Capstone Design (2)

Final Report

Date	2019. 06. 09.
Team Name	LAJI [team8]
Member1	20160237 Jisu An
Member2	20133096 Hyunjae Lee
Member3	20143954 Jinwoo Jeon
Prof.	Sangoh Park

<Contents>

[Project Introduction]

1-1. Project overview

This is ***a web development project which helps Twitch streamers get highlight videos just by uploading their broadcasting videos based on Django.*** Basically, this website allows users to login by Google OAuth. Not only this website analyzes user's videos but also manage their videos. Also, users can pay by Toss application since they need coupons for getting highlights. Fig 1 shows a basic use-case scenario below.

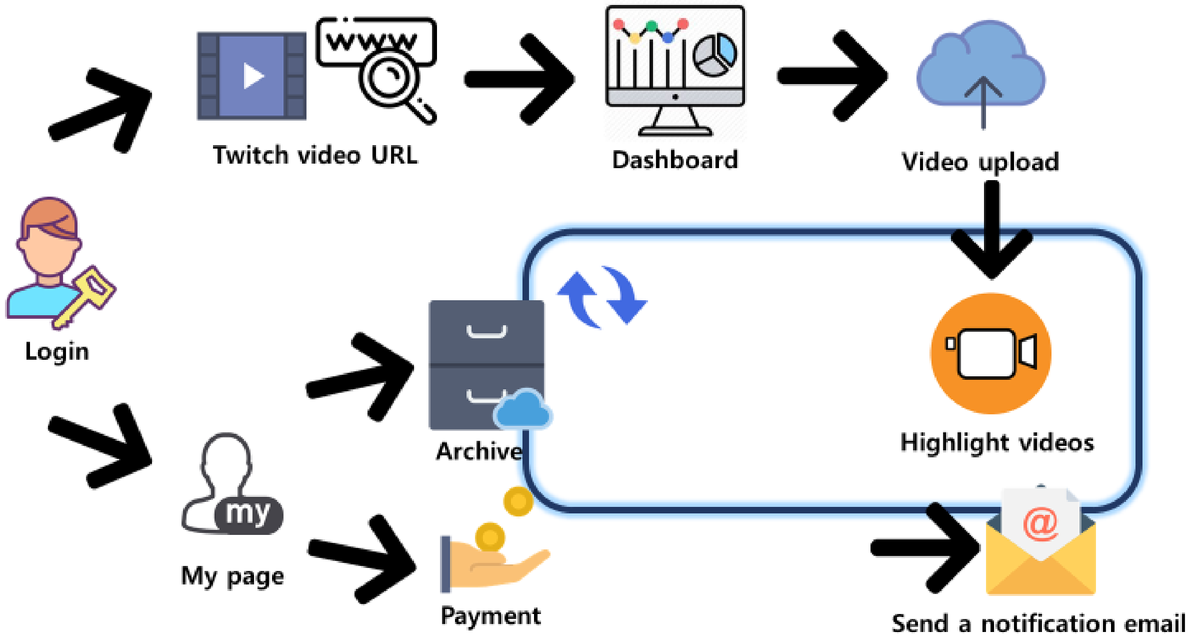
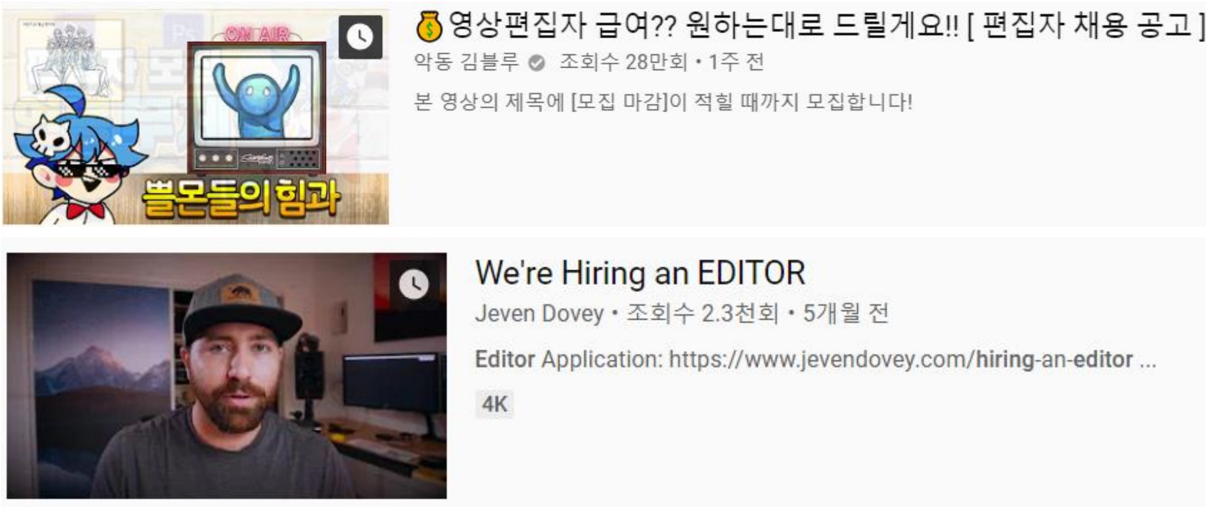


Fig 1 Basic use-case scenario

1-2. Idea proposal.



💰영상편집자 급여?? 원하는대로 드릴게요!! [편집자 채용 공고]

악동 김블루 ✓ 조회수 28만회 • 1주 전

본 영상의 제목에 [모집 마감]이 적힐 때까지 모집합니다!

We're Hiring an EDITOR

Jeven Dovey • 조회수 2.3천회 • 5개월 전

Editor Application: https://www.jevendovey.com/hiring-an-editor ...

4K

Fig 2 Recruitment Youtube videos.

Fig2 shows examples of recruitments for video editors. Let's think about the reason why they have announced. There are two facts to check.

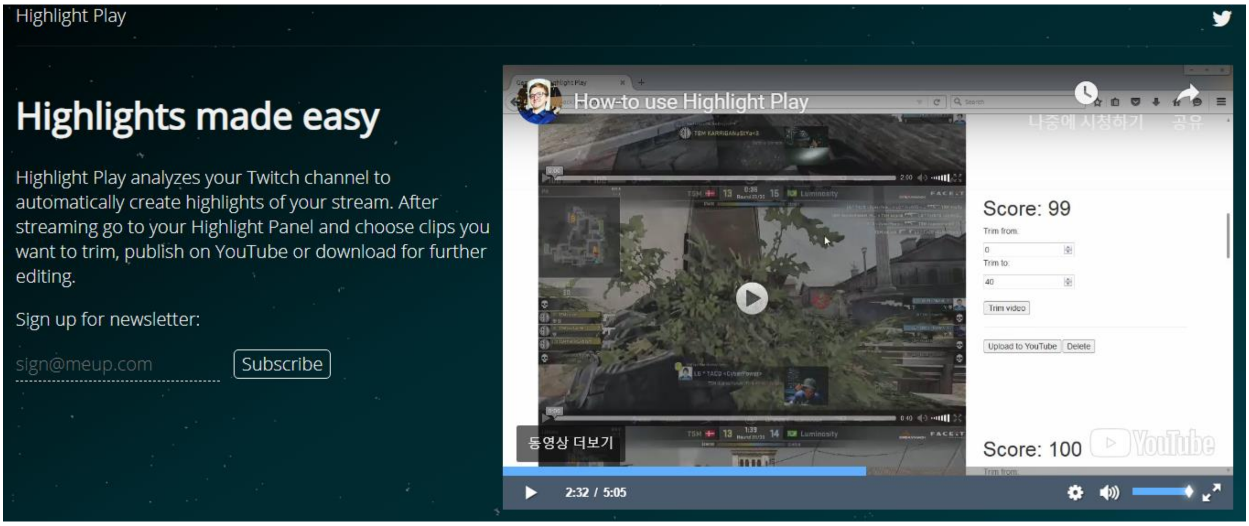
A. With the advent of personal broadcasting platforms such as Africa TV, Twitch and the growth of YouTube, a lot of people have started and been starting to broadcast what they like. Also, as the number of streamers increases, a bunch of people love to watch videos from YouTube, Twitch and so on. But they don't have enough time to watch many of them, so YouTube recommends videos users may like.

B. Twitch is a live video streaming platform which primarily focuses on video games. A lot of people are there to watch and broadcast. Once a streamer uploads a his or her broadcasting video on Twitch, normally his or her video editors make highlights videos from the uploaded one as soon as possible. But it takes a long time and big cost to hire editors.

You can see why streamers posted them through Fact A and B. ***Therefore, we proposed an idea to solve the social need with a web service which analyzes user's video to get what highlights are and gives back to users.***

1-3. Difference with similar services

a. Highlight Play



Highlight Play

Highlights made easy

Highlight Play analyzes your Twitch channel to automatically create highlights of your stream. After streaming go to your Highlight Panel and choose clips you want to trim, publish on YouTube or download for further editing.

Sign up for newsletter:

Fig 3 Highlight Play main page

Highlight Play analyzes your Twitch channel to automatically create highlights of your stream. After streaming go to your Highlight Panel and choose clips you want to trim, publish on YouTube or download for further editing.

When you start streaming the highlighter will automatically start recording and analyzing data.

The program also gives its own score for each clip and display keywords to hint you about the clip content. This helps you to find highlights that happened, but which you did not timestamp.

b. Twitch editor

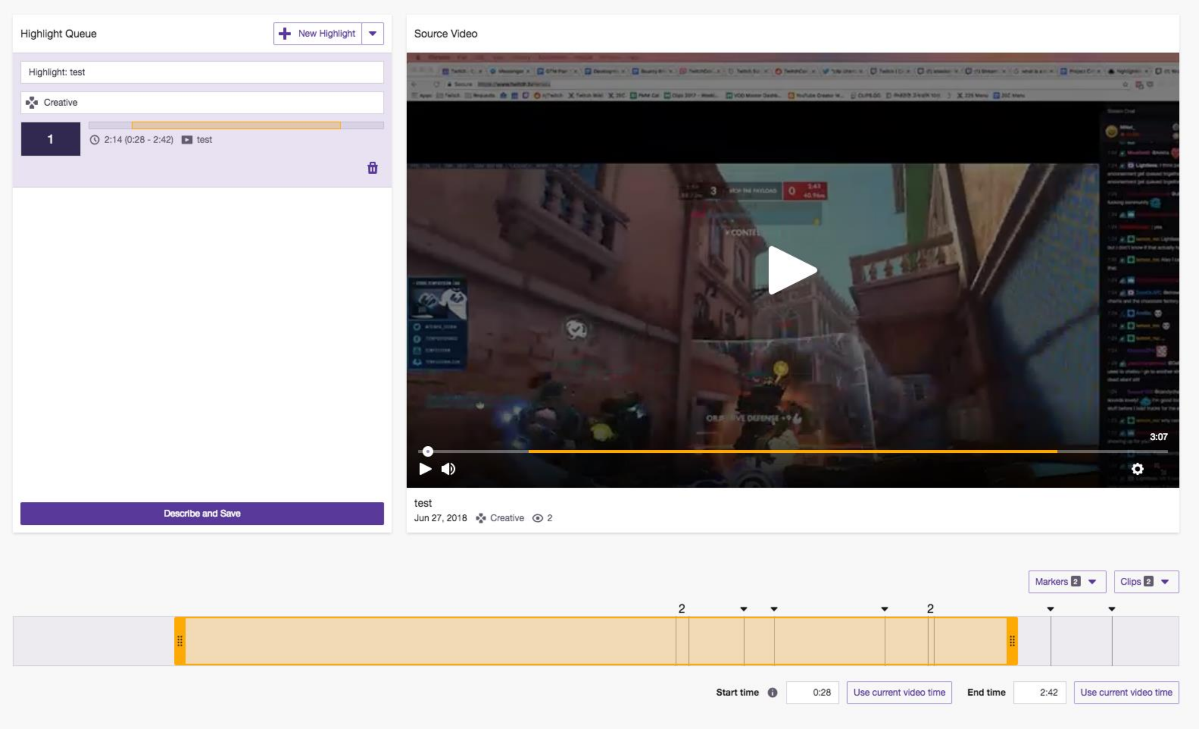


Fig 4 Twitch editor page

Twitch provides streamers with a highlight editing service

c. Movie Slice

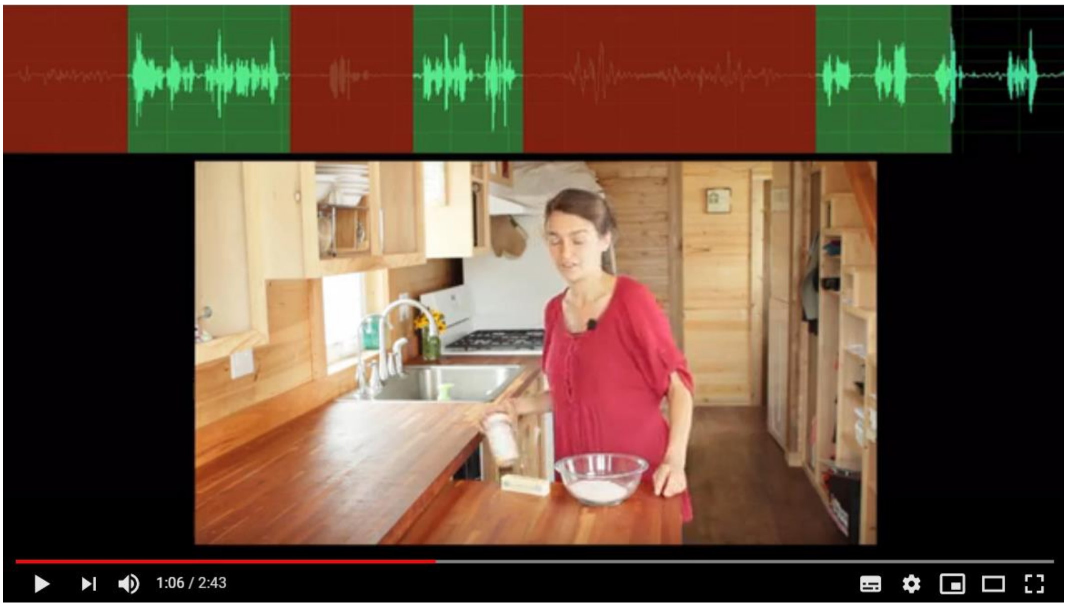


Fig 5 Movieslice use-case video

It is an automatic video editor that automatically removes unnecessary parts of the video.

1-4. Goal and target users.

" Our goal is to increase working efficiency for video editors and to give streamers a list of highlights with reasonable price."

We would like to give streamers or their video editors better choices when they edit videos. We primarily focus on streamers who are in *Twitch*.

[Developments]

2-1. Environments

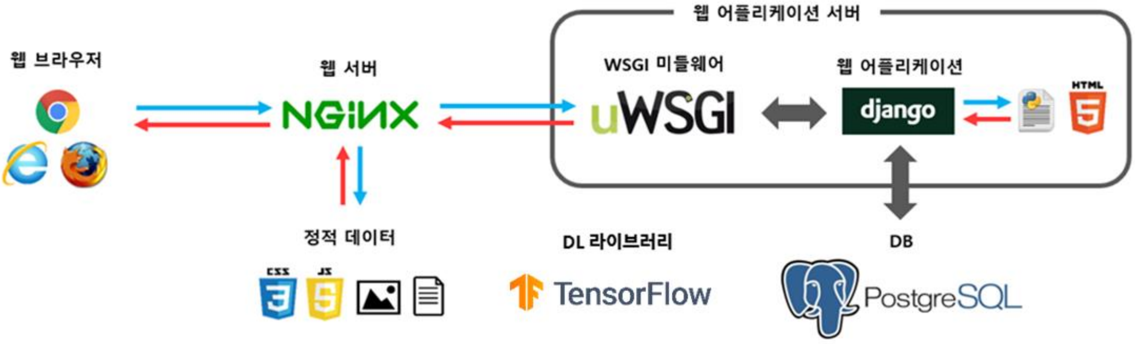


Fig 6 Development Environments

- Operating system: Ubuntu 16.04
- Web server: NginX
- Web Framework: Django 2.2.1
- WSGI MiddleWare: uWSGI
- Database server: PostgreSQL 9.6
- Security: http over ssl (https)
- Deep learning library: TensorFlow 1.13.1
- Sever specification: Intel I9-9900K

2-2. Features.

We have 5 main feature that are listed below.

A. Login with Google OAuth

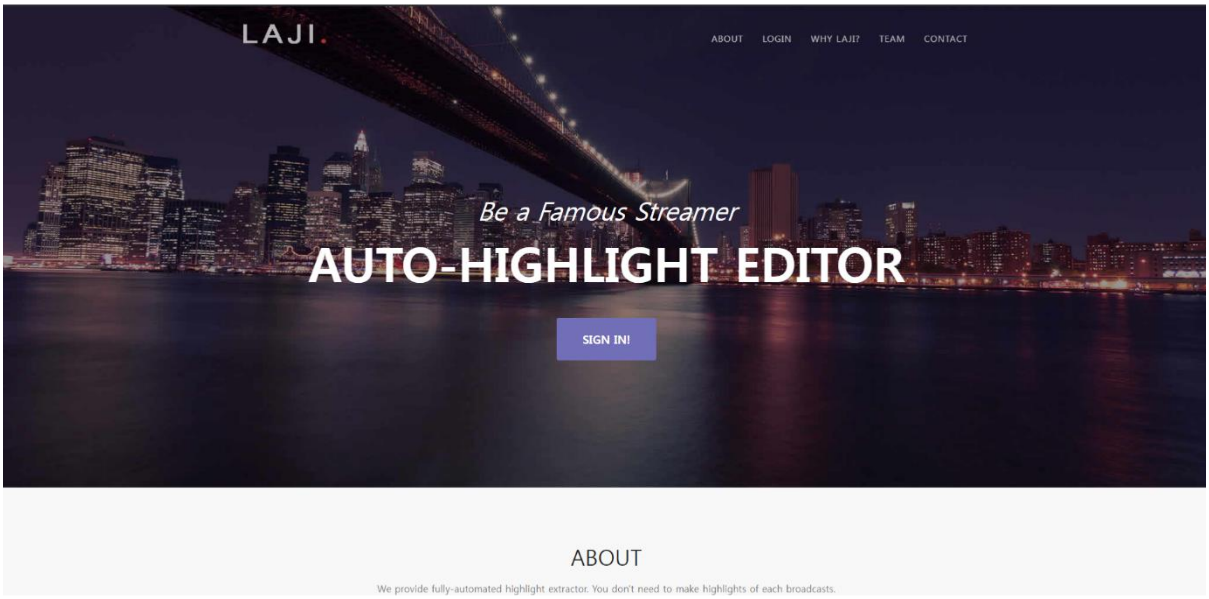
B. Dashboard

C. Algorithm to get highlights

D. Payment

E. Archive

< Login with Google OAuth>



Be a Famous Streamer

AUTO-HIGHLIGHT EDITOR

ABOUT

We provide fully-automated highlight extractor. You don't need to make highlights of each broadcasts.

Fig 7 Home page Before Login

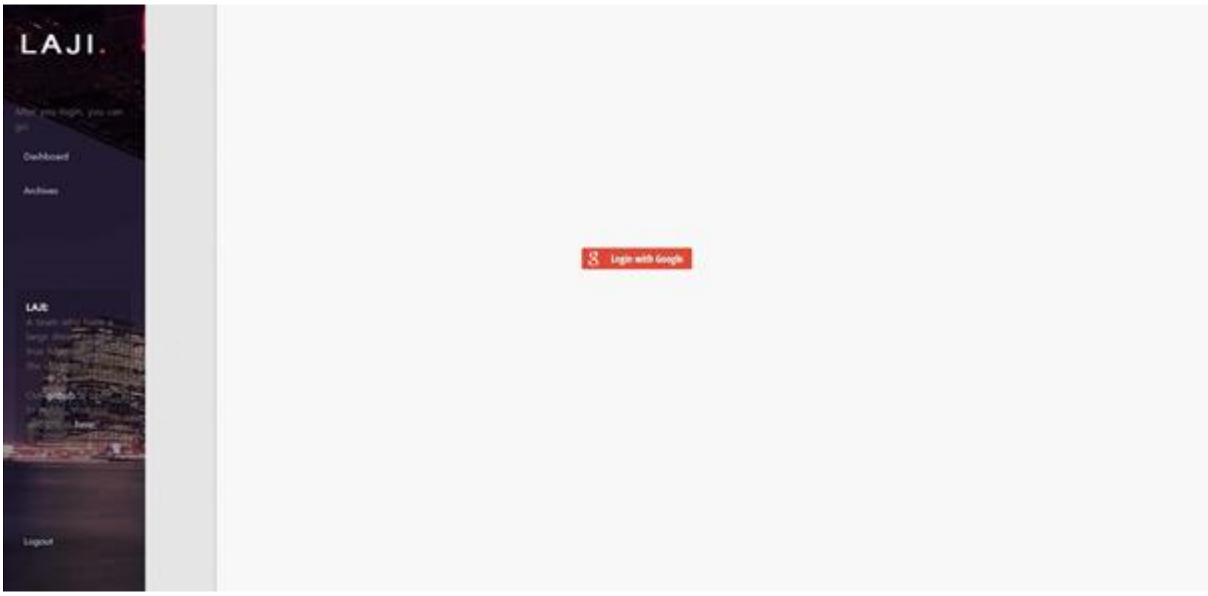


Fig 8 When users click "Login"

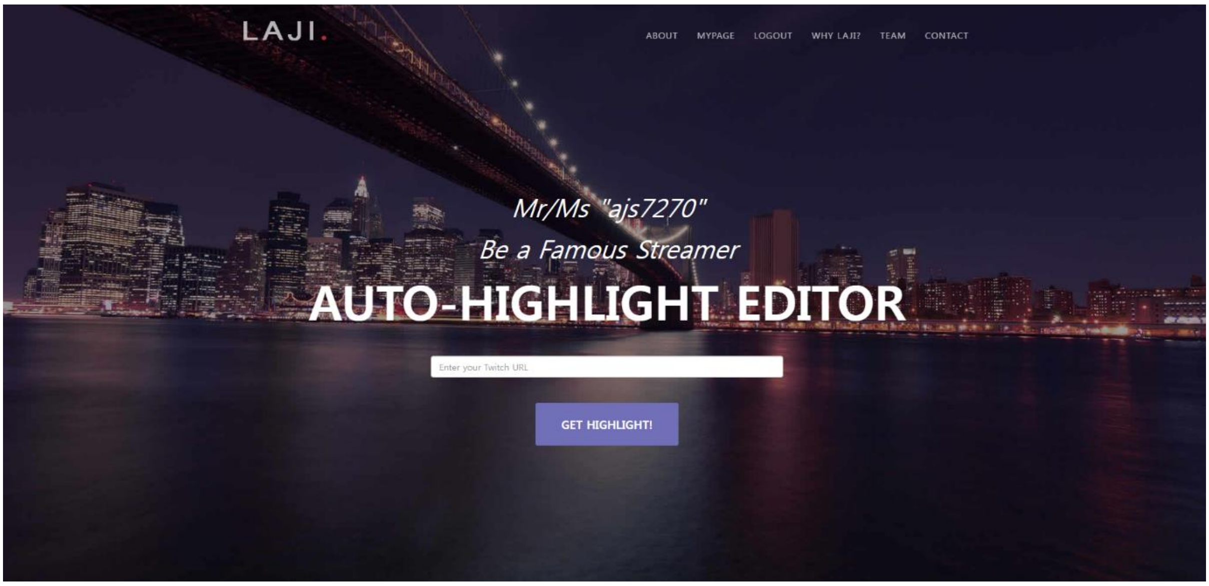
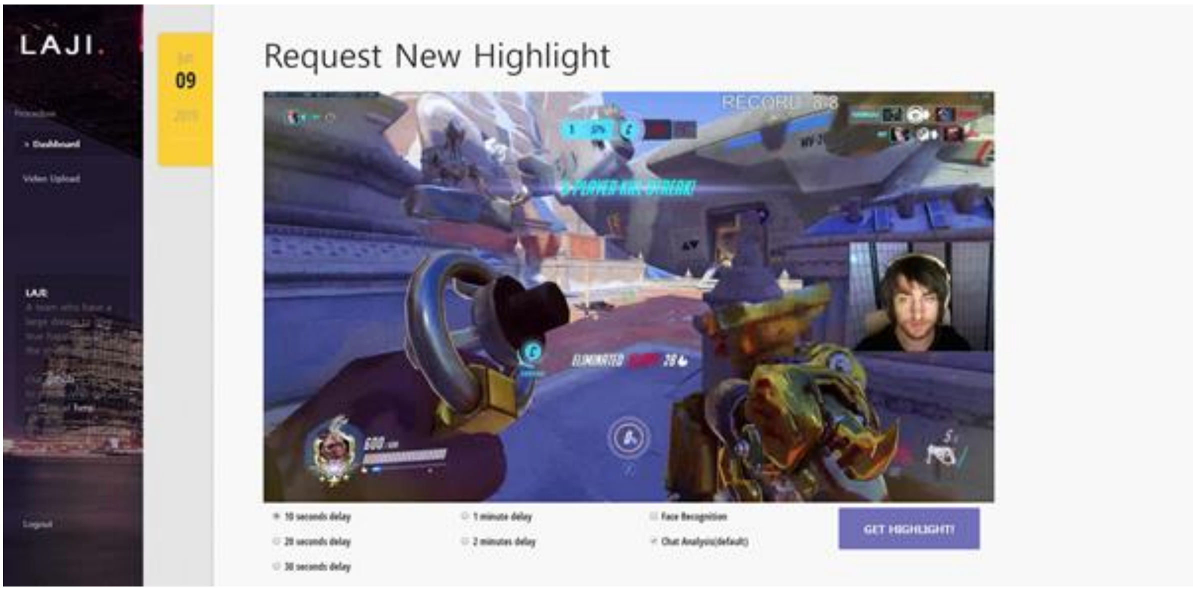


Fig 9 Home page After Login

We don’t have a sign-up form which makes users tired. Instead of that, we allow users to login in with their Google account. This is much intuitive and easy for users when they want to use our website. Fig 7, 8, 9 show how to login step by step. In Fig 9, Users can write down a Twitch video URL to access to Dashboard.

<Dashboard>



Request New Highlight

● 10 seconds delay
○ 20 seconds delay
○ 30 seconds delay
○ 1 minute delay
○ 2 minutes delay
□ Face Recognition
✓ Chat Analysis(default)

Fig 10 Dashboard main page

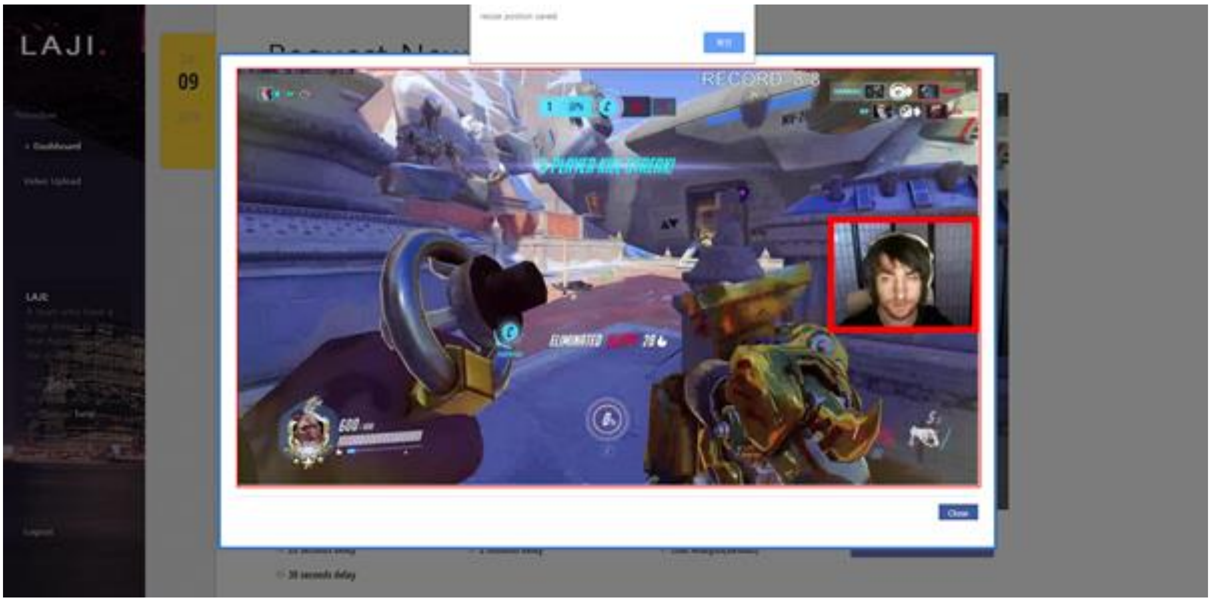
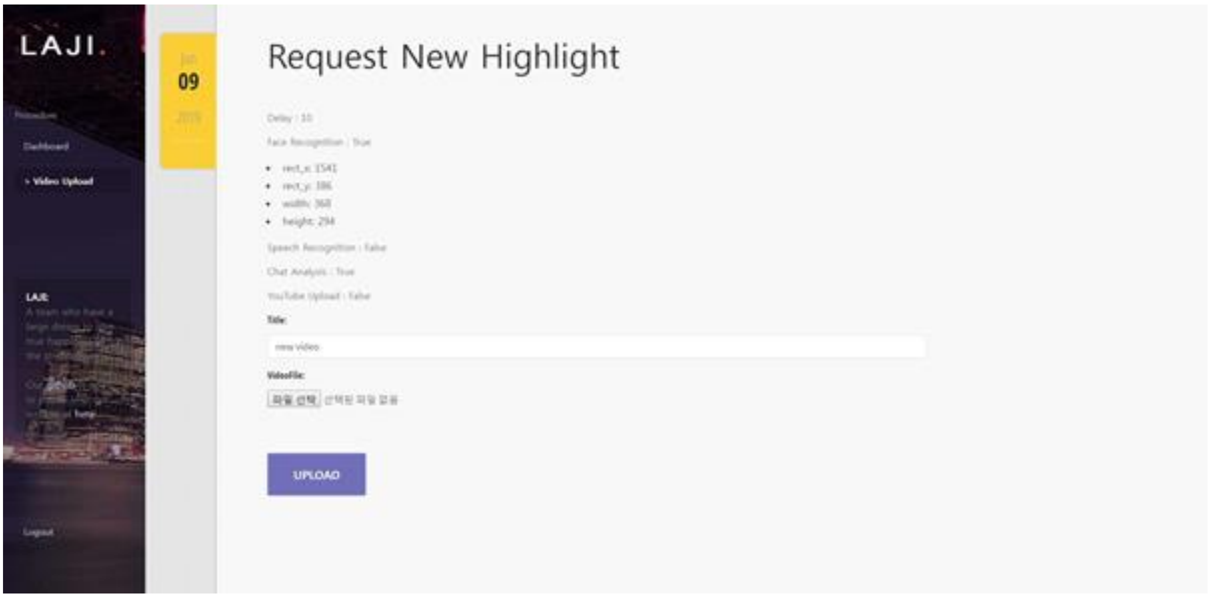


Fig 11 "Face recognition" is clicked



Request New Highlight

Delay : 10

Face Recognition : True

- rect_x: 1541
- rect_y: 386
- width: 368
- height: 294

Speech Recognition : False

Chat Analysis : True

YouTube Upload : False

Title:

new video

VideoFile:

파일 선택 선택된 파일 없음

Fig 12 Reviewing and Upload page

Dashboard is a page when users hit a correct Twitch video URL. When users hit a correct Twitch video URL, we will show users a Thumbnail of the given URL using Twitch API. In Fig 10, It shows Dashboard main page that users can decide whether they use a delay option and Face recognition option. They can choose a delay up to 2 minutes. A delay is a parameter that decide each highlight video's length. If users decide to use Face recognition, Fig 11 will be popped up and they will be able to draw a red rectangle where a streamer face is supposed to be located. After deciding options for making highlights, users will hit "GET HIGHLIGHT" and see an upload page that is showed in Fig 12. User can upload their broadcasting videos through this upload page.

<Algorithm to get highlights>

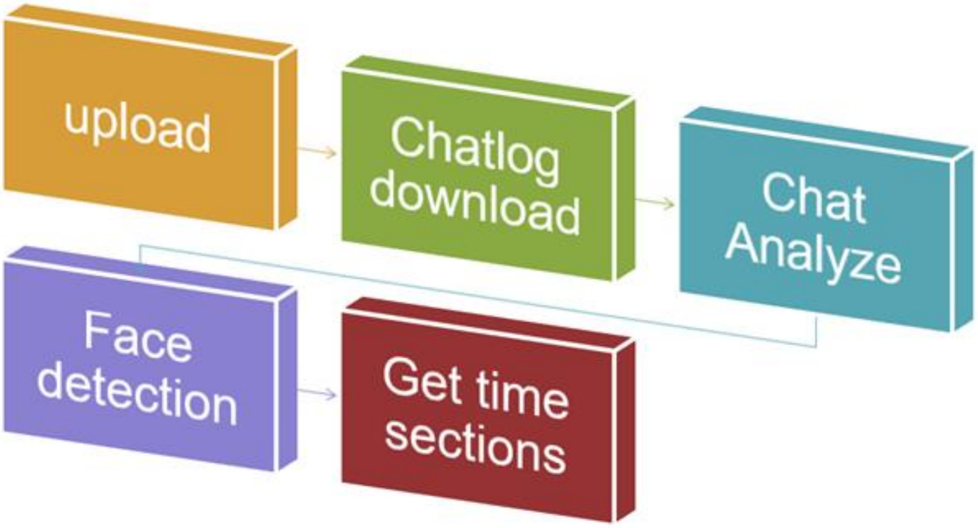


Fig 13 Steps for making highlights

Firstly, users upload their videos. Secondly, as soon as the videos are downloaded in our server, we open a thread in the queue and lets the thread to follow the remaining steps. So, it will download a chatlog of the given Twitch URL.

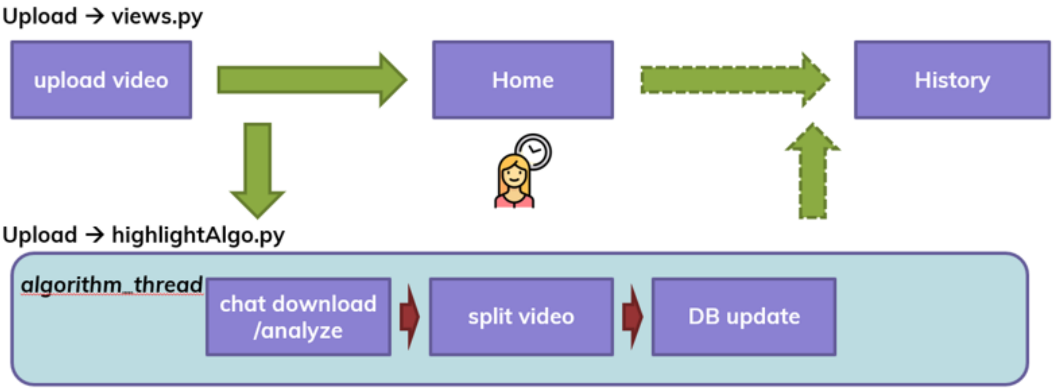


Fig 14 Threading

Thirdly, it will analyze the chatlog and here is a thing. we use labeled words to score each time. A score of a specific time in second is defined as counted how many labeled words are present at a given time.

$$Score(X) = \left(\sum_{i=1}^{5} score(x_i)\right)/5$$

Labeled words have two kinds. One is global and the other is local. Global labeled words are defined by PogChampNet [reference] and Twitch emotes [reference]. Local labeled words are most 10 common words in the chatlog.

After scoring, we use ***'K'*** numbers of specific times to do face recognitions.

Fourthly, it will use a given size of frames that users drawn at Dashboard to do face recognition in the video. We analyze ***'K'*** numbers of specific times whether frames in the times have a detected face with how many face expression percentage has.

The way to analyze each time in Face recognition section is as same as in Chat analyze section. The way to score each time is to average probabilities of detected face expressions which is calculated by Residual network model in the next 5 seconds

$$Score(M) = \left(\sum_{i=1}^{5} score(m_i)\right)/5$$

$$\mathrm{score}(m_i) = A\ probability\ of\ detected\ face\ expression.$$

After above steps, we will get final score ***'F'*** for selected times and defined as

$$Final\ Score(F) = \left(\sum_{i=1}^{\#i} score(X_i) * score(M_i)\right)$$

Fifthly, we will a list of 10 highlight time sections ***"H"*** and defined as

$$Highlights\ H = \{F_1\ , F_2\ , \dots , F_{10}\}$$

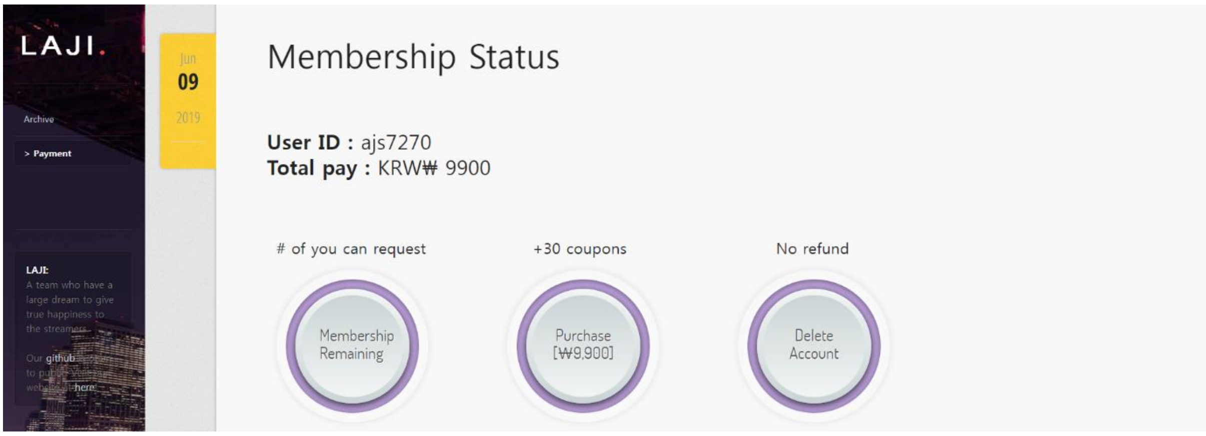
Then, we use the given delay to decide each video's length.

$$Sections\ S = \{S_1\ , S_2, \dots , S_i\}, \mathrm{i} \leq 10$$

$$S_j = [\ F_j - (2 * delay)\ , F_j + delay]$$

$$\mathrm{Iff}\ \ 1 \leq \mathrm{j} \leq 10, 0 \leq F_j - (2 * delay), F_j + delay \leq last\ time$$

<Payment>



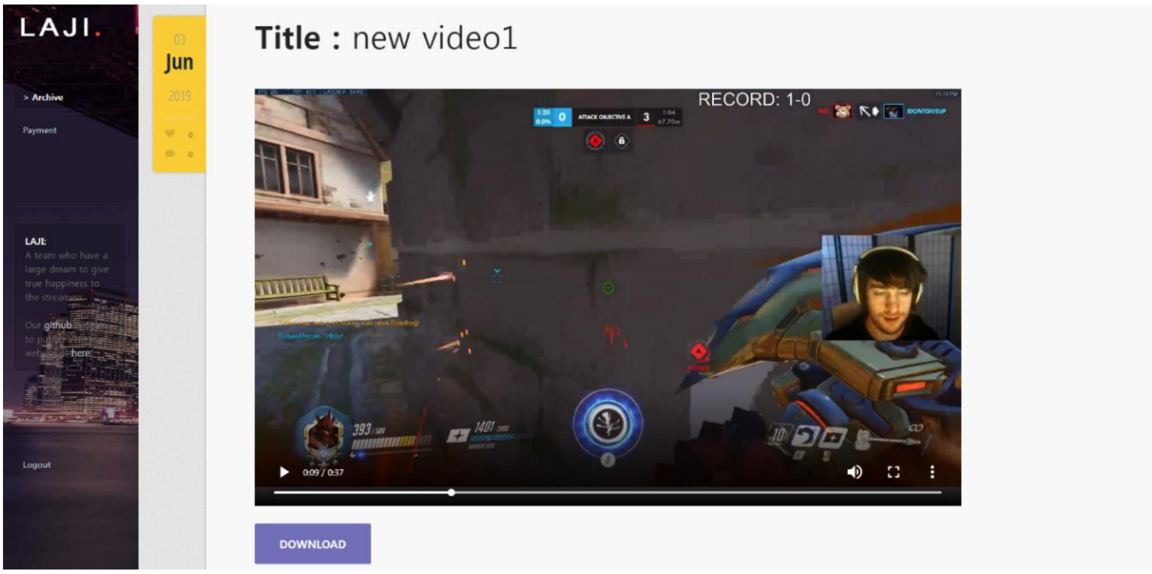
Membership Status

User ID : ajs7270
Total pay : KRW₩ 9900

Fig 15 Payment page

In Fig 15, it shows the payment page. Users can check their current coupons, buy coupons and leave their accounts.

<Archive>



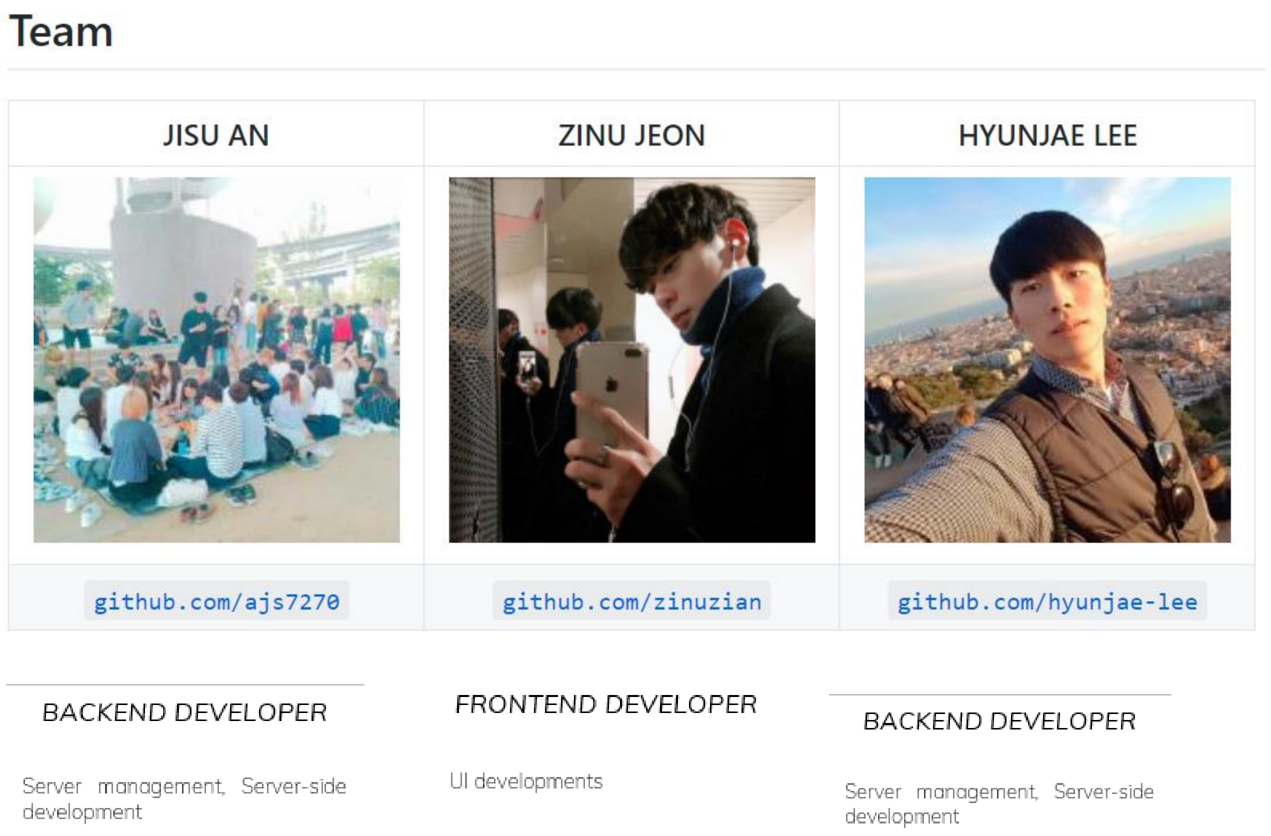
Title : new video1

Fig 16 User's Archive page

As soon as a thread finishes making highlights, it will update them to the user’s archive. And the user will be able to download the highlighted videos just by clicking the download buttons.

[Cooperation]

3-1. Member's roles



Team

JISU AN	ZINU JEON	HYUNJAE LEE
github.com/ajs7270	github.com/zinuzian	github.com/hyunjae-lee

BACKEND DEVELOPER

Server management, Server-side development

FRONTEND DEVELOPER

UI developments

BACKEND DEVELOPER

Server management, Server-side development

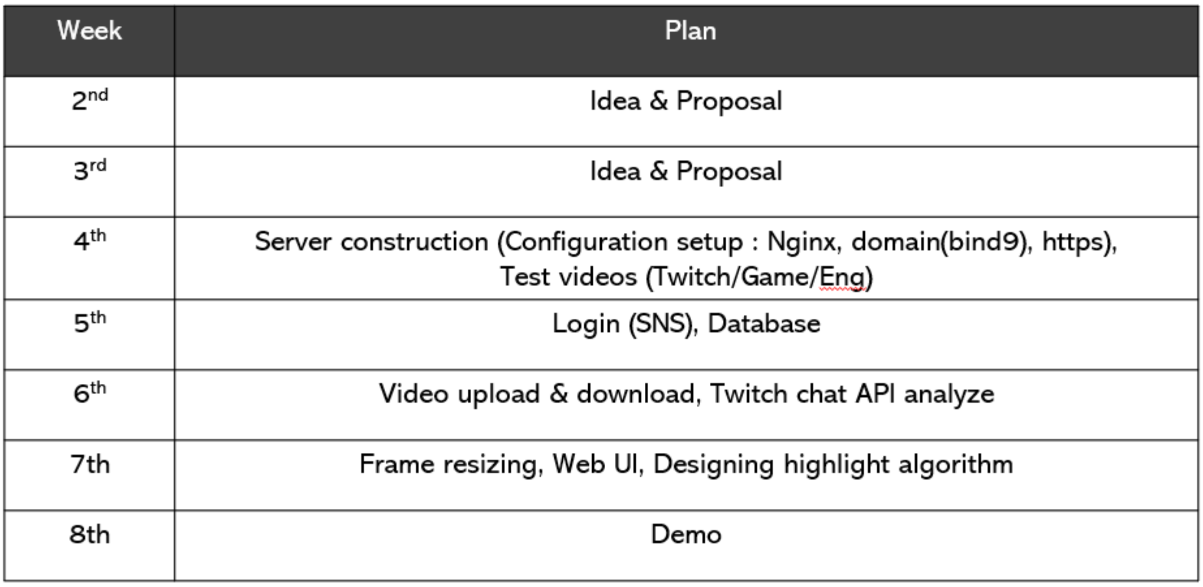
Jisu An is in charge of dealing with server setting and developing in server-side.

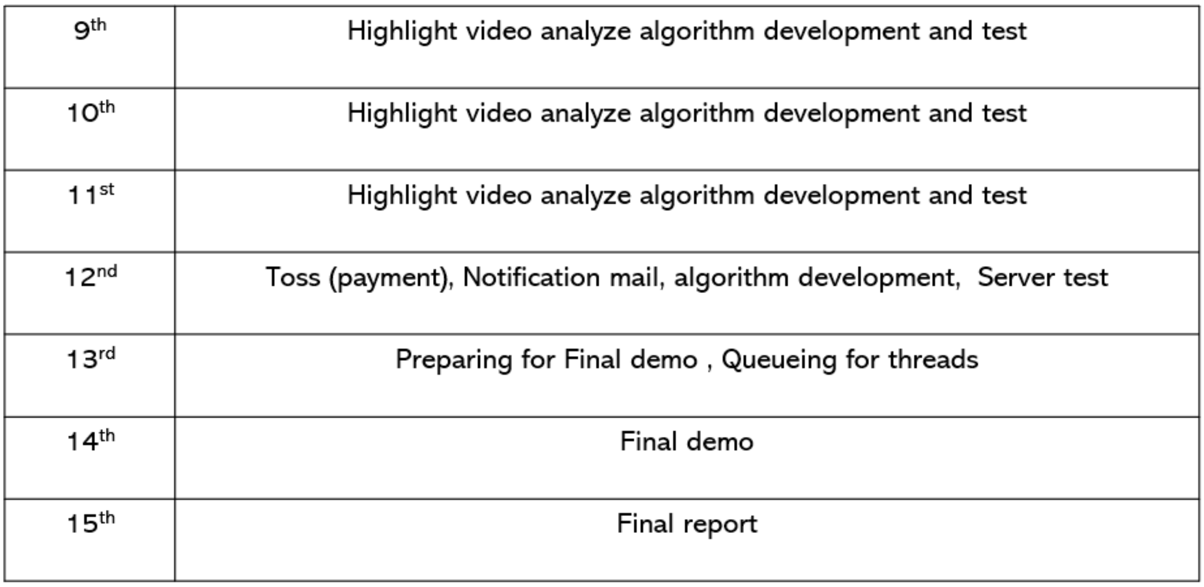
Jinwoo Jeon is in charge of developing User Interface and designing website front-end.

Hyunjae Lee is in charge of developing algorithm to get video's highlights.

3-2. Schedules

[Weekly plans]

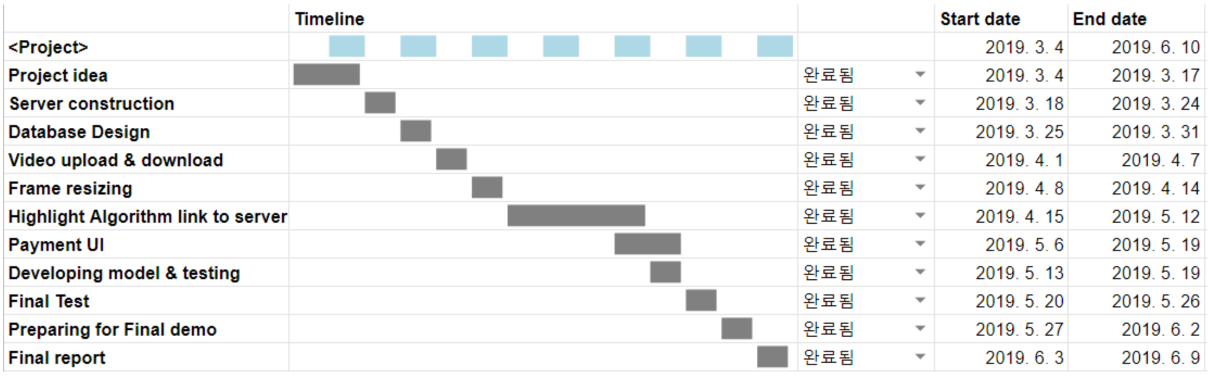




Week	Plan
2nd	Idea & Proposal
3rd	Idea & Proposal
4th	Server construction (Configuration setup : Nginx, domain(bind9), https), Test videos (Twitch/Game/Eng)
5th	Login (SNS), Database
6th	Video upload & download, Twitch chat API analyze
7th	Frame resizing, Web UI, Designing highlight algorithm
8th	Demo
9th	Highlight video analyze algorithm development and test
10th	Highlight video analyze algorithm development and test
11st	Highlight video analyze algorithm development and test
12nd	Toss (payment), Notification mail, algorithm development, Server test
13rd	Preparing for Final demo , Queueing for threads
14th	Final demo
15th	Final report

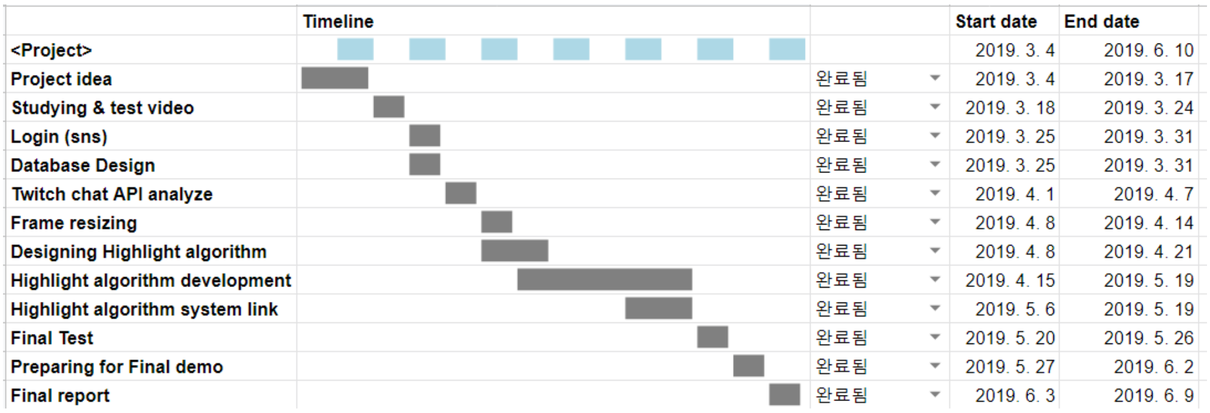
[Individual plans]

I. Jisu An



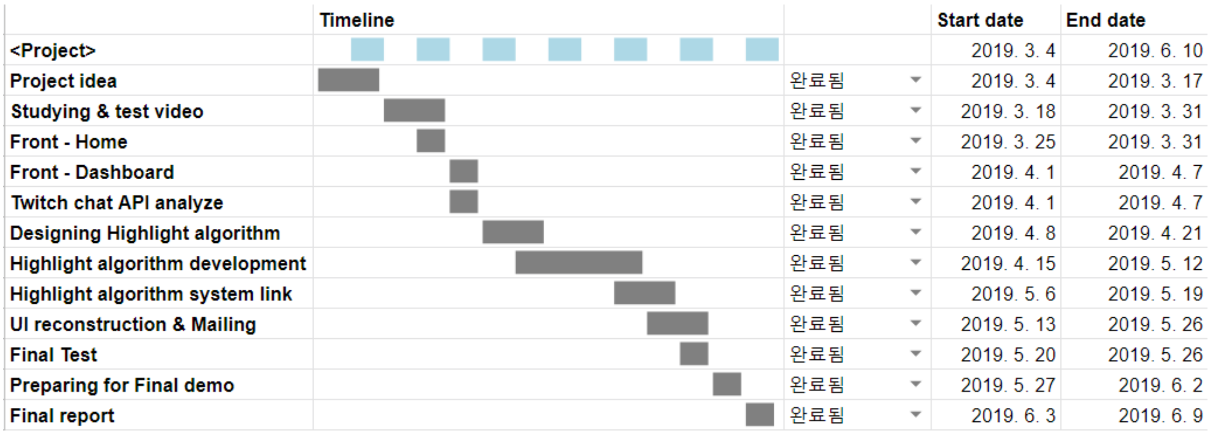
	Timeline			Start date	End date
<Project>				2019. 3. 4	2019. 6. 10
Project idea		완료됨	▾	2019. 3. 4	2019. 3. 17
Server construction		완료됨	▾	2019. 3. 18	2019. 3. 24
Database Design		완료됨	▾	2019. 3. 25	2019. 3. 31
Video upload & download		완료됨	▾	2019. 4. 1	2019. 4. 7
Frame resizing		완료됨	▾	2019. 4. 8	2019. 4. 14
Highlight Algorithm link to server		완료됨	▾	2019. 4. 15	2019. 5. 12
Payment UI		완료됨	▾	2019. 5. 6	2019. 5. 19
Developing model & testing		완료됨	▾	2019. 5. 13	2019. 5. 19
Final Test		완료됨	▾	2019. 5. 20	2019. 5. 26
Preparing for Final demo		완료됨	▾	2019. 5. 27	2019. 6. 2
Final report		완료됨	▾	2019. 6. 3	2019. 6. 9

II. Hyunjae Lee



	Timeline			Start date	End date
<Project>				2019. 3. 4	2019. 6. 10
Project idea		완료됨	▾	2019. 3. 4	2019. 3. 17
Studying & test video		완료됨	▾	2019. 3. 18	2019. 3. 24
Login (sns)		완료됨	▾	2019. 3. 25	2019. 3. 31
Database Design		완료됨	▾	2019. 3. 25	2019. 3. 31
Twitch chat API analyze		완료됨	▾	2019. 4. 1	2019. 4. 7
Frame resizing		완료됨	▾	2019. 4. 8	2019. 4. 14
Designing Highlight algorithm		완료됨	▾	2019. 4. 8	2019. 4. 21
Highlight algorithm development		완료됨	▾	2019. 4. 15	2019. 5. 19
Highlight algorithm system link		완료됨	▾	2019. 5. 6	2019. 5. 19
Final Test		완료됨	▾	2019. 5. 20	2019. 5. 26
Preparing for Final demo		완료됨	▾	2019. 5. 27	2019. 6. 2
Final report		완료됨	▾	2019. 6. 3	2019. 6. 9

III. Jinwoo Jeon

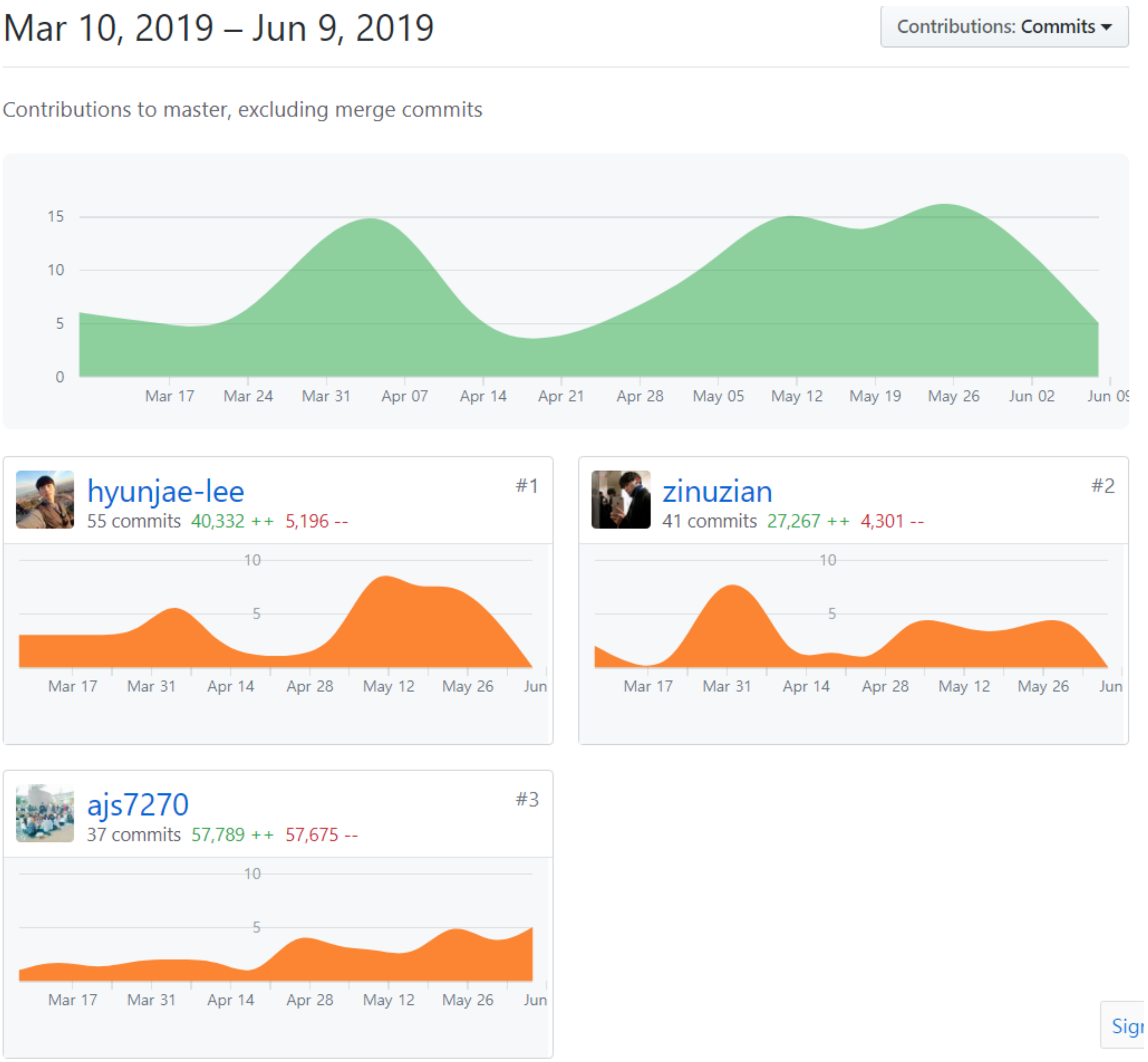


	Timeline		Start date	End date
<Project>			2019. 3. 4	2019. 6. 10
Project idea		완료됨 ▾	2019. 3. 4	2019. 3. 17
Studying & test video		완료됨 ▾	2019. 3. 18	2019. 3. 31
Front - Home		완료됨 ▾	2019. 3. 25	2019. 3. 31
Front - Dashboard		완료됨 ▾	2019. 4. 1	2019. 4. 7
Twitch chat API analyze		완료됨 ▾	2019. 4. 1	2019. 4. 7
Designing Highlight algorithm		완료됨 ▾	2019. 4. 8	2019. 4. 21
Highlight algorithm development		완료됨 ▾	2019. 4. 15	2019. 5. 12
Highlight algorithm system link		완료됨 ▾	2019. 5. 6	2019. 5. 19
UI reconstruction & Mailing		완료됨 ▾	2019. 5. 13	2019. 5. 26
Final Test		완료됨 ▾	2019. 5. 20	2019. 5. 26
Preparing for Final demo		완료됨 ▾	2019. 5. 27	2019. 6. 2
Final report		완료됨 ▾	2019. 6. 3	2019. 6. 9

3-3. Managements

Github link : https://github.com/laji-cau

(In the link, there are two repositories. One is called “LAJI-HIGHLIGITING” which is for developments and the other is called “Capstone-Project-2019-Doc”)



Mar 10, 2019 – Jun 9, 2019

Contributions to master, excluding merge commits

hyunjae-lee #1
55 commits 40,332 ++ 5,196 --

zinuzian #2
41 commits 27,267 ++ 4,301 --

ajs7270 #3
37 commits 57,789 ++ 57,675 --

[Result]

1. Users can login in and use this web service through the Google login form.

2. “My page” allows you to check and change your personal data.

A. Payment: You may purchase coupons through Toss application to use the web service. One coupon can use one highlighting service. It also provides the ability to leave this service.

B. Archive: Highlight videos are listed by date. In addition to being able to play and view highlight videos on the archive page, the videos can be downloaded from the server.

3. After login, the user enters Twitch video URL and moves to the Dashboard.

4. Dashboard shows the thumbnail image of the corresponding broadcasting video through the input URL.

5. Dashboard allows you to select options for highlighting as well as checking the corresponding thumbnail image.

A. Delay: It is a parameter that determines the length of one highlight video. If delay is 10 seconds, each highlight video is about 30 seconds long.

B. Face Recognition: It checks a given size of frames to check if there is a detected face and the face expression percentage.

C. Chat Analysis: This is the default option and analyzes the chat by scoring each time with labeled words.

6. After selecting options for highlighting in Dashboard, press the Get Highlight button to head to the next screen.

7. On the next page, the user will be able to review various options selected from the previous screen, and the user can directly upload a broadcasting video.

8. When the user finishes uploading the broadcast video, user will be redirected to the main page.

9. In the server of this service, when the highlighting extraction and the rendering are completed, the generated highlight videos will be uploaded to the user's archive. Then, the user will be notified that highlighting is finished by e-mail.

10. As we check the result (highlight video), we found that correlation between highlight and chat burst is very high and at that burst time, regardless of the genre of the game, the usual exclamation came out.

11. As we check the result (highlight video), we found that correlation between highlight and streamer’s non-neutral (happiness, fear,

sadness and surprise) expression is very high regardless of the genre of the game.

Finally, we made a web service with basic membership management system that automatically extracts highlight of the video based on chatlog and it gave us meaningful results.