

2021 캡스톤디자인 최종보고서

-채팅 데이터 분석을 통한 편집 자동화 SW-



팀 명 : HighLive

김 산 32170578

강정현 32170078

이원준 32173243

허 진 32174937

지도 교수 : 이재동 교수님

목차

I 개요	2
II 현황 및 문제점	2
III 비즈니스 모델	4
IV 팀 소개	7
V 프로젝트 세부사항	8
V-1 기술 개요	8
V-2 기술 상세	9
V-3 협업 로그	15
VI 결과	16
VII 향후 계획	18
VIII 기술적 위협요소 분석	18
IX 회의록 및 작업 후기	19
IX-1 회의록	19
IX-1 작업 후기	21

I 개요

과거에는 동영상 파일이라고 한다면, 청각적, 시각적 데이터와, 영상정보에 대한 부수적인 메타데이터만 존재했지만, 최근 1인 방송 플랫폼(Youtube, Twitch등)의 등장으로, 실시간 인터넷방송 기반의 동영상들은 시청자들의 실시간 채팅데이터를 가지고 있다. 이러한 채팅데이터를 기반으로 채팅 트래픽분석, 자연어처리 등의 기법들을 활용하여 영상에 대한 분석 데이터를 웹 서비스를 통해 제공한다. 이를 통해 영상의 편집을 담당하는 편집자로 하여금 보다 쉽게 영상의 하이라이트 구간을 판단할 수 있도록 해, 컷편집의 단순 반복작업을 자동화, 하이라이트 영상을 제공하는 것을 목표로 삼는다.

II 현황 및 문제점분석

1) 국내외 관련 지식재산권 현황

지식재산권명	지식재산권 출원인	출원국/출원번호
채팅 참여자의 감정 상태 인식 장치 및 이를 이용한 방법	한국전자통신연구원	한국/1020150063070 (2015.05.06)
분석 : 채팅참여자의 채팅입력시간, 표정을 통해 채팅 참여자의 감정분석을 한다는 개념. 제안기술과의 차별성 : 채팅데이터를 통해 채팅 참여자들의 감정 상태를 인식하는 점에서 동일하나 자연어 처리기술을 이용하여 채팅데이터의 문장 자체에서 감정을 분석을 한다.		
지식재산권명	지식재산권 출원인	출원국/출원번호
동영상 콘텐츠의 하이라이트 영상을 추출하여 제공하는 방법과 시스템 및 기록 매체	네이버 주식회사	한국/1017772420000 (2017.09.05)
분석 : 동영상의 댓글참여자가 추출한 일부구간, 구간에대한 누적재생시간 데이터를 통해 하이라이트 구간을 추출한다는 개념. 제안기술과의 차별성 : 동영상의 어떤 데이터를 활용하느냐 에서 차별점이 있다. 상기 특허의 경우 구간추출을 위해 시청자들이 직접 댓글을 통해 구간을 추출한 정보를 활용하나 주요구간에 대한 직접적인 정보가 없는 데이터를 활용한다는 점에서 다르다.		

2) 기존 편집 SW의 높은 진입장벽 및 긴 컷편집 시간 소요

기존 영상편집 소프트웨어(프리미어 프로, 베가스) 의 경우 초기 진입장벽이 높은 편이다. 이를 고려하여 최근에는 인터페이스가 간소화된 영상편집 소프트웨어가 출시되고 있는 실정이나 여전히 긴 영상을 컷 편집하는 단순 반복작업 과정에서 상당

한 시간이 소요된다. 실시간 인터넷 방송 기반의 동영상들은 방송 시작부터 종료까지 녹화된 영상이다.

2020년도 트위치 스트리머 방송시간 시청시간 (2020.01.01 ~ 2020.12.31)															
	닉네임	방송시간	시청시간	닉네임	방송시간	시청시간	닉네임	방송시간	시청시간	닉네임	방송시간	시청시간	닉네임	방송시간	시청시간
1	송철영	2,984	40,073,507	31	연두는말안드워	2,075	5,130,229	61	재속함	2,064	3,095,294	91	양아지	1,607	1,822,133
2	한동숙	2,315	26,018,717	32	고차비	2,755	5,107,509	62	개인	2,160	3,039,621	92	강소연	1,454	1,782,429
3	서세봉남	3,199	20,161,119	33	양영	566	5,061,153	63	태준미	1,352	2,966,272	93	밋배리	809	1,693,931
4	괴물작123	2,155	17,662,891	34	안간팔리	2,773	5,016,533	64	김불라	1,798	2,959,603	94	종박	938	1,690,391
5	미로니	2,642	13,675,299	35	해리	1,784	4,892,428	65	우굴	3,174	2,887,129	95	zoodasa	655	1,681,627
6	침착맨	1,699	13,506,688	36	피닉스빅1	2,992	4,684,198	66	팟타수아	1,461	2,838,288	96	과로사1	1,368	1,640,890
7	녹두로	2,568	13,064,300	37	이준향	2,896	4,628,999	67	매직박	1,667	2,810,567	97	스나랑	2,002	1,637,487
8	우육관	2,691	12,447,596	38	꽃핀	1,923	4,627,827	68	크갯66	3,161	2,653,313	98	예랜디라	1,485	1,548,459
9	김진우	2,136	10,997,540	39	옥남이	1,629	4,563,904	69	복산	1,250	2,615,198	99	박옥자누나	1,487	1,512,344
10	김도	2,382	10,938,882	40	소우물	2,934	4,505,471	70	김대르니야	3,009	2,596,140	100	던	1,353	1,504,841
11	연아	936	10,874,857	41	개복어	2,707	4,470,997	71	알보	604	2,526,025	101	핵가님	3,294	1,467,914
12	Faker	361	10,731,479	42	올개입방송	3,382	4,456,055	72	미라지	1,498	2,447,574	102	heavyRainism	1,971	1,464,076
13	살인마협회장	2,299	9,559,443	43	파카9999	985	3,363,703	73	배로	1,868	2,419,372	103	마파	1,071	1,451,437
14	울프	860	9,510,406	44	머독	1,316	4,344,365	74	혁징	2,029	2,385,617	104	김남봉	1,279	1,429,751
15	캠프	2,034	9,225,548	45	강무피	1,615	4,242,998	75	강귀	1,806	2,330,293	105	이조홍	1,284	1,407,138
16	물리리	2,127	8,910,408	46	혼옥이	1,454	4,131,817	76	송우이빠	1,432	2,296,218	106	마스카	3,869	1,379,123
17	텃밭바비	1,576	8,637,230	47	공학준	1,157	4,024,924	77	지진준	2,064	2,185,917	107	매드라이프	1,415	1,343,563
18	영배선	1,744	8,456,732	48	다주	1,942	3,959,851	78	알락킹	3,137	2,181,923	108	개리	4,118	1,325,981
19	김나성	3,870	7,336,499	49	홍방장	2,082	3,901,678	79	개리형	525	2,143,579	109	태디미	387	1,311,796
20	오징	1,788	6,604,446	50	나나알	2,123	3,867,009	80	박혁도	2,463	2,140,852	110	ryujehong	1,188	1,258,893
21	칠면조심	2,540	6,519,284	51	연두부	2,475	3,777,577	81	크캉크	2,317	2,134,666	111	소탕삼612	409	1,201,208
22	울왕울이	2,979	6,131,192	52	소용왕니	1,092	3,662,597	82	지수소녀	2,142	2,107,014	112	우주떡따	775	1,198,849
23	삼프	3,120	5,877,703	53	금사향	1,839	3,574,649	83	린봉	1,455	2,090,027	113	찰미님	1,564	1,193,811
24	감지	2,283	5,690,204	54	똥면	1,059	3,540,416	84	루다니	2,525	2,079,796	114	니콜	758	1,179,521
25	로저	2,546	5,614,670	55	건지채	1,417	3,454,330	85	스마드스닉	3,768	2,045,111	115	액시스트라이클	897	1,175,616
26	연대	2,546	5,614,670	56	자돌	2,365	3,406,700	86	수현스럼	1,392	2,004,262	116	담양생	2,525	1,171,868
27	알업	925	5,560,256	57	햇살살	2,052	3,361,719	87	배도	2,575	1,975,435	117	데사미	172	1,165,192
28	선배	1,620	5,276,740	58	dopa24	366	3,351,753	88	마렌타	1,711	1,953,134	118	손당무	1,382	1,164,112
29	도현	2,435	5,249,187	59	코앞트	2,394	3,252,172	89	서현	1,734	1,889,300	119	김블루	887	1,124,102
30	소니소	1,831	5,173,041	60	러너고	1,600	3,248,578	90	경성래비	547	1,889,040	120	김달갈	1,871	1,084,126

<2020년도 트위치 스트리머 방송시간, 시청시간>
(<https://tgd.kr/s/hatsalsal/49562496>)

위 표의 데이터에 따르면, 2020년도 트위치 스트리머 시청시간 기준 상위 149명의 연간 평균 스트리밍 시간은 1733시간이다. 1년은 52주이므로, 스트리머당 주간 평균 스트리밍시간 즉, 1주당 녹화된 동영상의 평균 길이는 33.3시간이다. 이를 10분 분량의 영상으로 간추린 ‘주간 핫클립’ 영상을 만든다고 가정하였을 때 편집자 입장에서 단순반복작업량이 꽤 상당하다는 것을 알 수 있다.

3) 영상편집 외주 비교

영상 편집 외주의 경우 적정가격을 정하는 객관적 기준이 없다. 온라인 아웃소싱 중개업체 ‘숨고’에 따르면영상 편집을 외주를 맡길 경우 건당 230,000원 정도의 높은 비용이 발생한다.

4) 모바일 영상 편집 애플리케이션과의 비교

영상편집 모바일 어플리케이션(Viva video, 키네마스터 등)의 경우 모바일 플랫폼 특성상 화면의 크기와 인터페이스가 제한적이기에 긴 영상의 컷편집에 어려움을 겪을 수 있다. 컷편집 과정을 반자동화 한다면 모바일 환경에서 인터페이스를 간소화하고 생산성을 높일 수 있다.

III 비즈니스 모델

1) 매출전망

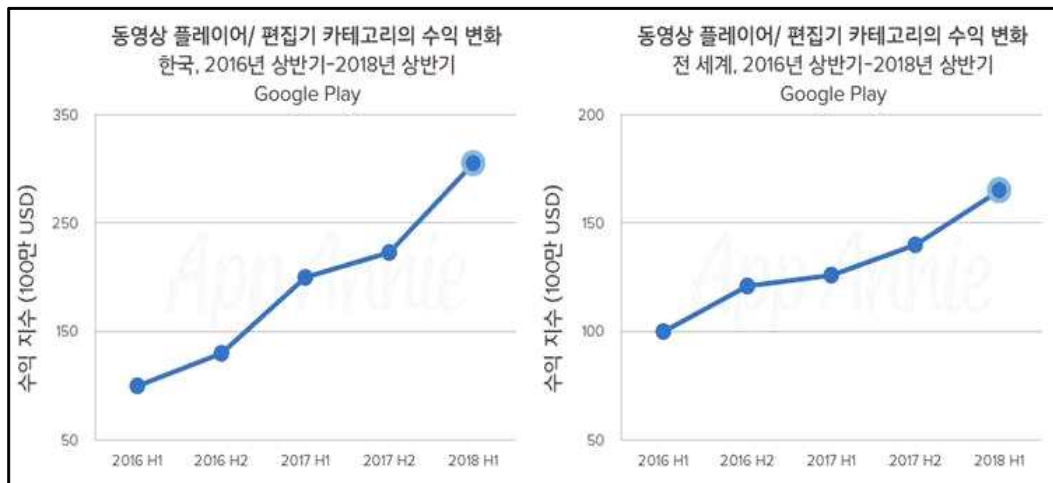
- 16년 국내 MCN 시장 규모는 대략 2,000억~3,000억 원 추산. (소프트웨어 정책 연구소)

순위	회사명	한 달간 구독자 증가 숫자	MCN	주요 내용
1	브로드밴드	3,000만	아프리카TV	· 2015년 9월 인도에 MCN 창작자 전용 YouTube Space 개설하고 인도의 영화 교육기관인 Whistling Woods International과 협력해서 오디오 및 동영상 편집 장비 등을 제공
2	플스크린	2,440만		
3	베보	2,180만		
4	머시나마	1,450만		
5	메이커젠	1,230만		
6	스타일홈	1,140만	CJ E&M	· 2013년 7월 CJ E&M이 처음 MCN 개념을 도입 · '다이아 TV'를 출시하면서 MCN 시장에 뛰어 현재 650명의 크리에이터를 보유하고 있으며, 구독자는 3,600만 명에 월평균 8억 뷰를 돌파
7	콜렉티브디지탈	910만		
...	트레저헌터	· CJ E&M MCN사업팀장 소속이었던 송재홍 대표가 국내 대표 크리에이터 양명, 악어, 김이브 등과 함께 2015년 1월 트레저헌터 설립 · 최근 트레저헌터에 소속된 크리에이터는 총 150명, 구독자는 약 1,200만 명 확보
31	CJ E&M	148만		
...		
96	트레저헌터	47만		

자료 : 소셜블레이드(2016.1.14.), 최근 언론기사

자료 : 소셜블레이드(2016.1.14.), 최근 언론기사

- MCN 시장의 지속적 성장으로 인해 개인방송 편집에 대한 수요증가가 예상된다. 개인 방송 콘텐츠의 다양화로 24시간 방송 콘텐츠 등 편집 부담이 큰 방송 콘텐츠 또한 증가하고 있는 상태.
- 짧은 시간내에 간편하게 즐기는 스낵컬처 문화가 확산됨에 따라 크리에이터들은 방송종료후 하이라이트 영상으로 콘텐츠를 재생산하여 수익을 키우고 있다. 예를 들어 10시간 분량의 방송 영상을 그대로 업로드하기 보다는 10분 안팎의 하이라이트로 크리에이터의 콘텐츠 진입장벽을 낮춰야 더 많은 구독자를 얻을 수 있고, 이는 더 많은 광고 수익으로 이어진다.
- 편집자들은 크리에이터의 실시간방송 원본을 통해 주요구간을 추출하는데 게임방송처럼 분량이 긴 방송의 경우 짧으면 3시간에서 길면 10시간 가량의 방송분량이 나오게 된다. 하이라이트 구간을 추출하기 위한 컷편집을 하기 위해선 편집자는 방송분량의 2~3배에 달하는 시간을 할애해야 한다. 드림챗은 이러한 컷편집을 위한 단순 작업을 자동화하여 편집을 위한 시간과 비용을 절감함은 물론 크리에이터들이 더 창의적인 콘텐츠 제작에 집중할 수 있게 해준다. 이는 신규 크리에이터들의 진입장벽을 낮추고 방송시장을 성장시킴으로써 편집수요를 증가시키는 선순환효과 또한 기대된다.



<구글플레이 동영상 플레이어 및 편집기 카테고리 총 소비자 지출 변화>

- 모바일 플랫폼에서의 영상편집 수요도 늘어나고있는 추세이다. 시장조사업체 앱애니에 따르면 구글플레이 기준 동영상 플레이어 및 편집기 카테고리 내 한국 스마트폰 이용자의 소비자 지출이 2016년 상반기 대비 2018년 상반기 3배 이상 성장을 기록했다. 드림챗은 간편한 인터페이스를 제공하여 모바일 시장도 공략하고자 한다

2) 첫 번째 개발 목표 - 개인방송 제작자와 협업

- 외주 편집자를 구하는 개인방송 제작자와 협업을 통해 채팅 데이터를 확보하고 지속적인 강화 학습을 통하여 완성도 높은 소프트웨어 출시.
- 개발 이후 영상 편집 기초에 관한 유튜브 콘텐츠를 제작하여 드림챗 간접 광고, 구글 애드센스를 통해 광고, 웹페이지를 통한 판매 진행.

3) SWOT

- S(Strength):
 - i) 하이라이트를 추출하는 과정을 자동화 함으로써 기존 편집시간 단축.
 - ii) 기존 콘텐츠 크리에이터들의 전문 편집자 고용부담 축소.
 - iii) 편집과정을 단축시킴으로써 신규 콘텐츠 크리에이터들의 진입장벽을 낮춤.
- W(Weakness):
 - i) 채팅 데이터 분석을 위한 NLP기술 뿐만 아니라 편집 프로그램 개발을 위한 복합적 기술력 요구.
 - ii) 채팅 트래픽 데이터 추출을 위한 각 개인방송 플랫폼(아프리카, Youtube, Twitch 등) 과의 호환성 확보 필요.
 - iii) 하이라이트 추출의 정확도가 채팅 데이터 양에 종속적.
- O(Opportunity):
 - i) 기존 콘텐츠 제작과정의 시간과 비용감축 -> 신규 콘텐츠 제작 촉진 -> 인터넷 방송 시장의 성장 -> 편집 프로그램의 수요 증가.
 - ii) 대형 개인방송 플랫폼과 협업을 통하여 접근성 향상.
 - iii) 개인방송 뿐만 아니라 빠른 하이라이트영상을 요구하는 스포츠 프로그램에 적용가능.
 - iv) 기존 모바일 어플리케이션에서 제한적이었던 긴 영상의 컷편집과정을 자동

- 화 함으로써 모바일 플랫폼에서 기존 영상편집 어플리케이션과 차별화.
- v) 유명 크리에이터들과의 협업을 통해 시청자에게 구독 크리에이터들에 대한 하이라이트 영상모음을 신속하게 제공하는 독자적인 플랫폼 구축.

- T(Threat):

- i) 신규 개인방송 플랫폼의 출현으로 인한 호환성 확보 요구.
- ii) 신조어 등 새로운 채팅데이터로 인한 지속적인 유지보수 필요.
- iii) 지속적인 업데이트를 위한 서버유지비용 소요발생.

4) 4P

- Product : 개인방송 콘텐츠 제작에 특화된 편집프로그램 제공
- Price : 기계학습의 특성상 지속적인 업데이트가 필요하고 주요 고객들의 가격에 대한 진입장벽을 낮추기 위하여 구독방식으로 판매. 프로그램 개발 비용과 운용 비용을 감안 하여 구독 가격 산정.
- Promotion : Google 애드센스를 통해 광고 진행. 인터넷 방송 영상편집 기초 영상을 제작하여 간접 광고.
- Place : 인터넷방송플랫폼 회사와 연계하여 B2B 판매 또는 직접 프로그램 판매 웹사이트 운영.

IV 팀 소개

Team

HighLive

HighLive는 Highlight와 Live를 합친 단어로, 실시간(라이브)방송의 하이라이트 구간을 추출한다는 주제에 맞도록, 그리고 HighFive를 연상시키는 단어로서 팀 합을 잘 맞춰보자는 의미에서 팀 이름을 선정하게 되었다.

- 3학년 김 산

담당분야 :

- RestController API설계 및 구현
- Crawler 파이썬 모듈 설계 및 구현
- Python모듈 실행 서비스 클래스 설계 및 구현

2017 창업동아리 디노 게임즈

2020 SW융합대학 창업동아리 드림챗

2021 임베디드 시스템 연구실

- 3학년 강정현

담당분야 :

- Javascript 차트 출력 설계 및 구현
- FindHighlight, Emotion 파이썬 모듈 설계 및 구현

2017 창업동아리 디노 게임즈

2020 SW융합대학 창업동아리 드림챗

2021 임베디드 시스템 연구실

- 4학년 허 진

담당 분야 : Spring 백엔드

- DomainController 설계 및 구현
- 캐시 구현
- Python모듈 실행 서비스 클래스 설계 및 구현

2020 임베디드 시스템 연구실

- 3학년 이원준

담당분야 :

- 일정 조율
- 비즈니스 요구사항 정리
- 회의록 및 보고서 작성

2017 창업동아리 디노 게임즈

V 프로젝트 세부사항

V-1 기술 개요

1) 개발 목표

i) 채팅트래픽 분석 :

채팅트래픽분석을 통한 하이라이트 구간을 제공한다.

ii) 메시지 자연어처리 :

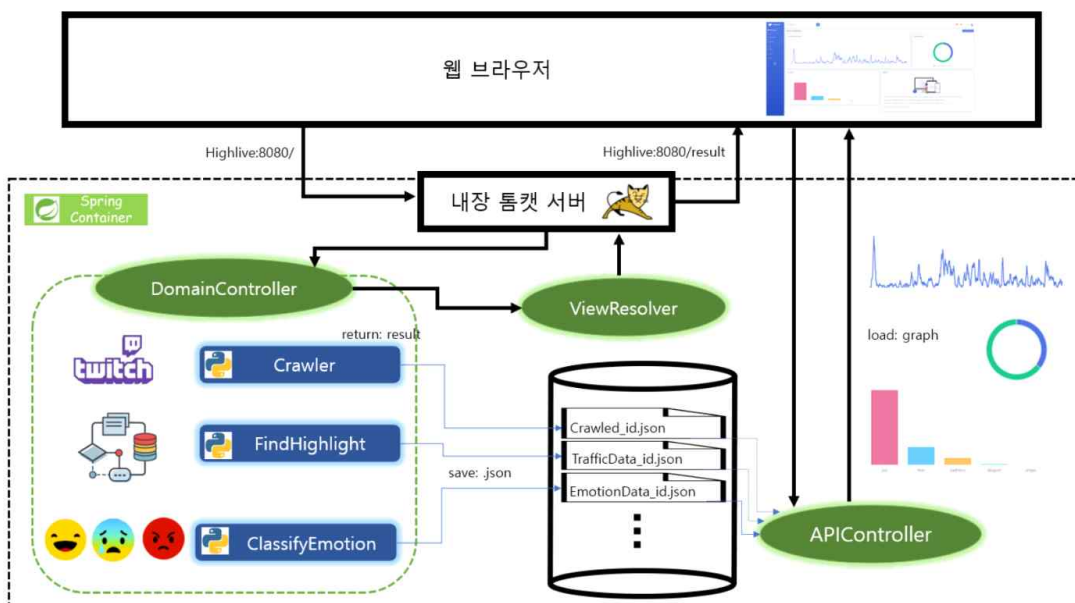
자연어 처리를 통한 데이터 전처리 하이라이트 구간 추출 정확도 향상 및 해당 구간의 감정 분석

iii) REST기반 웹 서비스 :

전문 편집자 뿐만 아니라 일반 사용자들의 진입장벽을 낮추고 접근성, 편의성을 높이기 위해 웹서비스로 제공

*사용자 친화적 UI구성

2) 설계 및 구현



사용자가 웹 브라우저를 통해 트위치 동영상의 URL을 입력하면 스프링의 내장 톱캣서버를 통해 이를 전달받아 파이썬 분석 모듈을 실행시킨다. 파이썬 모듈은 해당 영상에 대한 분석데이터들을 .json데이터 형태로 저장하고 결과 페이지를 출력한다. 결과페이지는 저장된 .json 파일을 통해 분석결과를 제공한다.

HighLive github 주소 : <https://github.com/HighLive/Highlive>

V-2 기술 상세

1) Python 모듈 구현

– crawler.py

응용 프로그램 총: 1			
		모두 보기	내 응용 프로그램 등록
이름	URL	제작 날짜	마지막 업데이트
HighLive	http://localhost	07/10/2020	11/15/2021

Twitch의 경우 twitch developers(<https://dev.twitch.tv/>)를 통해 API를 제공하고 있다. twitch developers에 Highlive 라는 이름으로 응용프로그램을 등록하고, OAuth access 토큰을 발급받아 Twitch API를 사용할 수 있도록 하였다.

The screenshot shows the Twitch Developers console interface. At the top, there's a navigation bar with links like '개요', '소셜 링크', '자문', '블로그', and '문서'. The main heading is '콘솔' (Console). Below it, there's a section for '응용 프로그램 관리: HighLive' (Application Management: HighLive). This section contains fields for '이름' (Name) set to 'HighLive', 'OAuth 리디렉션 URL' (OAuth Redirect URL) set to 'http://localhost', and 'Analytics Tool'. There's also a section for '클라이언트 ID' (Client ID) and '클라이언트 시크릿' (Client Secret). The interface is in Korean and includes various instructions and links for managing the application.

크롬의 개발자 도구의 Network 탭에서,
[https://api.twitch.tv/v5/videos/\(video_id\)/comments](https://api.twitch.tv/v5/videos/(video_id)/comments)
주소의 API를 통해 채팅을 비롯한 영상에 대한 데이터를 실시간으로 가져오는 것을 확인하였고, 이를 통해 채팅 데이터를 GET 방식의 형태로 요청할 수 있었다.

```
def getChatLog(video_id: str, client_id: str) -> list:
    next_cursor = ""
    params = {}
    params['client_id'] = client_id

    chat_log = []
    log_cnt = 0

    # 한번의 Get요청에 약 57 ~ 59개의 comment에 대한 json파일을 load한다
```

```

# next_cursor는 다음 채팅데이터를 가리키고 있음
while True:
    URL = "https://api.twitch.tv/v5/videos/" + \
        video_id + "/comments?cursor=" + next_cursor

    response = requests.get(URL, params=params)
    chat_json = json.loads(response.text)
    chat_log.append(chat_json['comments'])
    try:
        next_cursor = chat_json["_next"]
        log_cnt += 1
    except KeyError:
        break

return chat_log

```

받은 데이터에서 중복 사용자를 처리하기 위해 user_id, 시간당 올라오는 채팅 데이터를 카운트 하기 위해 time, 채팅 내용을 확인하여 감정분석에 활용할 수 있도록 content를 가져오도록 전처리 과정을 거쳐 (video_id).json 파일을 "./python/Data/traffic_data/" 의 경로에 저장하도록 하였다. 이후 생성된 json파일은 하이라이트 구간을 판단하기 위한 모듈인 'highlight.py'가 활용한다.

- highlight.py

'highlight.py'에서는 채팅 트래픽 추출 및 노이즈 제거, 피크 선정, 구간 설정의 네 단계를 거쳐 하이라이트 구간을 검출한다.

i) 'crawler.py'에서 생성한 json파일을 읽어와 초당 채팅 수를 카운트한다.

```

def check_frequency(json_data):
    n = int(json_data.popitem()[1]['time'])+1
    freq = [0]*n
    for chat in json_data:
        freq[int(json_data[chat]['time'])] += 1
    chat_count = pd.Series(freq)
    return chat_count

```

ii). 이후 검출 오류를 낮추기 위해 잡음 제거 필터를 적용하여 노이즈를 감소시킨다. 사용된 필터는 윈도우 크기가 7인 moving average 필터를 사용하였다.

```

def moving_average(chat_count, WIN=7):
    moving_avg = chat_count.rolling(WIN).mean()
    moving_avg[range(WIN-1)] = 0
    return moving_avg

```

iii) 채팅 트래픽이 몰려있는 피크를 찾기 위해 필터를 적용한 데이터를 내림차순으로 정렬한다.

```
def data_preprocessing(moving_avg, START=0):
    moving_avg = moving_avg.drop(range(1, START))
    processed_data = moving_avg.sort_values(ascending=[False])
    return processed_data
```

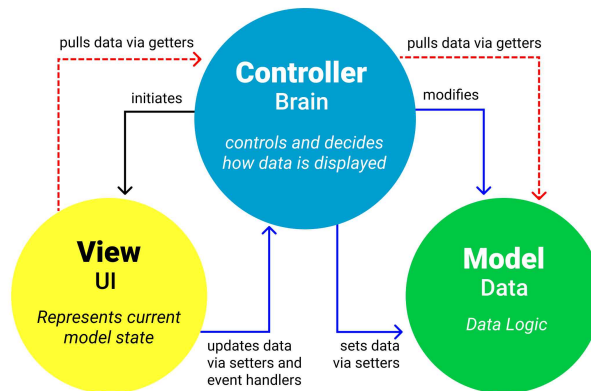
iv) 모든 피크타임에 각각 TERM(default: 10)초씩 구간을 설정한다. 내림차순으로 정렬된 피크를 하나씩 탐색해가며 하이라이트를 갱신한다. 이 때 구간이 겹치는 피크들은 합쳐서 하나의 구간으로 통합한다. 전체 하이라이트 길이가 미리 설정한 LENGTH를 넘어가는 시점에 탐색을 중단하고 찾아낸 하이라이트 구간과 전체 하이라이트 길이를 반환한다.

```
def find_highlight(data, TERM=10, LENGTH=60 * 10):
    # TERM: 단편 하이라이트의 최소 길이 (default: 10sec)
    # LENGTH: 전체 하이라이트 길이 (default: 10min)
    H = []
    hileng = 0
    num = 0
    while (hileng < LENGTH and num < len(data)):
        hileng = 0
        peaktime = int(data.index[num])
        H.append([max(peaktime - TERM, 0), peaktime])
        H.sort()
        try:
            for i in range(0, len(H)):
                while (True):
                    if H[i][1] >= H[i + 1][0]:
                        if H[i][1] <= H[i + 1][1]:
                            H[i] = [H[i][0], H[i + 1][1]]
                            del H[i + 1]
                        else: break
                    else: break
        except IndexError:
            for t in H:
                hileng += t[1] - t[0]
            num += 1
    return H, hileng
```

2) Spring Backend 구현

본 웹서비스는 스프링 MVC패턴을 활용하였다. MVC패턴은 디자인 패턴중 하나로 Model, View, Controller의 약자이다.

MVC Architecture Pattern



MVC Pattern(www.freecodecamp.org)

위의 그림과 같이 사용자가 controller를 조작하면, controller는 model을 통해 데이터를 가져오고 그정보를 바탕으로 시각적인 표현을 담당하는 View를 제어하여 사용자에게 전달 한다.

컨트롤러는 데이터와 사용자 인터페이스를 잇는 다리역할로, 사용자 발생시키는 이벤트를 처리하는 역할을 담당한다.

뷰는 객체의 입출력을 담당하며 본 프로젝트에서 활용한 URL입력폼과 같은 인터페이스 요소를 나타낸다.

모델의 경우 데이터 로직을 나타내며 본 프로젝트에서는 Model을 통해 사용자 인터페이스로 videold를 전달하도록 구현하였다.

- Controller설계

```
@Controller
public class DomainController {
    private final InputHandler inputHandler;

    @Autowired
    public DomainController(InputHandler inputHandler) {
        this.inputHandler = inputHandler;
    }

    @GetMapping("/")
    public String index() {
        return "index";
    }

    @PostMapping("/result")
    public String result(HttpServletRequest request, Model model) {
```

```

String url = request.getParameter("url");
StringTokenizer st = new StringTokenizer(url, "/");
String video_id = "";

List<String> pstr = new ArrayList<>();
while(st.hasMoreTokens()) {
    pstr.add(st.nextToken());
}

boolean validCheck = false;
try {
    for(int i=0; i<pstr.size(); i++){
        if(pstr.get(i).equals("videos")) {
            video_id = pstr.get(i+1);
            validCheck = true;
            break;
        }
    }
} catch(IndexOutOfBoundsException e) {
    return "invalid_videoId";
}

if(!validCheck){
    return "404";
}

if(!inputHandler.checkCache(video_id)) {
    // 캐시 미스
    System.out.println("cache miss! ");

    // 초기화
    inputHandler.runInit(video_id);
    // 크롤링
    inputHandler.runCrawler(video_id);
    // 감정분석
    inputHandler.runClassifyEmotion(video_id);
    // 트래픽 추출
    inputHandler.runFindHighlight(video_id);
}
else
    System.out.println("cache hit!");

model.addAttribute("videoId", video_id);
return "result";
}
}

```

template폴더의 html파일을 보여주기 위한 DomainController와 프론트에서 json파일을 요청하여 접근할 수 있도록 하기 위한 ApiController로 나누어 설계하였다.

DomainController의 경우 처음 highlive의 주소로 접근하였을 때 보여줄 index페이지를 Get요청의 반환값으로 처리하고, 상단 URL입력폼에 트위치 영상주소를 입력

하였을 때의 Post요청을 받아 영상의 데이터를 크롤링하고 하이라이트 구간에 대한 분석을 진행하는 파이썬 모듈들을 실행한다.

- ApiController설계

```
@RestController
public class ApiController {

    static final String dataPath = "./python/Data/";
    @RequestMapping(value = "/api/traffic/{id}", method = RequestMethod.GET)
    public String getTraffic(@PathVariable("id") String videoId) throws IOException {
        File resource = new File(dataPath + "traffic_data/" + videoId + ".json");

        return new String(Files.readAllBytes(resource.toPath()));
    }

    @RequestMapping(value = "/api/emotion/{id}", method = RequestMethod.GET)
    public String getEmotion(@PathVariable("id") String videoId) throws IOException {
        File resource = new File(dataPath + "emotion_data/" + videoId + ".json");

        return new String(Files.readAllBytes(resource.toPath()));
    }

    @RequestMapping(value = "/api/highlight/{id}", method = RequestMethod.GET)
    public String getHighlight(@PathVariable("id") String videoId) throws IOException {
        File resource = new File(dataPath + "highlight_data/" + videoId + ".json");

        return new String(Files.readAllBytes(resource.toPath()));
    }

    @RequestMapping(value = "/api/valid/{id}", method = RequestMethod.GET)
    public String getValid(@PathVariable("id") String videoId) throws IOException {
        File resource = new File(dataPath + "valid_data/" + videoId + ".json");

        return new String(Files.readAllBytes(resource.toPath()));
    }

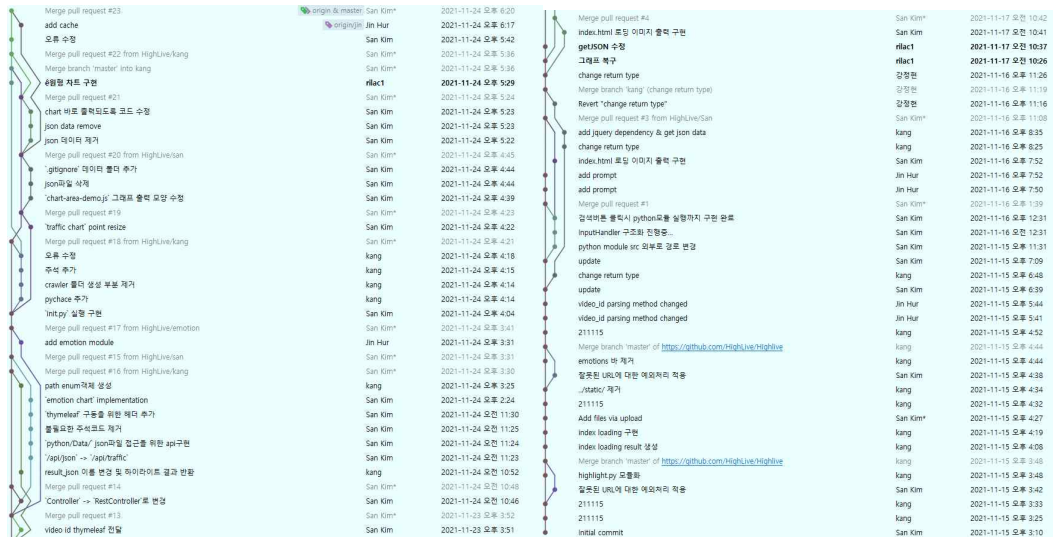
    @RequestMapping(value = "/api/raw/{id}", method = RequestMethod.GET)
    public String getRaw(@PathVariable("id") String videoId) throws IOException {
        File resource = new File(dataPath + "raw_data/" + videoId + ".json");

        return new String(Files.readAllBytes(resource.toPath()));
    }
}
```

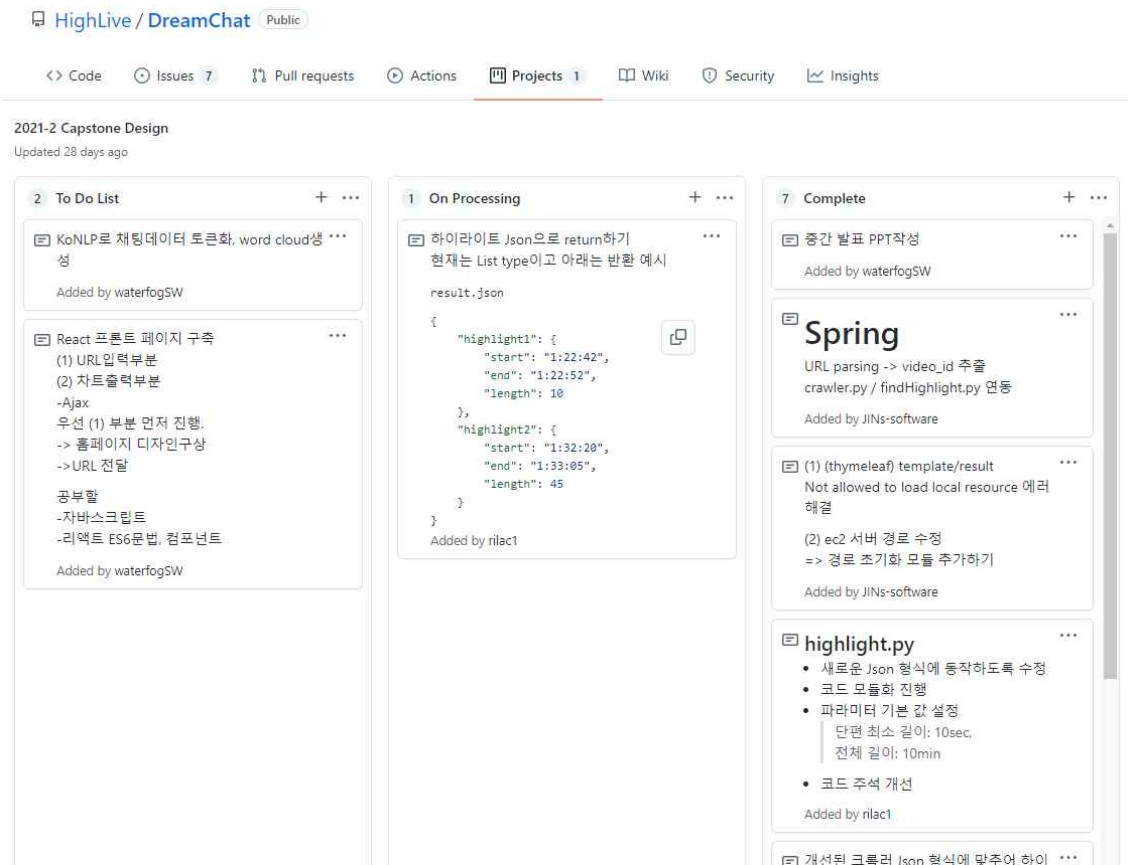
앞선 DomainController에서는 View를 반환하기 위해 @Controller라는 어노테이션을 사용하였다. 반면 ApiController에서는 Json 형태로 객체 데이터를 반환하기 위해 @RestController라는 어노테이션을 사용하였다.

Api Controller는 프론트 페이지가 로드 될 때 Javascript코드가 파일시스템 상에 저장되어 있는 트래픽, 하이라이트 구간 및 감정분석 데이터에 대한 json파일을 Get요청을 통해 읽어올 수 있도록 하는 Api를 제공한다.

V-3 협업 로그



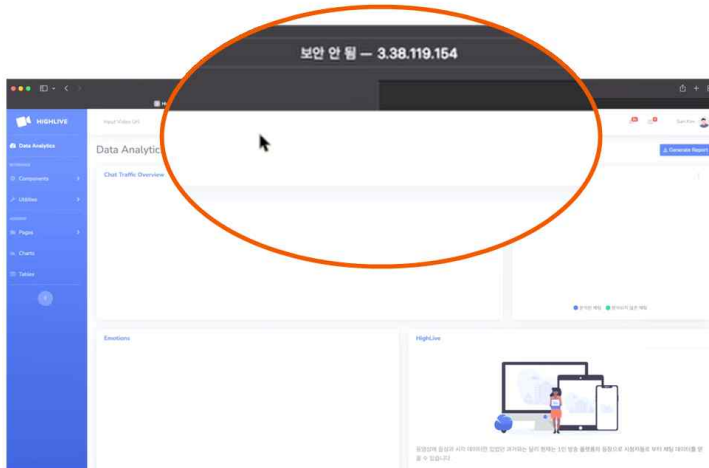
<그림 1. 깃허브 commit 및 pull request 로그>



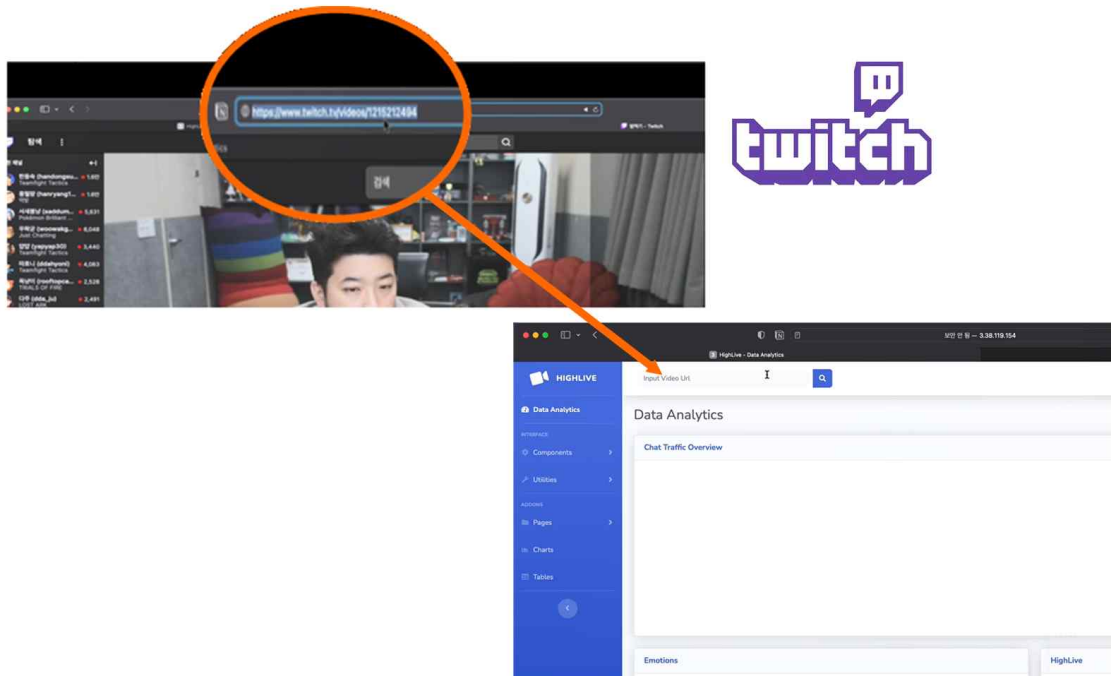
<그림 2. 깃허브 Project 패널 활용>

VI 결과

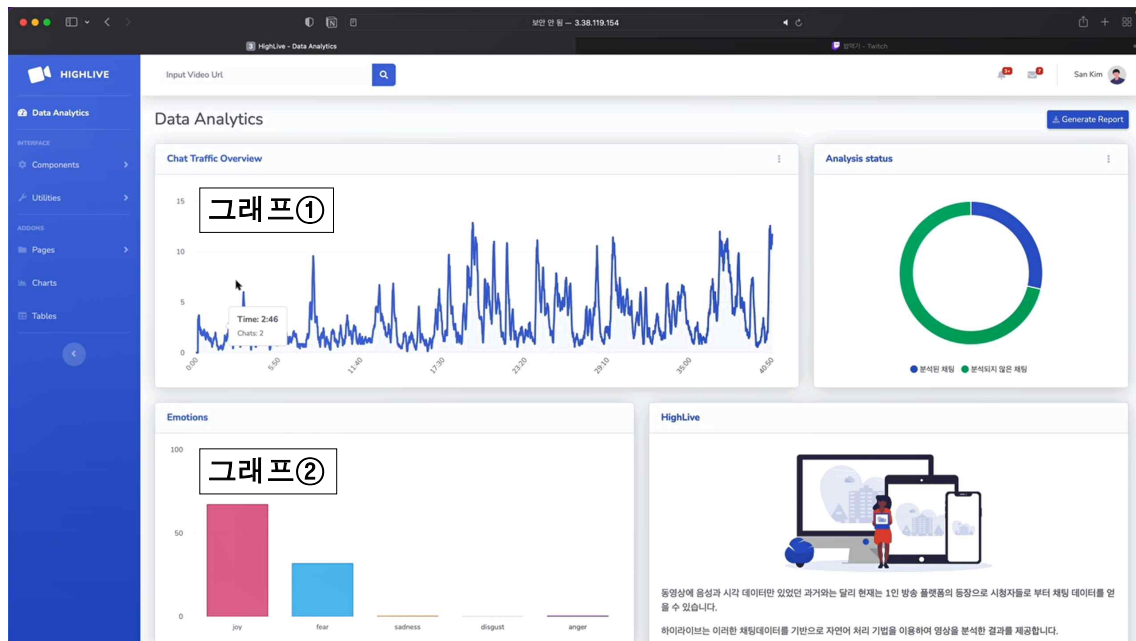
1) <http://3.38.119.154:8080/> 접속



2) 영상 URL 입력



3) 결과화면



- i) **그래프①**의 가로축은 시간, 세로축은 채팅 트래픽량이다. 그래프상에서 고점을 찍는 구간이 하이라이트 구간이다.
- ii) **그래프②**는 감정분석 그래프이다 채팅데이터를 기반으로 joy, fear, sadness, disgust, anger의 감정을 나타낸다.

VII 향후계획

- 1) 현재 구현된 감정분석 방식에 채팅에 최적화된 자연어처리 모델을 구현하여 보다 정확성을 높일 예정.
- 2) 채팅의 트래픽뿐만 아니라 채팅의 콘텐츠까지 분석하여 차별성 강화.
- 3) 하이라이트로 추출된 구간을 모아 최종 편집본을 사용자에게 제공하여 자동 편집 서비스 제공.
- 4) 현재는 Twitch 플랫폼에만 적용됨. Youtube, AfreecaTV등 다양한 플랫폼에서 활용할 수 있도록 제작.

VIII 기술적 위협요소 분석

- 1) crawler.py파일에 트위치 api를 활용하기 위해 발급받은 보안 토큰이 코드에 노출되어있다.

=> 보안 키 자체를 다른 방식으로 암호화하여 이를 형상서버에 등록해 주고, 이를 암호화된 환경 변수로 제공하여 처리하는 방식으로 추후 개선이 필요하다.

- 2) 트위치 API에 종속적

트위치에서 자체적으로 api접근 방식을 변경한다면 전반적인 코드 수정이 불가피하므로 유지보수에 어려움이 있을뿐더러 약관이 변경될경우 전반적인 서비스 운영이 불가능 할 수도 있다.

=>API를통한 채팅데이터 추출 외에 beautiful soup를 이용한 크롤링 방식 등 채팅 데이터 접근 방식을 다각화 하여 안정성을 높이고, 다양한 플랫폼에서의 지원을 통해 위협 요소를 분산시킨다.

- 3) 현재는 파일시스템에 데이터 분석결과를 저장하는데 이는 데이터 중복문제를 발생시킬 수 있고, Disk I/O를 증가시킨다.

=> NoSql데이터 베이스를 도입하여데이터 중복을 최소화 하고, 인기있는 영상에 대한 데이터는 redis와 같은 인메모리 데이터 베이스를 사용하여 disk I/O를 최소화 한다.

- 4) 입력폼을 통해 입력된 URL을 통해 video_id를 추출하는데 이는 코드 인젝션 공격에 취약할 수 있다. =>URL입력의 제한조건을 강화하여 이러한 공격을 예방한다.

IX 회의록 및 작업 후기

1) 회의록

9/13(월)

장 소	Zoom을 통한 비대면 회의
회의 주제	프로젝트 주제 선정 및 팀 발표 ppt준비
주요 내용	프로젝트 주제 → 트위치 채팅데이터 분석을 통한 하이라이트 구간 추출 팀 발표 준비 → 구글문서 온라인 편집기를 이용하여 ppt 공동 제작

9/29(수)

장 소	ICT관 5층 테이블
회의 주제	주제에 대한 세부적인 프로세스 설계, 역할분배
주요 내용	세부적인 프로세스 설계 → 크롤링모듈, 하이라이트 추출 모듈, 프론트엔드로 세분화 → '하이라이트'에 대한 기술적 정의 설정 역할 분배 → 허진 학생이 Spring 백엔드파트를 담당하기로 함 → 김산 학생이 데이터 크롤링, 백엔드 파트를 담당하기로 함. → 강정현 학생이 데이터 분석, 백엔드 파트를 담당하기로 함. → 이원준 학생이 React 프론트엔드파트를 담당하기로 함

10/6(수)

장 소	소프트웨어학과 과방
회의 주제	10/13 팀별 미팅 준비
주요 내용	현재 진행중인 사항 종합 → ppt제작

10/20(수)

장 소	Zoom을 통한 비대면 회의
회의 주제	문제점 해결
주요 내용	→ Twitch API에서 Get 요청시 받아올 수 있는 .json파일을 처리하는 방법에 대해 논의 → URL입력에서 차트 출력까지 걸리는 시간을 최소화 하기 위한 크롤링 모듈 수정 방법 논의.

10/27(수)

장 소	Zoom을 통한 비대면 회의
회의 주제	진행상황 점검
주요 내용	현재 진행상황 점검 및, 향후 개발 일정 구체화. 채팅데이터를 분석하여 감정분석기능을 추가하기로 함.

11/10(수)

장 소	학교 앞 카페
회의 주제	프로젝트 발표 점검
주요 내용	프로젝트 발표 점검 → 김산 학생의 주도 하에 최종 발표를 자체적으로 진행해 보고 피드백을 주고받음. → 발표 시간에 맞추기 위해 주요 내용만 정리하는 방향으로 나아가도록 하였고, 발표에 담지 못한 내용들은 최종 보고서에 포함하도록 함. → 감정분석기능 개발방법 아이디어 토의.

11/24(수)

장 소	Zoom을 통한 비대면 회의
회의 주제	졸업작품전시회 제출자료 준비
주요 내용	졸업작품 전시회 제출 영상 제작 프로젝트 이름을 변경할지에 대한 여부 토의 → 기존 이름대로 진행하기로 함. 대신 향후 계획에 대한 내용을 구체화 시켜서 정리.

12/01(수)

장 소	Zoom을 통한 비대면 회의
회의 주제	최종보고서 준비 및 발표 점검
주요 내용	최종보고서 문서, 발표용 PPT제작, 발표 점검하며 피드백 주고받음.

2) 작업 후기

작업 후기	
학번 / 이름	32170578 김산
<p>작업 후기 - 김산</p> <p>효율적으로 협업하는 방안에 대해 많이 고민하고 배울 수 있었습니다. 프로젝트 아이디어는 제가 기존에 SW 중심대학 사업단 창업동아리를 진행하면서 냈던 아이디어와 사업기획서로부터 가져왔습니다. 각자 강점이 있는 분야가 조금씩 겹쳤기에 역할을 분배하면서 어려움을 겪기도 했습니다. 하지만 각자 맡은 부분에 대해 의견을 나누고 역할을 잘 조율하고 나니 혼자 할 때 보다 더 작업이 빨리 진행됨을 느꼈습니다.</p> <p>또 깃을 적극적으로 활용하면서 commit메시지와 pull request메시지를 참고 하면서 각자 진행하는 부분들을 실시간으로 공유하고, 마지막에 merge하면서 하나하나 맞아 들어가는 과정에서 큰 재미를 느꼈습니다.</p> <p>서비스를 기획하고 구현하여 배포하는 모든 과정을 경험해 보았던 첫 협업 프로젝트였습니다. 기술 스택으로 Spring 프레임 워크를 선정하여 진행하였는데, node-js와 같이 진입장벽이 낮은 프레임 워크를 사용하기보다는, 공부할 게 많고 복잡한 프레임 워크를 사용하면 더 많이 배울 수 있겠다는 이유에서였습니다. Spring에서 python모듈을 실행시킬 때 멀티 스레드로 실행시키면 시간을 더 절약할 수 있지 않을까, 그렇게 구현 했을 때 스레드 간 자원 공유 문제를 어떻게 해결할 것인가에 대한 이야기도 나누면서 운영체제 과목에서 배웠던 지식을 활용하기도 하였습니다. 또 동료들이 제가 작성한 코드를 쉽게 이해할 수 있도록 명명 규칙과 현업에서의 코드 작성방식을 찾아보면서 디자인 패턴에 대한 공부도 할 수 있었습니다.</p> <p>프론트 페이지를 구성하면서 평소에 관심을 크게 두지않았던 자바스크립트도 조금씩 공부해 보면서 프론트 엔드라는 다른 도메인의 지식도 쌓을 수 있었습니다. 또 백엔드에서 프론트엔드로, 프론트엔드에서 백엔드로 데이터를 전달하는 로직을 구현하면서 백엔드와 프론트의 역할에 대해 고민하고 배울 수 있었습니다.</p> <p>이번 캡스톤 디자인을 진행하면서 개발과 협업과정에 많은 흥미를 느꼈고, 덕분에 제 진로에 대해서도 확신을 가질수 있었습니다. 프레임 워크를 활용하고, 깃을 활용하는데 부족함을 느끼기도 했기에 이러한 부분을 보완하면서 새로운 프로젝트들을 이어나가고 싶습니다.</p>	

작업 후기

학번 / 이름

32170078 강정현

학교 생활 중에 다양한 프로젝트를 해보았지만 항상 마무리를 제대로 짓지 못하고 흐지부지되는 경우가 부지기수였습니다. 그래서 이번 캡스톤 디자인 수업에서는 꼭 결과물을 내고 싶었고 교수님의 훌륭한 지도와 열정적인 팀원들 덕에 잘 마무리 할 수 있었던 것 같습니다.

이번 캡스톤 디자인 수업은 기술적인 역량뿐만 아니라 협업과 관련한 능력을 기르는 데 큰 도움이 되었습니다. 팀원 개개인이 중요시하는 부분이나 잘하는 부분이 모두 다르기 때문에 지속적인 커뮤니케이션을 통해 프로젝트를 조율해 나가는 것이 완성도 있는 결과물을 만들어 내는 방법이라고 느꼈습니다. 다행히도 팀원 모두가 열정적이고 서로 마음도 잘 맞아서 큰 의견 차이 없이 프로젝트를 진행하였고 서로에게 부족한 부분이 발생하면 같이 나서서 문제를 해결해나갈 수 있었습니다.

최종 발표 전 교수님과의 미팅에서는 교수님께서 해주신 말씀이 큰 도움이 되었습니다. 미팅을 준비했었던 시점에서는 프로젝트를 처음 기획했을 때 생각했었던 기능들을 많이 구현해내지 못한 상태였습니다. 따라서 ‘이것을 다 할 수 있을까?’라는 생각을 자연스레 하게 되었고 결국 마음만 앞서고 결과물은 따라오지 않았던, 지금껏 제가 해온 프로젝트들 처럼 급하게 마무리를 해야 할 수도 있는 상황이었습니다. 하지만 교수님께서서는 적어도 저희가 내세우는 핵심적인 기능은 구현을 해야 의미있는 결과물이라고 말씀해주셨습니다. 거기에 더해 지금 단계에서는 각각의 완성도에 힘을 쏟기보다는 지금까지 해왔던 것들을 잘 매듭지어서 결과물을 만드는데 집중해야 한다는 말씀을 해주셨습니다. 이러한 피드백을 받고 저희는 구현을 간소화하되, 핵심적인 기능은 반드시 구현해내야겠다는 목표로 매달렸고 결국 만족스러운 결과물을 뽑아낼 수 있었습니다.

이번 프로젝트를 진행하면서 정말 많은 것을 경험할 수 있었습니다. 알게나마 직접 서비스를 만들어 배포까지 해보면서 제가 부족한 점들을 찾을 수 있었고 앞으로 어떤 식으로 공부하면 좋을지 등을 생각해보게 되었습니다. 앞으로 이러한 프로젝트를 계속해서 진행해 나갈 예정이고 이번 수업에서 교수님이 해주신 말씀들을 잘 기억했다가 필요한 순간에 잘 복기하도록 하겠습니다. 감사합니다.

작업 후기

학번 / 이름

32174937 허진

이번 프로젝트를 진행하며 크게 두 가지를 얻을 수 있었습니다. 첫 번째는 실제 응용 서비스를 제작하며, 요구 분석에서 설계, 구현 그리고 배포까지의 경험을 쌓을 수 있었고, 두 번째는 팀원들과의 협업을 통해 의사소통과 효율적 작업 방식의 중요성을 깨달은 것입니다.

이전 캡스톤의 경우 논문 작성 형식의 프로젝트를 진행하였기에, 응용 프로그램과 서비스 구축의 경험을 이번 학기에 처음으로 해보았습니다. 요구 분석에서부터 설계 그리고 구현까지 소프트웨어 개발 사이클에 참여할 수 있었고, 이를 통해 사용자 측면에서 서비스를 바라보는 관점을 수립하는 것부터 시작해, 소프트웨어 아키텍처 설계, 설계를 기반으로 한 구현까지 팀원들과 함께 진행하며 매 순간 색다른 경험과 교훈을 얻을 수 있었습니다.

프로젝트의 첫 계획은 팀원이 제시한 기발한 아이템과 서비스 내용을 바탕으로 시작하였습니다. 하지만 초기 계획을 바탕으로 막연히 주먹구구식으로 진행하였더니 일부 모듈(기능적 구성요소)들을 만드는 데에 기반 지식이나 구현 역량의 부족으로 막혀버리는 일이 발생하였습니다. 물론 프로젝트를 진행하는 과정 동안 학습하며 실현할 수 있는 부분도 존재하였으나, 자연어 처리 기술 도입에 있어선 원하는 수준의 기능 동작까지 시간적, 인력적 한계가 있다는 것을 파악하였습니다. 따라서 저희는 서로가 클라이언트이면서 동시에 분석가, 설계자가 되어 요구 수준 분석과 실현할 수 있는 수준의 설계를 진행하였습니다. 요구 사항을 잘 정립하는 것도 중요하지만, 한정된 인력, 기간뿐 아니라 실현할 수 있는 기술을 바탕으로 프로젝트의 이해당사자 모두를 만족시킬 수 있는 공통의 목표를 추출하는 것 또한 중요하다 생각되었기 때문입니다. 따라서 선택과 집중이란 전략을 택하여 반드시 구현해야 할 핵심적 기능 구현을 더 확실히, 견고히 할 수 있었습니다. 결과적으로 요구되는 서비스에 대해선 확실히 구현할 수 있었습니다.

설계 과정에선 초기 분석 단계에서 고려하지 않았던 상세 내용을 충분히 반영하여 구현할 수 있도록 준비하는 과정이었고, 웹 서비스에서 대표적으로 사용하는 MVC 아키텍처 모델에 대한 학습, 시스템 분할을 통한 각 기능 모듈화, 인터페이스 정의 등과 같은 설계 과정을 통해 이후 진행하는 구현과정에서 효율적인 작업을 할 수 있었습니다. 특히 시스템 분할을 통한 각 기능의 모듈화를 진행하여 팀원들이 작업 분담으로 구현을 수행하며 효율적으로 진행할 수 있었습니다. 나아가 특정 모듈을 추가하거나 삭제하는 과정, 인터페이스 수정의 작업에서도 설계도를 바탕으로 한눈에 파악하며 손쉽게 진행하는데 도움을 받을 수 있었습니다.

구현에 있어서는 파이썬을 통해 각 기능 모듈을 구현하는 것부터 MVC 아키텍처를 기반으로 한 클래스의 동작 프로세스를 공부하며 백엔드를 구축하는 경험을 할 수 있어서 좋았습니다. 이 과정에서 스프링 플랫폼을 알게나마 공부하며 필요한 부분 부분을 찾아가며 구축하였고, 이를 통해 실제 구현한 서비스가 웹 페이지에서 작동하는 것에 흥미로웠습니다.

마지막으로 협업에 있어서 기존에 익히 사용하지 않았던 깃과 깃허브 저장소를 활용하며 팀원들과 협업하는 경험을 할 수 있어서 좋았습니다. 혼자서는 힘들 작업을 작업 분담을 통해 진행하니 시간이 많이 절약되었고, 구현과정에서 실시간으로 커밋 메시지를 전달하고, 확인하며 진행해보니 효율적인 작업이 진행되었다는 장점도 발견하게 되었습니다.

작업 후기	
학번 / 이름	32173243 이원준
<p> 하이라이브 팀 프로젝트는 팀원의 기술적인 역할 분배가 확실하게 이루어 졌고, 회의 때마다 진행상황을 종합하고, 방향성을 수정해가며 진행되었습니다. 저는 프론트엔드를 담당했습니다. 처음에는 보다 동적인 그래프를 그리고자 자바스크립트 라이브러리인 리액트를 이용하여 그래프 구현을 목표하였지만, 회의를 거치면서 굳이 리액트까지 사용할 필요가 없다고 판단하여, REST 기반으로 웹을 개발하는 것으로 변경하게 되었습니다. 깃허브를 사용해서 협업을 하는 것에 잘 익숙하지 않아서 어려움을 겪기도 했습니다. 그렇지만 어려운 부분이 발생해서 막힐 때마다 팀원들이 잘 도와주고 저를 이끌어 주었고, 이런 좋은 팀원들 덕분에 만족스러운 결과물이 나온 것 같습니다. 다른 팀들도 마찬가지겠지만, 회의를 일부 비대면으로 진행하게 됨에 따라, 회의 효율이 떨어지는 것 같아 불편하기도 했지만, 그럼에도 팀원들이 적극적으로 회의에 참여했기 때문에 비대면의 불편함도 상대적으로 극복할 수 있었고, 각자 열정적으로 임했기에 회의를 하는 과정 자체도 재미있게 했던 것 같습니다. 프로젝트를 진행함으로써 자바스크립트와 그 라이브러리에 대한 이해도를 넓힐 수 있는 기회가 되었습니다. 본 프로젝트는 웹 서비스이고, 주요 아이템은 하이라이트 추출이지만, 이외에도 웹 페이지에 들어갈 기능을 다양화시켜 사용자에게 보다 다양한 서비스를 제공함으로써 클라이언트의 접근성을 높이고자 하는 계획을 갖고 있습니다. </p>	