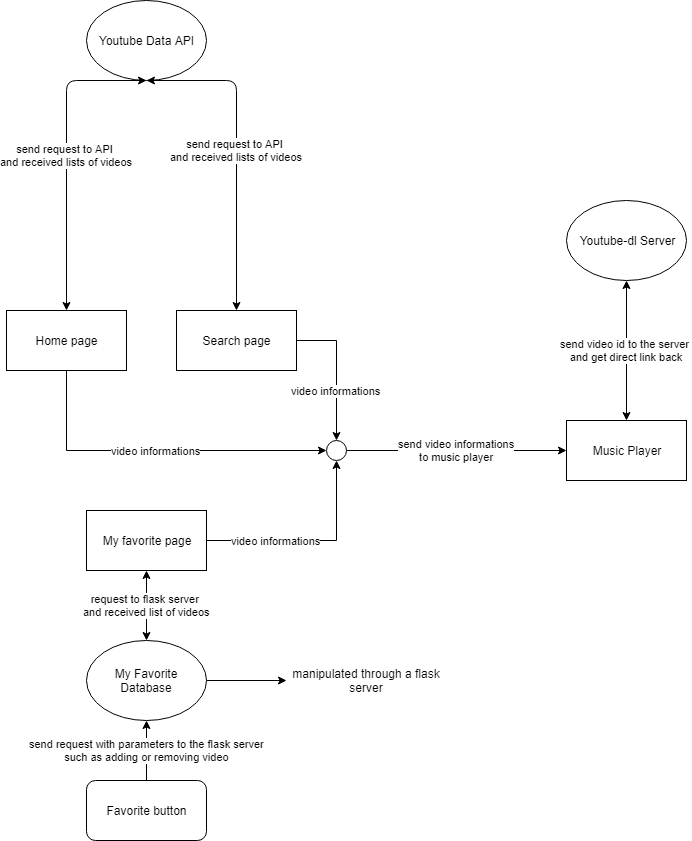
**Obstacles**

1. Early on, we decided to use the sound cloud API for the music, but later found out that they have shut down the registration for the API.

2.We then change to use Spotify API, while the API works, its limitation such as not being able to choose specific songs unless you are a premium members along with little to no examples or tutorials on the internet makes us drop the API and have to change some major codes.

3. While we can get most video information from youtube Data API, we can’t get the direct link to the video itself. At first we thought of using Youtube official API, but drop it because we can’t get full control of the video and have to use Youtube built in player to play, which is not ideal. We settle with using a 3rd party custom Heroku server based on the Youtube-dl library, which allow us to get direct link to the video file and allow us to have full control over the video itself.

**General program flow (picture)**



**Work Distributions between group members**

1. Tantai Layluxsiri (6031756621)

* Front-end of the application.
* Communications between the application and various servers and APIs.
* Manage the general flows of the applications, along with the work distribution of the application among group members.
* Project file management on Github.

1. Thananont Chevaphatrakul (6031757221)

* Creation of the music player functionality of the application.
* Creation of music player services, which allows the music to be played independent from the application lifecycle.

1. Sirikarn Yodvanich (6031873321) and Praewpun cheevamongkol (6031810821)

* Creation of the flask My favorite servers.
* Creation of cloud-based mLab database.
* Back-end managements.