

2021학년도 1학기 종합설계 교과목 캡스톤 디자인 결과보고서

소 속 (전공)	디지털콘텐츠학과	
팀 명	용감한쿠키	
지 도 교 수	권 순 일 (인)	
팀 장	학번: 15011187	이름: 한재원
팀 원	학번: 15011163	이름: 강전호
	학번: 18011813	이름: 김수지
제 출 일 자	21 . 6 . 18	

세종대학교 공학교육센터

목 차

제 1 장 종합설계 개요

1. 종합 설계 제목
2. 설계의 개요(Abstract or Concept)
3. 설계의 배경 및 필요성

제 2 장 설계의 현실적 제한 조건 기술

1. 현실적 제한 조건과 이에 따른 고려 내용의 기술

제 3 장 설계 구성 요소에 따른 결과 기술

● 설계의 구성 요소 체크 항목

1. 목표 설정

- 1.1 문제 해결을 위한 아이디어 및 구체적인 방법
- 1.2 수행목표

2. 합성

- 2.1 기초 조사
- 2.2 개념의 합성(개념 설계)

3. 분석(작품 구현 과정 중의 문제점 분석 및 해결 방법)

- 3.1 과제수행에 사용된 이론 및 기술의 조사 및 분석 결과
- 3.2 설계물에 대한 분석 및 보완

4. 제작

- 4.1 완성품 제작 결과 (사진)
- 4.2 완성품 설명
- 4.3 작품 제작 과정 정리
- 4.4 작품의 특징 및 종합설계 수행 결론
- 4.5 완성품의 사용 매뉴얼

5. 시험 (시험 결과 기술)

5.1 최종 결과물에 대한 시험 결과

6. 평가

6.1 작품의 완성도 및 기능 평가

6.2 기대효과 및 영향

6.3 작품제작 후기

6.4 팀 개요 및 역할분담

6.5 참고문헌

[첨부 1] 작품 사진 첨부

볼드체의 각장과 각절의 제목은 되도록 유지하여 주시기 바라며 그 밖의 소제목은 지도 교수의 지도하에 가감 또는 조정이 가능합니다.

제 1 장 종합설계 개요

1. 종합 설계 제목

제목 : Meeting transcripts with emotional situation

(부제 : 실시간 감정 분석 화상 회의 및 감정 상황 회의록 서비스)

2. 설계의 개요(Abstract or Concept)

2.1 설계의 요약

코로나 장기화로 인한 비대면 협업 툴의 수요가 증가했습니다. 그러나 기존의 화상 회의 서비스는 IT 산업의 발전에 비해 여전히 과거에 머물러 있다고 느꼈습니다. 따라서 보다 전문적이고 체계적인 화상 회의 및 회의록 서비스를 개발하고자 하였으며, 음성을 통한 감정 인식 결과를 제공하여 텍스트로만 이루어진 회의록에 비해 현장감이 있는 회의록을 제공할 수 있는 프로젝트를 기획했습니다.

2.2 현황 조사 및 경쟁 서비스 분석

2.2.1 현황 조사

웍스 모바일이 발표한 바에 따르면, 라인웍스에서 제공하는 화상회의 기능 사용량은 다자간 영상통화는 28배, 음성통화는 25배, PC화면 공유는 15배 이상 상승했다고 합니다



그림 1) 라인웍스 화상회의

이처럼, 길어지는 재난 상황으로 인해 비대면 비즈니스 툴 앱, 특히 화상회의 기

능에 대한 수요가 높아졌습니다. 실제로 화상회의 기능을 지원하는 앱의 사용자는 지원하지 않는 앱에 비해 사용률이 7배 이상 많은 것을 알 수 있습니다. 또한 원격 및 재택근무가 확산됨에 따라 일하는 방식이 변화해, 코로나19 상황이 진정된 후에도 여전히 업무용 협업 툴의 수요는 높을 것으로 예상됩니다. 따라서, 쉽고 간편하게 사용 가능하며, 비대면 업무에 최적화된 화상 회의 및 회의록 서비스가 필요한 상황입니다.



그림2) 화상회의의 증가 추이

그러나 화상 회의의 가장 큰 단점은 집중력 약화입니다. 한 조사에 따르면, 직장인들은 화상회의를 진행하는 동안 다른 일을 하거나, 이메일을 보내는 등 다른 일에 쉽게 주의를 팔리는 것으로 나타났습니다. 따라서 참여를 이끌어낼 수 있는 회의 설계가 필요합니다.

2.2.2 경쟁 서비스 분석

FLOG가 중점적으로 제공하는 음성 감정 인식과 회의록 자동 텍스트화 기능에 있어서 경쟁 업체들을 선정하고 분석하였습니다.



그림3) AI 음성 인식 받아쓰기 서비스 '다글로'

다글로는 음성 파일, 동영상 파일, 유튜브 링크의 음성을 텍스트로 변환하는 서비스입니다. 화자 분리 기술이 적용되어 있으며, 결과물은 자막 파일, 텍스트 파일 등의 형식으로 다운로드하여 사용할 수 있습니다. 그러나 실시간 서비스가 아닌 녹음 파일을 업로드 해야 하는 한계가 있습니다.

음성인식 회의록 시스템

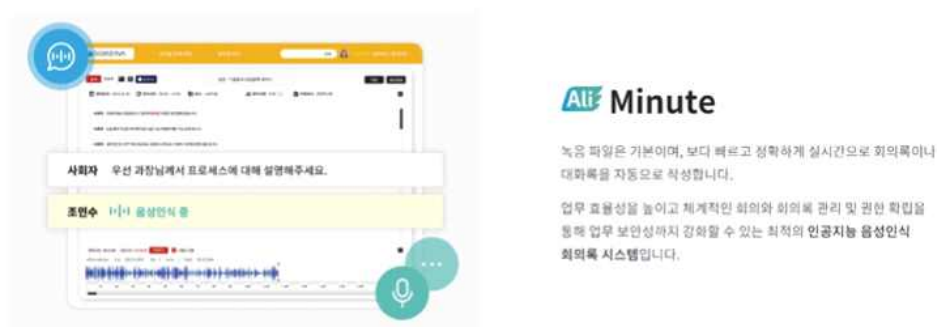


그림4) 음성인식 회의록 시스템 'Minute'

녹음파일은 물론, 실시간으로 회의록이나 대화록을 자동으로 작성하는 서비스입니다. 소리자바 Ali STT 기술을 바탕으로 체계적인 회의록을 작성할 수 있어 업무 효율성을 높일 수 있습니다. 다자간 음성인식을 지원합니다.



그림5) 음성 감정 분석 AI 서비스 'Empath'


Empath는 어떠한 언어에서도 기쁨, 분노, 고요함, 슬픔 중 하나의 감정을 실시간으로 자동 감지할 수 있습니다. 또한 피치, 톤, 스피드 등의 특성을 찾아냅니다. Empath의 감정인식 AI는 재난 상황에서 피해자의 정신 상태를 감지하여 도움을 주는 목적으로 이용되고 있으며, 콜센터 등 고객의 얼굴을 볼 수 없는 비즈니스 현장

에서도 활용되고 있습니다.

2.3 차별성

경쟁 서비스를 분석한 결과 화자를 인식하고 음성을 텍스트화 해주는 서비스는 화상 회의를 함께 제공하거나 이에 대한 감정 인식 결과를 제공하지 않았으며, 감정 인식 AI 서비스를 제공하는 경우 음성을 텍스트화 하여 회의록으로 제공하지 않았습니다.

FLOG는 실시간 화상 회의에서 화자를 인식하고 모든 발언을 텍스트화 하여 저장할 뿐 아니라 음성에 대한 감정 분석 결과를 제공하며, 회의록에도 감정 인식 결과를 저장하여 사용자에게 제공함으로써 차별성을 두었습니다. 또한 이외에도 회의록 요약 서비스 및 음성 TTS를 제공하여 회의록에 보다 중점을 두어 회의에 참여하지 못한 참여자가 회의록만을 보고도 당시의 분위기를 생생하게 느낄 수 있도록 하였습니다.

				
실시간 화상회의	지원	지원	미지원	미지원
회의록 저장	지원	미지원	지원	미지원
화자 인식	지원	지원	지원	지원
음성 텍스트화	지원	미지원	지원	미지원
감정 분석	지원	미지원	미지원	지원

3. 설계의 배경 및 필요성

3.1 실시간 회의 설계의 배경 및 필요성

Microsoft의 Human Factors Lab의 연구 결과에 따르면 화상 회의 시 스크린에 집중해 비언어적 단서를 찾는 행동이 대면 회의에 비해 스트레스를 높인다고 발표했습니다. 따라서 상대방의 감정을 인공지능을 통해 분석해서 시각화 하여 제공한다면 회의의 집중도를 높일 수 있을 것으로 판단했습니다. 또한 화상 회의 화면 내에 사용자의 참여도 순위를 함께 제공한다면 참여자가 경각심을 가지고 더욱 열심히 참여하게 만들 수 있는 요인이 될 수 있을 것으로 생각했습니다.

3.2 감정 회의록 설계의 배경 및 필요성

단순히 음성을 텍스트화 한 결과를 회의록으로 대신하기에는 부족한 점이 많습니다. 정리되지 않은 불필요한 내용이 모두 회의록에 저장되므로 요점을 파악하기 위해, 또는 중요한 부분만을 다시 확인하고자 할 경우 모든 내용을 살펴보아야 하기 때문에 시간이 낭비됩니다.

또한 회의에 부득이하게 참여하지 못한 팀원이 회의록을 통해 회의 내용을 따라가게 될 경우, 텍스트로만 이루어진 회의록을 통해서는 전체적인 회의의 분위기나, 결정된 사안이 큰 불화 없이 모든 팀원이 동의해서 나타난 결과인지 등을 파악하기 어렵습니다. 이외에도 프로젝트의 책임자가 진행 상황, 참여도 등을 확인하기 위해 회의록을 열람하게 될 경우, 모두 적극적으로 회의에 참여했는지를 객관적으로 판단할 수 있어야 합니다.

따라서 FLOG는 단순히 음성을 텍스트화 하는 것에 그치지 않고, 많이 등장한 단어들을 통해 워드 클라우드를 형성하고 요약문을 함께 제공함으로써 회의의 요점을 빠르게 파악할 수 있는 서비스를 개발하고자 했습니다. 더불어 감정 분석 그래프를 통해 회의의 분위기를 파악하고 다음 회의에서의 개선점을 찾아 나갈 수 있는 방안을 마련하고자 했으며, 참여도 랭킹을 통해 사용자의 회의 참여 적극도를 파악할 수 있도록 했습니다.

제 2 장 설계의 현실적 제한 조건 기술

1. 현실적 제한 조건과 이에 따른 고려 내용의 기술(필요시 자료 첨부)
(본 과제 수행함에 있어 고려할 현실적 제한조건을 기술)

현실적 제한조건		
제한요소		고려할 내용
1. 산업표준	설계 제작품의 산업 표준 규격 참조	해당 사항 없음
2. 경제성	가능한 한 저렴한 비용과 주어진 여건 아래에서 제작	외부 API 및 배포 인프라 선택 시 추가 비용을 고려하여 최소한의 비용으로 합리적인 선정
3. 윤리성	참고 문헌/제품 인용 표시	6.5 참고 문헌에 명시
4. 안전성	안전하게 구현	사이트 보안을 위한 도메인 SSL 인증서 발급과 서버와의 https 통신
5. 신뢰성	지속적으로 구동	백엔드 서버의 무결성, 신뢰성을 고려하여 오류없이 지속적으로 구동
6. 미학	가급적 공학적 실용성을 갖춘 외형 구비	UI/UX의 기본원칙 4요소(유연성, 유효성, 직관성, 학습성)를 준수하여 디자인 실시
7. 환경에 미치는 영향	환경 유해 물질의 사용과 설계 제작품의 폐기 시 절차 규정	해당 사항 없음
8. 사회에 미치는 영향	사회 전반에 유익한 영향을 미치는 설계 제작품 창작 및 적용 분야 명기	온라인 화상회의의 회의 문화 개선 방향 제시 연택트 협업도구로써의 편의성 제공
9. 기타	지역 특성화 산업과 연계성 고려	해당 사항 없음

제 3 장 설계 구성 요소에 따른 결과 기술

◎ 설계의 구성 요소 체크 항목

(본 과제 수행함에 있어 고려할 설계의 구성요소 기술, 필수 항목)

설계 구성요소		
구성요소		실시여부
1. 목표 설정	<ul style="list-style-type: none"> - 브레인스토밍 등의 아이디어 창출 도구를 이용하여 설계 목표를 설정 - 현실적인 제한 요소와 공학적인 제한 요소를 감안하여 설정 	실시 완료
2. 합성	<ul style="list-style-type: none"> - 설계목표에 달성에 필요한 관련 기술을 조사 분석하여 제작 가능한 설계안 제시 (작품의 개념을 1차 합성함) 	실시 완료
3. 분석	<ul style="list-style-type: none"> - 다양한 방법으로 자료를 수집하고, 포괄적인 문제에 대한 분석 또는 결과물에 대한 유용성 분석을 실시 - 다양한 도구를 이용하여 설계서 작성 및 주요 부분에 대한 해석 결과 제시 	실시 완료
4. 제작	<ul style="list-style-type: none"> - 공학실무에 필요한 기술 방법, 도구들을 사용하여 설계서에 따른 제작, 혹은 프로그램 작성 	실시 완료
5. 시험	<ul style="list-style-type: none"> - 최종 결과물에 대한 시험 - 안전하고 지속적으로 구동가능한가를 테스트 	실시 완료
6. 평가	<ul style="list-style-type: none"> - 최종 시작품이 설계 가이드라인을 만족하고 결론이 일치하는지 평가하고 일치하지 않을 경우 개선 방안 고찰 - 발표 능력 평가 	실시 완료

1. 목표 설정

1.1 문제 해결을 위한 아이디어 및 구체적인 방법

1) 프로젝트 단위 회의

FLOG를 사용하기 위해선 회원가입이 필수적입니다. ID와 Email이 중복되지 않는다면 쉽게 회원가입이 가능하도록 합니다.

어떠한 업무 상황에 회의를 진행한다고 했을 때, 많은 회의가 진행됩니다. 따라서 회의의 간편한 관리를 위해 프로젝트 단위로 회의를 관리하도록 하였습니다.(프로젝트는 폴더 단위와 유사합니다.) 사용자가 회의를 시작하고 싶다면 프로젝트를 생성하고, 회의에 참여할 사람의 이름을 검색하여 프로젝트에 초대합니다. 해당 프로젝트에 소속된 인원만 회의에 참가할 수 있으며, 회의록 정보 또한 프로젝트 폴더 안에 생성됩니다. 이는 관계형 데이터베이스 MySQL의 관계를 설정하여 구현합니다.

2) 음성 감정분석 모델 학습

저희 프로젝트 주제인 '감정 회의록'을 구현하는 데 있어 가장 중요한 파트입니다. 감정 분석을 하기 위해선 모델을 학습시켜야 하며, 학습시킬 데이터셋은 교수님께서 제공해주신 더미 데이터를 활용합니다. 제공받은 데이터셋의 개수가 1000개 미만으로, 일반적으로 모델을 학습시킬 때 사용되는 데이터셋의 개수보다 적었습니다. 보통 모델 학습은 최적화 튜닝에 따라 정확도가 결정되고, 데이터셋의 개수가 적을 땐 딥러닝보다 머신러닝의 튜닝이 더 적합하다고 판단하여 모델 학습 방법으로 머신러닝을 채택하였습니다.

제공된 데이터셋은 파일명에 감정이 라벨링되어 있어, 이를 Observation Value로 적용하여 Regression 모델이 아닌 Classification 모델을 채택하였습니다. 저희는 현재 분류 방법중 가장 진화된 형태인 SVM(Support Vector Machine)을 사용합니다. 또한 분류 모델을 학습시키기 전에 음성 데이터셋의 전처리를 진행합니다. 감정 분류는 '감성기반 서비스를 위한 통화 음성 감정인식 기법 (Call Speech Emotion Recognition for Emotion based Services)' 논문을 참고하여 기쁨, 슬픔, 화남, 긴장, 무감정 총 5가지의 감정을 분류합니다. 학습된 모델은 반드시 모듈화를 거쳐 백엔드 서버에서 즉시 사용이 가능하도록 만들어야 합니다.

3) 실시간 화상회의

감정 분석 기능과 더불어 프로젝트에서 가장 주요한 기능입니다. FLOG는 웹 어플리케이션이니만큼, 웹 브라우저와 호환되는 화상회의 API가 필요하였습니다. 조사

결과 Web RTC 라는 웹 화상회의 표준이 있다는 것을 알게되었습니다. Web RTC 는 P2P로 동작하기 때문에 서버 구축에 많은 시간을 투자할 필요가 없고, 튜닝을 잘 해준다면 영상 화질과 음질 또한 준수하였기 때문에 해당 API를 채택하였습니다.

또한 사용자의 회의 참가 접근성을 높이고 보안을 위해 회의 참여 코드를 사용합니다. 참여코드는 회의방 생성 시간을 기점으로 해시코드를 사용하여 암호화되어 생성됩니다. 이는 javascript의 모듈을 사용해 구현합니다. 회의방 참가 후 회의 시작 및 회의 종료는 호스트만 선택할 수 있습니다.

실시간 화상회의 진행중에는 각 참가자의 화면,음성 뿐만 아니라 실시간으로 실행되는 STT의 결과 텍스트와 감정 분석 결과, 회의 평균 감정 및 참여도 순위를 지속적으로 업데이트해주어야 합니다. 각 브라우저의 화면이 모두 동일하게 업데이트되어야 하므로 Socket 통신을 사용해야 했습니다. 따라서 Flask 프레임워크를 사용, 소켓 서버를 구축하여 실시간 통신 결과를 처리하도록 구현합니다. STT는 Chrome STT API를 사용하며, 감정분석은 이미 학습된 모델 모듈을 소켓 서버에 올려 실시합니다. 감정분석 전 음성 데이터 전처리과정에서 실제 발언 시간이 나오게 되는데, 이를 활용하여 참여도를 계산합니다.

4) 감정 회의록

프로젝트 주제인 ‘감정 회의록’에 집중하여 사용자에게 감정 요소를 활용한 회의록 분석 기능을 제공해야 합니다. 회의록에 포함된 감정 요소를 활용하여 해당 회의의 분위기를 생생하게 느낄 수 있게 하거나, 다음 회의 분위기를 개선하는 등의 효과를 기대합니다. 감정 회의록은 실시간 회의에서 DB에 저장되며, 사용자가 제공받는 감정 회의록 분석 기능은 모두 이 원본 회의록을 통해 구현합니다. 회의록은 발언, 감정, 발언시간 등을 저장하고 있으며, 이 정보를 통해 감정 필터링, 회의록 요약, 감정 그래프, 참여도 랭킹 등의 기능을 제공합니다.

1.2 수행목표

저희의 1차 목표는 음성 감정 분석 모델의 정확도 향상이었습니다. 사용자의 만족도를 높이기 위해선 우선 감정 분석 모델이 정확한 결과를 추출해야만 합니다. 저희는 모델의 최종 정확도를 90%로 목표하여 모델을 구현하였습니다.

2차 목표는 웹 화상회의의 품질입니다. Web RTC를 사용할 때 API의 튜닝 상태에 따라 영상의 화질 및 음질이 달라지기 때문에 가장 고품질의 화질, 음질을 제공하도록 목표하였습니다.

3차 목표는 실시간 화상회의의 소켓 통신 지연률 관리입니다. 사용자는 실시간으로 변하는 감정과 발언 텍스트가 업데이트되는 것을 오래 기다려선 안됩니다. 소켓 통신에 지연이 생긴다면 사용자 경험이 매우 나빠질 것이고, 사용자의 발언과 실시간 채팅창의 싱크가 맞지 않는 결과를 초래합니다. 따라서 평균감정 및 참여도 연산 알고리즘의 효율성을 극대화하도록 목표하였습니다.

마지막 목표는 서비스 배포입니다. 현재 많은 사람들은 Zoom이나 Webex로 소통하고, 이는 상업적 서비스이므로 과금이 필수적입니다. 오픈소스인 Web RTC를 사용해 구현한 FLOG는 이러한 과금 문제에서 자유롭기 때문에 다양한 사용자가 무료로 사용이 가능합니다. 따라서 발전된 화상회의를 체험해보고 싶은 사용자를 위해 서비스 배포를 목표로 하였습니다.

2. 합성

2.1 기초 조사

2.1.1 관련 분야의 이론 및 기술 현황 조사

1) SVM(Support Vector Machine)

감정분석 모델을 구축하기 위해서는 필요한 알고리즘은 대표적으로 SVM(Support Vector Machine) 머신러닝 알고리즘과 CNN(Convolutional Neural Network) 딥러닝 알고리즘이 존재합니다. 모델을 구축하기에 제공받은 데이터셋이 1000개 미만이라 CNN 딥러닝 알고리즘 보다 SVM 머신러닝 알고리즘이 적합하다고 판단하여 SVM 머신러닝 알고리즘을 사용하여 모델을 구축하기로 결정하였습니다.

SVM(Support Vector Machine)은 기계학습의 분야 중 하나로 패턴 인식, 자료 분석을 위한 지도 학습 모델이며, 주로 분류와 회귀 분석을 위해 사용됩니다. 1995년 Vapnik에 의해 발표된 개념이며, 서로 다른 두 그룹을 분류(classification)하기 위해 제안된 방법입니다.

두 카테고리 중 어느 하나에 속한 데이터의 집합이 주어졌을 때, SVM 알고리즘은 주어진 데이터 집합을 바탕으로 하여 새로운 데이터가 어느 카테고리에 속할지 판단하는 비확률적 이진 선형 분류 모델을 만듭니다. 만들어진 분류 모델은 데이터가 사상된 공간(서로 다른 특징의 데이터들을 한 공간에 할당하는 것)에서 경계로 표현되는데 SVM 알고리즘은 그 중 가장 큰 폭을 가진 경계를 찾는 알고리즘입니다. SVM은 선형 분류와 더불어 비선형 분류에도 사용될 수 있습니다. 비선형 분류를 하기 위해서는 주어진 데이터를 고차원 특징 공간으로 mapping하는 작업이 필요한데, 이를 효율적으로 하기 위해 커널 트릭을 사용하기도 합니다.

커널 트릭이 트릭인 이유는 모든 데이터를 mapping 하지 않으면서 똑같은 효과를 내게 만들었기 때문입니다.

$$L = -\frac{1}{2} \sum_j \sum_i \alpha_i \alpha_j y_i y_j x_i^T x_j + \sum \alpha_i$$

그림1) 커널 함수1

여기서 데이터는 $x_i^T x_j$ 인데, 우리는 데이터가 자체가 필요한 게 아니고 데이터의 내적이 필요합니다. 즉 feature space 에서 내적을 계산할 수 있다면 데이터의 mapping 은 구할 필요가 없다는 것입니다. $\phi(x)$ 를 구하지 않고 바로 $\phi(x_i)^T \phi(x_j)$ 를 구합니다. 고차원의 각 점들을 구하는 것보다 고차원에서 내적을 바로 구하는 trick 입니다. 따라서 커널 함수를 다음처럼 정의합니다.

$$L = -\frac{1}{2} \sum_j \sum_i \alpha_i \alpha_j y_i y_j K(x_i, x_j) + \sum \alpha_i$$

그림2) 커널 함수2

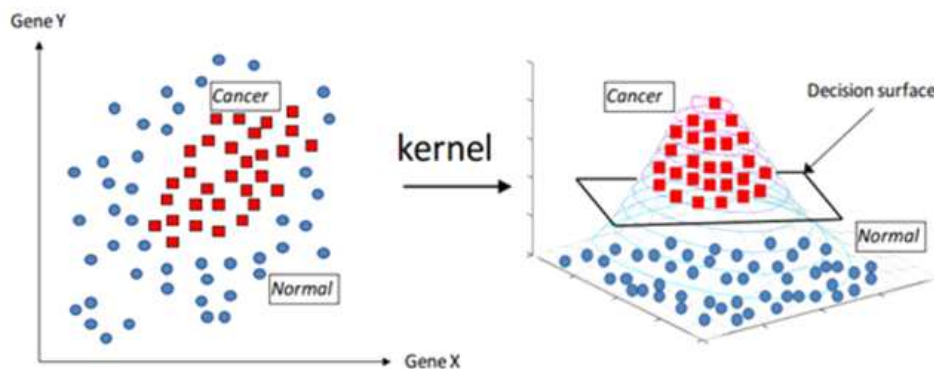


그림3) 커널에 대한 기하학적 설명

2) Web RTC(Web Real-Time Communications)

저희가 웹 화상회의를 구현하기 위해 꼭 필요했던 기술입니다. Web RTC란 웹 어플리케이션에서 별도의 소프트웨어가 필요 없이 음성, 영상 등의 미디어 데이터를 브라우저를 통해 주고받을 수 있게하는 통신 기술입니다. 현재 구글이 오픈소스화하였고, IETF의 프로토콜 표준화를 거쳐 W3C의 API 정의를 진행하여 누구나 사용 가능합니다. WebRTC로 연결된 프로그램은 p2p 방식으로 연결되며, 각 브라우저의 JavaScript API를 통해 실행됩니다.

브라우저의 p2p 통신을 위해선 각 peer의 방화벽 뒤 IP 주소 정보를 알아내야 합니다. 이 과정을 NAT traversal이라 칭하며, 이를 수행하는 서버를 STUN 서버라 부릅니다. STUN 서버는 서로의 IP를 식별할 수 있도록 중재하고, 각 peer를 연결시키는 역할을 합니다. 하지만 사용자마다 방화벽 설정이 다르기 때문에 STUN 서버가 모든 IP 주소를 알아낼 수 있는 것은 아닙니다. 따라서 이런 문제의 대안책으로 사용하는 것이 TURN 서버인데, 방화벽을 뚫지 않고 네트워크 미디어만을 중재하는 역할을 합니다. TURN 서버를 통한 연결은 대안책이기에 지연률이 증가하게 되지만, 혹시 모를 상황을 대비해 반드시 구현해야 합니다.

위 두 개의 서버를 통해 얻게 된 네트워크 주소의 조합은 Candidate라 불리고, ICE 프레임워크 위에서 Candidate를 사용해 실질적인 p2p 연결을 시켜줍니다. 연결에 성공했다면 미디어 해상도, 코덱 등이 SDP 프로토콜을 통해 설정되며 각 사용자의 화면을 공유하게 됩니다. 이 모든 과정을 포함하는 서버를 시그널링 서버라 부르고, Web RTC를 사용하려면 이 시그널링 서버의 구축이 필수적입니다. 시그널링 서버는 실시간 통신이 매우 중요하기 때문에 TCP가 아닌 UDP를 사용합니다.

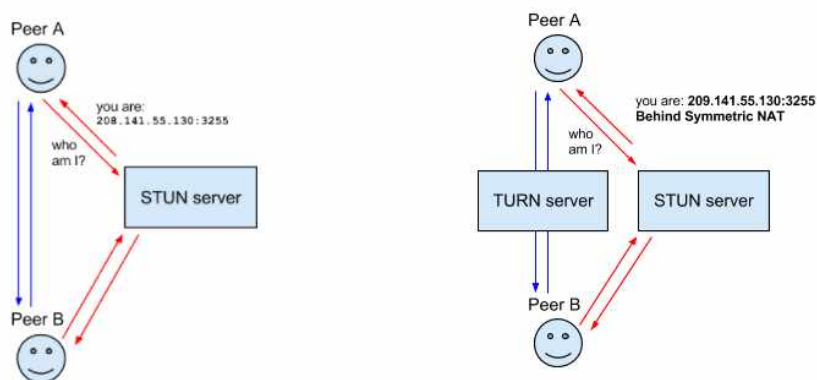


그림4) 시그널링 서버 개념도

3) Rest API / Socket 통신

Client와 Server 간의 통신의 종류는 다양합니다. FLOG는 Restful 통신의 한 종류인 Axios와 Socket 통신을 사용합니다. 그 중 Rest API 통신이란 http 통신의 한 형태이며, 자원을 이름으로 구분하여 해당 자원의 정보를 주고 받는 것을 의미합니다. Rest 통신은 HTTP Method를 통해 데이터에 대한 CRUD를 명시하여 사용합니다. Method 종류에는 POST, GET, PUT, DELETE, HEAD 가 존재합니다. Restful 통신은 http이므로 Client의 요청이 있을 경우에만 Server가 응답하는 단방향적 통신입니다.

그와 반대로 Socket 통신은 Client와 Server가 특정 포트를 통해 실시간으로 양방향 통신을 하는 방식입니다. 연결이 한번 성사되면 양측의 엔드포인트가 지속적인 연결이 됩니다. 소켓 통신은 TCP를 사용하여 통신하며, 웹 소켓이 전송계층에서 연결시켜줍니다. 보통 웹 소켓을 사용하기 위해선 외부 모듈이 필요한데, 보편적으로 사용되는 모듈은 Socket.io 모듈입니다. 소켓 통신은 실시간 통신에 특화되어 있으므로, 사용자의 입력없이 어떤 조건에서 데이터를 받아오는 경우에 적합합니다.

4) koNLpy 단어 토큰 추출

FLOG의 주요 기능인 참여도 산정과 워드클라우드에 사용되는 koNLpy 라이브러리는 NLP 전처리 및 Tokenizing 라이브러리입니다. 자연어처리 NLP에서 한국어의 형태소를 분리해주는 역할을 하며, 문장을 입력하면 형태소를 제거한 단어 토큰을 추출하여 줍니다.

koNLpy는 기본적으로 파이썬 라이브러리지만 자바를 기반으로 개발되었기에 JDK를 필수적으로 요구합니다. 파이썬 라이브러리인 Jpye가 Jvm과 연결시켜주며 이와 같은 환경설정이 끝난다면 5가지의 내부 클래스를 사용할 수 있습니다. 이 중 저희는 Okt 클래스를 사용합니다.

5) Docker

Docker는 컨테이너 기반의 오픈소스 가상화 플랫폼입니다. Docker는 배포환경에서 강력한 성능을 가지고 있습니다. 본 프로젝트의 목표 설정에서 배포를 목표로 하였기에 Docker를 사용합니다. 로컬 환경에서 개발하여 외부 클라우드 서버를 빌려 배포하는 방식은 로컬 환경과 대여 서버 환경을 동일하게 세팅해야하는데, 로컬 개발 단계에서부터 도커를 활용해 개발을 진행한다면, 배포시 대여 서버에서 세팅 없이 Docker Image의 공유만으로 배포가 가능합니다.

Docker는 기존 VM처럼 OS를 가상화하는 것이 아닌 컨테이너 방식으로 프로세스를 격리시켜 동작합니다. Docker엔 Image라는 개념이 있는데, 이미지란 컨테이너를 실행할 수 있는 실행파일과 설정 값을 가지고 있는 기본 베이스입니다. Dockerfile

을 코딩하여 Docker 이미지를 생성하고, 생성된 이미지를 여러 옵션을 넣어 컨테이너로 구동시킬 수 있습니다.

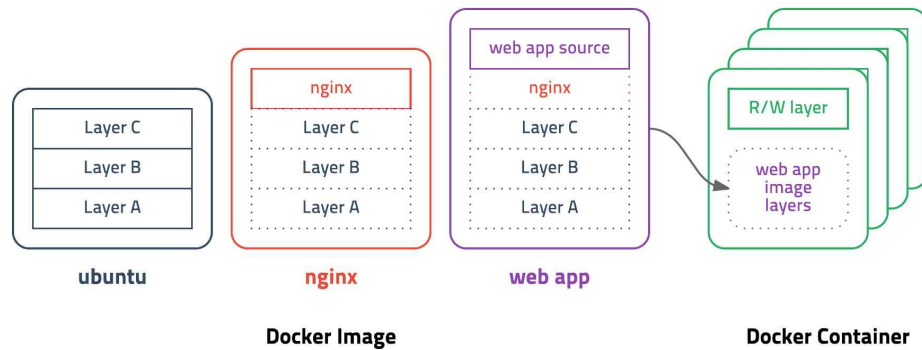


그림5) Docker 구조도

2.1.2 현 상황에서의 문제점 또는 해결이 필요한 사항

1) 감정 분석 모델 정확도 향상

학습 모델의 정확도를 높이려면 데이터셋의 정확한 데이터 전처리는 필수적입니다. 음성 데이터 전처리 과정에서 librosa 라이브러리를 사용해 wav 파일에서 시그널을 추출하는데, 이 시그널에서 묵음 및 잡음을 제거하여 정확한 특징만을 뽑아낼 수 있다고 생각하였습니다. 따라서 pydub 라이브러리를 사용해 묵음 및 잡음을 제거하였습니다. 추가적으로 Sequential Feature selection을 통해 유효한 특징만을 뽑아 학습 시키는 과정 또한 필요합니다.

또한 머신러닝 분류 모델의 정확도는 파라미터 튜닝이 매우 큰 영향을 미치기 때문에, 모델 학습 전 Hyperparameter Optimization을 진행하여 최적 파라미터 값을 찾아내었습니다.

2) Web RTC를 위한 HTTPS 적용

Web RTC는 사용자의 미디어 스트림에 접근하기 때문에 보안이 필수적입니다. 때문에 Web RTC API가 브라우저에서 실행되려면 해당 사이트의 주소는 반드시 SSL 인증서가 적용된 https여야 합니다.

실제로 Web RTC가 동작하는 프론트엔드(React)에만 fake SSL 인증서를 부여하여 실행하면 화상회의는 동작하지만, https와 http 프로토콜의 교차 통신은 허용되지 않기 때문에 프론트엔드와 통신하는 백엔드 Rest API 모두 https를 적용시켜줘야 정상적으로 동작합니다. 하지만 백엔드 API를 fake SSL 인증서를 부여하여 배포한다

면 브라우저가 SSL의 유효성 검사를 통해 통신을 차단하는 경우가 발생합니다.

따라서 유효한 SSL 인증서를 부여해야하고, 유효한 SSL 인증서를 발급받기 위해선 서버별로 도메인이 필요했습니다. 결국 서버별 도메인을 발급 후 각 서버에 https를 적용하였습니다.

3) Web RTC 시그널링 서버의 구축

앞서 설명했듯 Web RTC를 사용한 화상회의가 실행되려면 각 peer의 IP 주소를 매핑시키는 시그널링 서버가 필요합니다. 시그널링 서버의 구축 상태에 따라 지연률 및 화상회의 품질이 결정되므로, 시그널링 서버의 퀄리티가 매우 중요합니다.

저희가 직접 시그널링 서버를 구축해 보았으나 화질 이슈가 있어 이를 실제로 사용하지 않고 오픈소스로 공개된 외부 시그널링 소켓 서버(RTCMulticonnetion)를 사용하기로 결정하였습니다.

4) Chrome STT 브라우저 의존성

브라우저의 종류는 매우 다양합니다. 따라서 사용자들이 사용하는 브라우저의 종류가 각기 다를 수 있습니다. 저희가 사용한 STT API는 Chrome STT입니다. 이 API는 크롬 브라우저 내부에서 존재하므로 크롬 브라우저를 사용하지 않는 사용자는 STT가 작동하지 않습니다. FLOG는 데모 버전까지만 구현하므로 이 의존성을 해결하지 않습니다.

2.2 개념의 합성(개념 설계)

2.2.1 작동원리

1) 전처리 및 모델 학습 관련

저희는 감정분석 AI를 구현하기 위해 가지고 있는 데이터가 약 300개의 각각 형태와 크기가 다른 wav파일을 가지고 있기 때문에 통일을 시켜주기 위해 서로다른 음성의 길이를 4초로 자르도록 결정하였고, 4초로 자른 음성들을 MFCC를 적용하여 음성의 특징벡터를 추출하여 가지고 있는 데이터를 크기와 형태를 통일시켜 주었습니다.

이때 파이썬에서 음성파일을 처리할 때 주로 사용되는 librosa 라이브러리의 mfcc 함수를 이용하여 음성의 특징벡터를 추출하였습니다. 이후 특징벡터의 값을 일정 범위 내로 조정하기 위해 Scaler를 사용하여 특징벡터를 다시한번 전처리를 하여주었습니다. 이후 전처리를 한 데이터를 가지고 학습데이터와 검증 데이터로 나눈 뒤, 모델을 SVM머신러닝 알고리즘을 사용하여 모델을 학습시켰습니다.

학습 시키는 과정에서 가지고 있는 데이터들이 비선형이기 때문에 비선형 분류를 위해 커널은 방사 기저함수(RBF ; Radial Bias Function)을 선택하여 학습시키기로 결정하였고, 파라미터에 넣을 수 있는 값들을 순차적으로 입력한 뒤에 가장 높은 성능을 보이는 하이퍼 파라미터들을 찾는 탐색 방법인 GridSearch를 사용하여 최적의 파라미터를 찾은 후 모델을 구현하여 pkl파일로 저장하였습니다.

2) 서버 환경

FLOG는 프론트엔드(React) 서버와 백엔드 Rest API 서버(Express, Flask) 2개로 이루어져 있습니다. 또한 데이터베이스는 MySQL을 사용하며 AWS RDS 서비스를 사용하여 DB 서버를 구축하였습니다. Express 서버에서는 Sequelizer ORM을 사용하여 DB에 접근하고, Flask 서버에서는 SQLAlchemy ORM을 사용하여 DB에 접근합니다. 백엔드 서버를 2개를 사용하는 이유는 개발 도중 음성 데이터 전처리나 워크 클라우드 같은 기능을 수행할 python 서버가 필요해졌기 때문입니다. Express 서버에서는 인증 및 생성등의 일반적인 기능을 수행합니다.

프론트엔드와 백엔드의 Rest API 통신은 보통 Axios를 통해 이루어집니다.

3) 실시간 소켓 통신

실시간 화상회의에서의 모든 데이터 교환은 소켓 통신을 통해 이루어집니다. 호스트가 회의방을 개설하게 되면 참여코드가 랜덤 생성되고, 참가자는 참여코드를 통해 입장합니다. 입장하는 참가자 정보 리스트를 호스트가 관리하고, 참가자가 입장할 때 마다 해당 리스트를 다시 새로운 참가자에게 전달합니다. 이로써 새로운 참가자가 기존 참가자의 정보를 알 수 있게 됩니다.

회의 시작과 종료는 호스트만 실행할 수 있습니다. 호스트가 회의를 시작하게 되면, 회의시작 신호는 소켓 서버를 거쳐 모든 참가자 클라이언트에게 뿌려지게 됩니다. 회의 종료도 마찬가지입니다.

회의가 시작되면 모든 참가자의 브라우저에 내장되어 있는 Chrome STT가 실행됩니다. 참가자의 브라우저 STT가 각각 음성을 인식하기 때문에 자연스럽게 화자 구분이 이루어집니다. 음성이 인식되면 STT가 이를 감지하고, 음성이 끝나면 STT가 결과를 출력합니다. 음성 녹음 또한 STT의 시작과 끝의 시점이 동일합니다. 음성이 끝난 시점에서 STT 결과를 화면에 업데이트하는 것과 감정을 분석하는 것이 동시에 일어나야 하지만, 감정을 분석하는 것은 시간이 최대 3초 소요되므로, STT 결과를 한번만 소켓 전송하게되면 채팅창 업데이트에 딜레이가 생기게 됩니다. 따라서 화면 업데이트 소켓과, 감정분석 소켓을 분리하여 전송합니다. STT 결과는 사용자의 이름과 매핑되어 소켓 서버에 전송되고 이는 그대로 모든 클라이언트에게 뿌려져 모든 화면은 동시에 업데이트됩니다.

음성 녹음 또한 STT 결과와 함께 소켓 서버에 전송됩니다. 음성 녹음이 전송되기 전, 녹음 결과는 브라우저의 미디어 스트림으로 녹음되었기 때문에 파일 포맷이 webm입니다. webm은 웹 형식이므로 python에서는 이 형식을 읽을 수 없습니다. 따라서 클라이언트에서 포맷 변환 모듈을 사용하여 wav 포맷으로 변환합니다. 변환 후 FormData로 파일을 압축하여 소켓 서버에 전송합니다. 서버에서는 이 파일을 받고 librosa 라이브러리를 통해 음성 시그널로 변환합니다. 변환된 음성 시그널에서 pydub 라이브러리를 사용해 묵음 및 잡음을 제거하게 되고, 이 시점의 데이터 길이가 실제 발언 길이로 저장됩니다. 그 다음 mfcc를 통해 음성의 특징 벡터를 추출하고 학습된 모델에 넣기 전 scaler를 통해 차원을 변환합니다. 변환된 데이터를 4초 간격으로 분할하여 학습된 모델에 넣으면 해당 발언의 4초 단위 감정 리스트가 추출됩니다. 이 감정 리스트에서 가장 많이 등장한 단어가 감정 결과가 됩니다. 이렇게 얻게된 STT결과, 감정결과 및 발언 길이 등은 회의록 테이블에 저장되고, 클라이언트에게 전송해줍니다.

4) 워드 클라우드 및 회의록 요약

워드 클라우드 및 회의록 요약에는 텍스트 랭크 알고리즘이 필요합니다. koNLpy 라이브러리의 Okt 클래스에 회의의 모든 발언 리스트를 입력하면 해당 리스트에서 형태소를 제거한 단어 토큰들을 추출할 수 있습니다. 이 추출된 단어 토큰의 빈도수를 계산하여 텍스트 랭크를 생성합니다.

워드 클라우드는 이 텍스트 랭크를 프론트엔드에게 넘겨주면, 프론트엔드에서 React의 워드 클라우드 모듈을 통해 단어 크기에 순위 가중치를 주어 워드 클라우드를 생성합니다.

회의록 요약은 gensim 라이브러리를 사용해 구현합니다. gensim의 문장 요약 기능을 사용하여 파라미터 값을 조절해 회의록 요약이 가능합니다. gensim은 word2vec 알고리즘을 사용하여 유사한 문장과 중요한 문장을 추출하여 줍니다. 여기에 추가 파라미터로 텍스트 랭크의 반영하여 중요 단어에 가중치를 반영합니다.

5) 평균 감정 및 참여도 순위 산정

평균 감정 및 참여도 순위는 회의 시작 후 30초마다 산정됩니다. 회의 시작시 각 브라우저에서 타이머가 시작되며, 호스트의 타이머를 기준으로 산정 기준이 지정됩니다.

호스트의 브라우저에서 모든 참가자의 감정결과 리스트를 30초 단위로 누적하며, 이 리스트에서 가장 많이 등장한 감정이 해당 회의의 평균 감정이 됩니다.

참여도 순위는 STT 결과의 문장 길이, 실제 발언 시간, 텍스트 랭크를 기준으로 산

정됩니다. 이 누적 값 또한 호스트의 브라우저에서 참가자별 점수 리스트로 관리합니다. 참가자별 문장 길이와 실제 발언 시간의 합계에 텍스트 랭크를 참조하여 참가자 발언별로 텍스트 랭크의 단어가 포함된다면 이에 가중치를 주어 점수를 계산합니다.

6) 배포 환경

프론트엔드(React)는 Netlify와 Git Hub를 연동하여 자동 빌드 배포되도록 세팅하였고, 이에 도메인 flog.tk를 부여하였습니다.

2개의 백엔드 서버는 AWS EC2에 Docker를 활용해 배포하였고, Nginx를 사용해 Reverse Proxy를 적용하여 각 포트별 도메인을 매핑하였습니다. OpenSSL로 발급받은 SSL 인증서 또한 Nginx를 통해 적용합니다.

2.2.2 논리적인 구조도

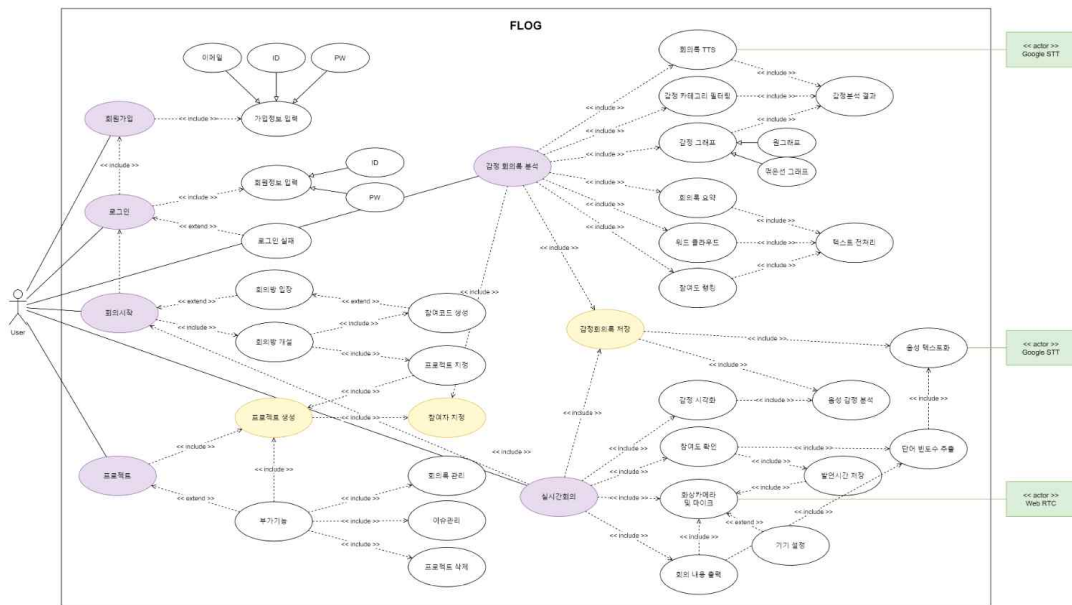


그림1) Usecase Diagram

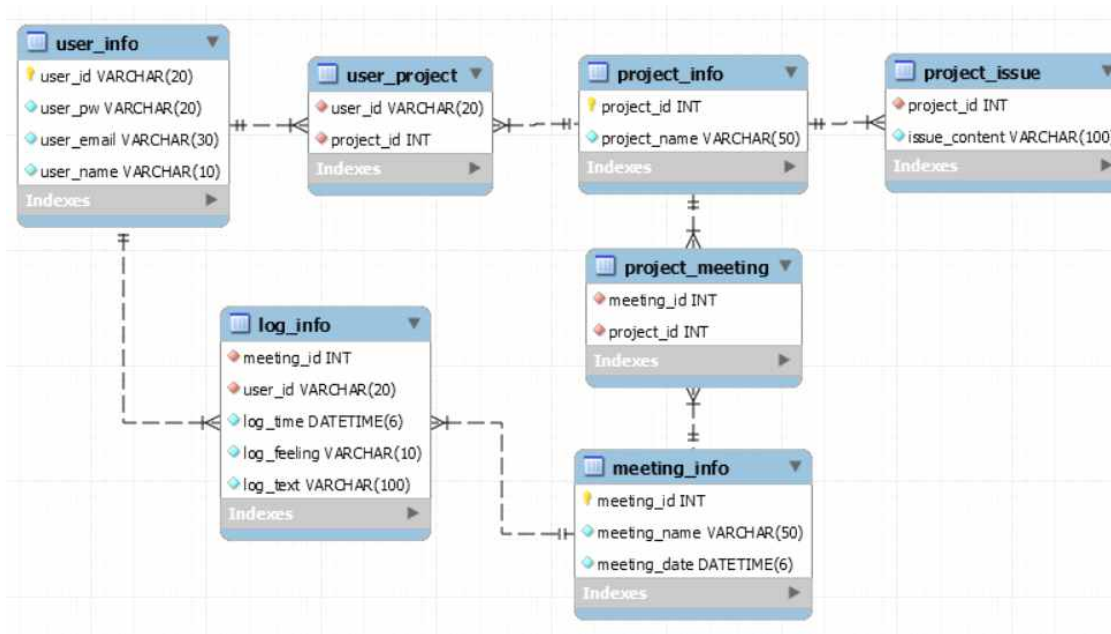


그림2) Schema Diagram

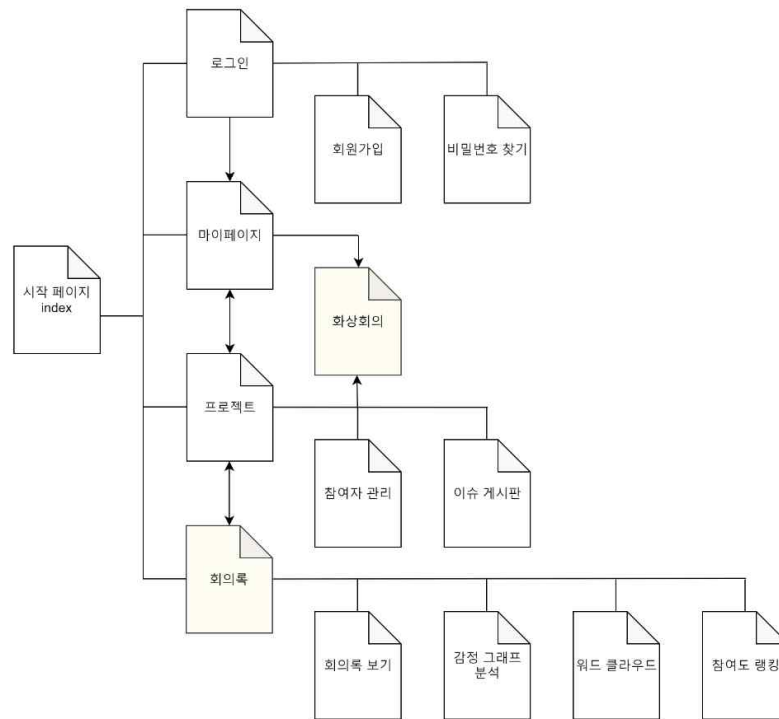


그림3) Site Map

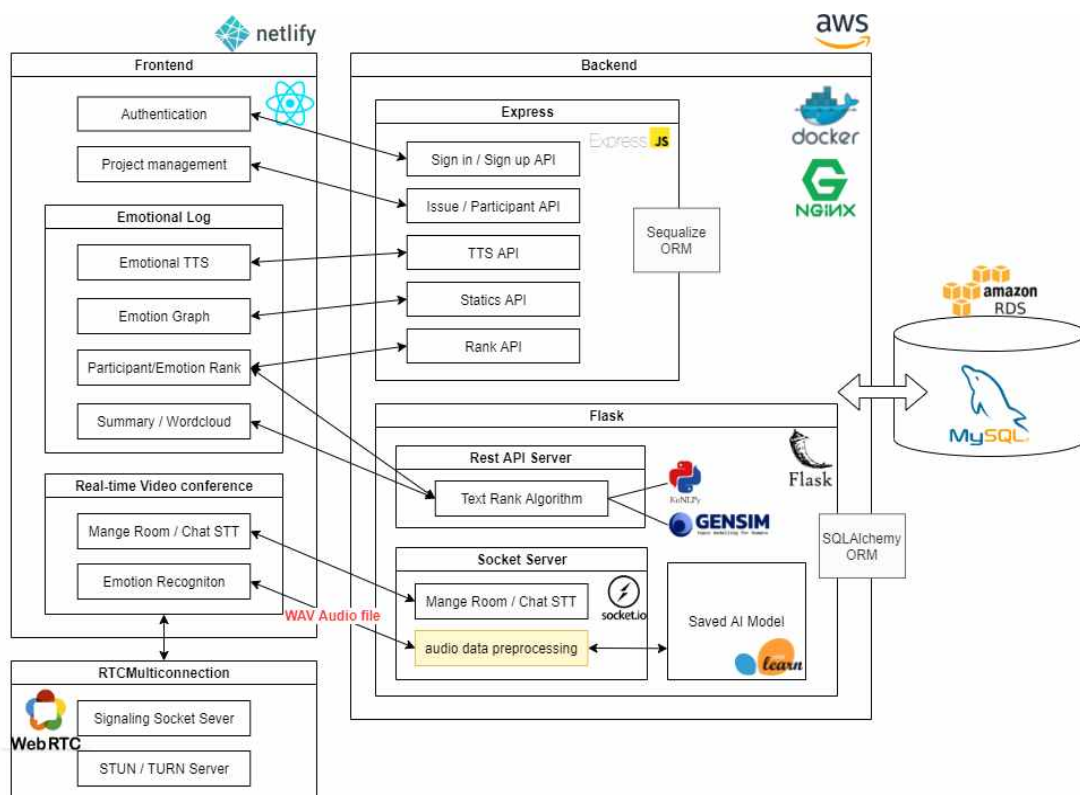


그림4) System Architecture

2.2.3 주요 기능

위에서 설명하지 않은 주요 기능으로 TTS 기능과 감정 시각화 차트 기능이 있습니다. TTS 기능은 외부 API인 프로소디를 통해 구현합니다. 감정 회의록 테이블은 하나의 레코드에 STT 결과와 감정 결과가 함께 저장되어 있습니다. 이를 활용하여 텍스트에 감정을 반영하여 TTS로 읽어줄 수 있습니다. 회의 종료시 해당 회의록은 곧바로 텍스트와 감정 파라미터를 API에 전달하여 TTS 인코딩이 시작됩니다. 인코딩된 결과는 mp3 포맷으로 저장되며, 프론트엔드의 Audio 객체를 활용하여 들을 수 있습니다.

감정 시각화 차트는 React 모듈인 chart.js를 사용합니다. 회의 평균 감정 그래프는 실시간 회의에서 30초마다 산정되어 저장되었으므로 테이블에서 꺼내어 시각화만 진행하면 됩니다. 감정 비율 그래프에서 프론트엔드는 백엔드에 해당 회의록 등장 감정을 백분율로 환산하여 요청합니다. 이는 chart.js의 원 그래프로 시각화됩니다.

3. 분석(작품 구현 과정 중의 문제점 분석 및 해결 방법)

3.1 과제수행에 사용된 이론 및 기술의 조사 및 분석 결과

3.1.1 본 과제를 수행함에 있어 활용된 수학, 기초과학

1) 선형대수 및 확률과 통계

머신러닝은 애초에 선형대수 및 확률과 통계 이론에 기초합니다. 저희가 사용한 SVM 모델과 이에 추가되는 전처리 및 튜닝 과정 모두 동일합니다. SVM은 선형대수의 평면 방정식을 사용하여 hyperplane을 표현합니다. 또한 Sequential Feature selection 등의 최적화 과정에서는 통계 이론이 사용됩니다.

2) 기계학습 - 라그랑주 승수법

SVM은 오류를 최소화 하여 분류하는 것으로부터 한발짝 더 나아가 분류된 데이터셋을 고려해 margin을 최대화 하여 일반화 능력을 극대화 시키는 것입니다. 모든 샘플들의 집합을 올바르게 분류한다는 가정하에 최대 여백을 갖는 초평면을 찾으려는, 즉 조건부 최적화 문제입니다. 이때 조건부 최적화 문제를 해결하기 위해 라그랑주 승수법을 사용합니다.

3) 기계학습 - 커널 트릭

매핑함수를 직접 사용하는 대신 SVM 초평면 계산에서 사용되는 벡터 내적 연산을 대체하는 비선형 커널함수를 정의하여 사용합니다.

- Polynomial kernel $K(x, y) = (x^T y + 1)^d$
- Radial kernel $K(x, y) = \exp(-|x - y|^2 / (2\sigma^2))$
- Sigmoid kernel $K(x, y) = \tanh(kx^T y + \theta)$

그림1) 커널 함수

3.1.2 본 과제를 수행함에 있어 활용된 전공 이론과 정보기술

1) 기계학습

SVM 분류 모델은 머신러닝 알고리즘 중 지도 학습 알고리즘의 한 종류로 결정 경계를 사용하여 모델을 학습하는 알고리즘입니다. SVM 알고리즘은 감정 분석 모델을 구축하기 위해서 사용하였으며 학습을 통해 행복, 무감정, 슬픔, 화남 및 긴장 감정을 최선으로 분류해주는 벡터를 결정 경계로 결정합니다.

2) 컴퓨터 네트워크

프로젝트에서 사용된 Web RTC, Axios, Socket.io 등의 통신 기술은 모두 컴퓨터 네트워크에 기초합니다. Web RTC는 실시간 미디어를 주고받아야 하므로 UDP 프로토콜을 사용합니다. Restful 통신인 Axios는 http 프로토콜을 사용하며, Socket.io 모듈은 Application 계층과 Transport 계층 사이의 웹 소켓을 사용하고 TCP 통신을 사용합니다. 따라서 프로토콜의 특성과 OSI 7 계층의 통신 이론을 이해하고 있어야 합니다.

3.1.3 본 과제를 수행함에 있어 활용된 공학도구, 기술 및 장비

저희가 기획한 프로젝트의 결과물은 웹 어플리케이션, 즉 소프트웨어 서비스이므로 공학도구나 기술, 장비는 활용하지 않았습니다.

3.2 설계물에 대한 분석 및 보완

3.2.1 작동원리상에서 나타난 문제점 분석 및 보완

1) Flask 서버 배포시 koNLpy 이슈

koNLpy는 python 라이브러리이지만 java를 기반으로 개발되었기에 jdk를 필요로 합니다. 또한 Jpype 라이브러리를 통해 python과 java를 연결합니다.

로컬 개발서버(windows)에서 koNLpy가 문제없이 잘 작동되지만, 이를 배포서버(Ubuntu)에서 구동하면 다음과 같은 에러가 발생합니다.

```
A fatal error has been detected by the Java Runtime Environment:

SIGSEGV (0xb) at pc=0x00007fb5dbf3ad58, pid=18444, tid=0x00007fb5aa3ec700

JRE version: OpenJDK Runtime Environment (8.0_292-b10) (build 1.8.0_292-8u292-b10-0ubuntu1~18.04-b10)
Java VM: OpenJDK 64-Bit Server VM (25.292-b10 mixed mode linux-amd64 compressed oops)
Problematic frame:
C [_jpype.cpython-36m-x86_64-linux-gnu.so+0x3cd58] JPJavaEnv::FindClass(char const*)+0x28

Failed to write core dump. Core dumps have been disabled. To enable core dumping, try "ulimit -c unlimited" before starting Java ag

If you would like to submit a bug report, please visit:
http://bugreport.java.com/bugreport/crash.jsp
```

그림2) java 오류

이는 koNLpy의 Jpype가 Flask 서버의 스레드 환경에서 구동되었을 때 발생하는 오류로, 로컬서버의 CPU보다 Ubuntu 서버의 CPU 성능이 떨어지기 때문이라고 판단하였습니다. 따라서 Jpype가 이미 실행되었다면 바로 JVM의 스레드를 연결시켜 오버헤드가 발생하지 않도록하여 해결하였습니다.

2) SQLAlchemy ORM 이슈

Flask 서버에서 ORM 사용시 발생하는 오류입니다. 프론트엔드에서 백엔드로 단기간에 많은 요청을 보내게 되면 ORM이 작동이 중단됩니다. 요청이 발생할때마다 ORM은 DB에 접근해야 하는데, SQLAlchemy DB 접근 방식에서 Queue pool이라는 개념을 사용하기 때문에 오류가 발생하였습니다.

```
sqlalchemy.exc.TimeoutError: QueuePool limit of size 10 overflow 10 reached, connection timed out, timeout 30.00 (Background on this error
e/14/3o7r)
58.122.52.25 - - [29/May/2021 09:02:53] "GET /api/log/summary/1 HTTP/1.1" 500 -
ubuntu@ip-172-31-36-54:~$ ^C
ubuntu@ip-172-31-36-54:~$ █
```

그림3) Queue pool Overflow

따라서 요청을 받아 DB에 접근한 후 응답까지 완료되면 연결된 세션을 닫아 Queue pool을 비워주는 방식으로 해결하였습니다.

3) 학습된 모델 사용

저희는 모델 학습이 완료되면 학습된 모델을 pkl 확장자로 저장하였습니다. 그리고 이 pkl 파일을 백엔드 서버에 올려 사용하였는데, 학습시 데이터 전처리와 동일한 과정을 거친 테스트 음성 데이터의 감정을 분석하려 하니 오류가 발생하였습니다. 학습 과정에서 분류 모델 뿐 아니라 Scaler 또한 학습하였기 때문에 모델 사용시에 데이터 전처리 과정에서 동일한 과정을 거쳐줘야 했기 때문에 Scaler 또한 모듈화시켜 사용하였습니다.

3.2.2 구조도상에서 나타난 문제점 분석 및 보완

1) Database 관계 설정

초기 데이터베이스를 설계할 때, 불필요한 관계가 많이 설정되어 있었습니다. 이는 백엔드에서 ORM을 사용함에 있어 객체 모델을 세팅하는데 많은 시간을 소요하게 만들었습니다. 따라서 불필요한 관계를 제거하고 직관적인 데이터베이스를 재설계 하였습니다.

3.2.3 주요 기능상에서 나타난 문제점 분석 및 보완

1) STT

앞서 설명했듯 저희는 Chrome STT를 사용합니다. 이는 브라우저에서 지원하는 API이므로 비용이 들지 않습니다. 하지만 서비스를 사용하려면 사용자가 반드시 크롬 브라우저를 사용해야하는 의존성 문제가 있으므로, 다른 STT API를 프론트엔드에 올려 사용한다면 이와 같은 의존성 문제를 해결할 수 있을 것으로 보입니다. 하지만 일반적인 STT API는 비용을 지불해야 하기에 해결하기 쉽지 않은 문제입니다.

2) TTS

저희는 프로소디 API를 사용하여 TTS를 구현하였습니다. 하지만 프로소디 또한 상업적 서비스이므로 무료로 TTS 변환은 6천개의 단어까지 가능하였습니다. 이후 생성되는 TTS는 과금이 발생합니다. 여러 TTS API는 존재하지만 감정요소를 부여한 TTS 서비스는 많지 않기 때문에 이와 같은 과금 문제는 아직 해결할 수 없었습니다.

4. 제작

4.1 완성품 제작 결과 (사진)



그림1) 메인 화면

The image displays two side-by-side screens from the FLOG application. The left screen is titled '로그인' (Login) and features two input fields labeled 'ID' and 'Password', followed by a '로그인' (Login) button and a link '회원가입 | 비밀번호 찾기' (Sign up | Find password). The right screen is titled '회원가입' (Sign up) and includes five input fields: '이름' (Name), '아이디' (ID), '이메일' (Email), '비밀번호' (Password), and '비밀번호확인' (Confirm password), followed by a '회원가입' (Sign up) button.

그림2) 로그인 및 회원가입 화면

'김수지'님 환영합니다 :)

프로젝트 생성



프로젝트 폴더를 만들어 회의를 관리해보세요.

회의 참여하기



코드를 입력하면 회의에 참여할 수 있어요.

<

>

나의 프로젝트

- 캡스톤프로젝트** | 참여자 수: [3 명] | 최근 회의 날짜: [2021-06-03]

그림3) 마이페이지 화면


FLOG
감정을 기록하다. Feeling LOG

[MYPAGE](#)
[LOGOUT](#)

캡스톤프로젝트

- [회의LOG 관리](#)
- [ISSUE 관리](#)
- [참여자 관리](#)

방 만들기



방장이 되어 회의를 시작해보세요.

회의 LOG

- 2021-06-02 | 제목: [캡스톤회의] 참여자: [한재원 김수지 강전호]
- 2021-06-03 | 제목: [캡스톤회의] 참여자: [한재원 김수지 강전호]
- 2021-06-03 | 제목: [캡스톤회의] 참여자: [한재원 김수지 강전호]
- 2021-06-03 | 제목: [캡스톤회의] 참여자: [한재원 김수지 강전호]

[감정 회의록](#)

[감정 회의록](#)

[감정 회의록](#)

[감정 회의록](#)

Issue

원한 모두가 공유해야 하는 간단한 공지사항이나 링크 등을 이슈에 등록해보세요.

[+ 이슈 등록](#)

- 0월 4일 금요일 제출
- 이슈를 등록합니다

참여자 관리

현재 참여자 목록

- 강전호
- 한재원
- 김수지

그림4) 프로젝트 하위 회의록 리스트 및 이슈, 참가자 등록 화면



그림5) 실시간 회의 화면

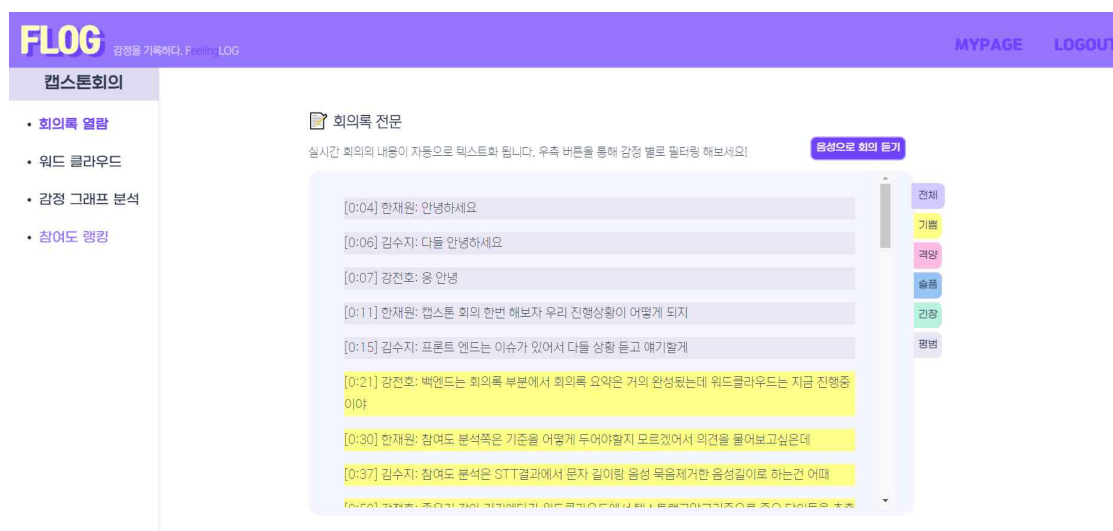


그림6) 회의록 필터링 및 TTS 화면



그림7) 워드 클라우드 및 회의 요약문 화면

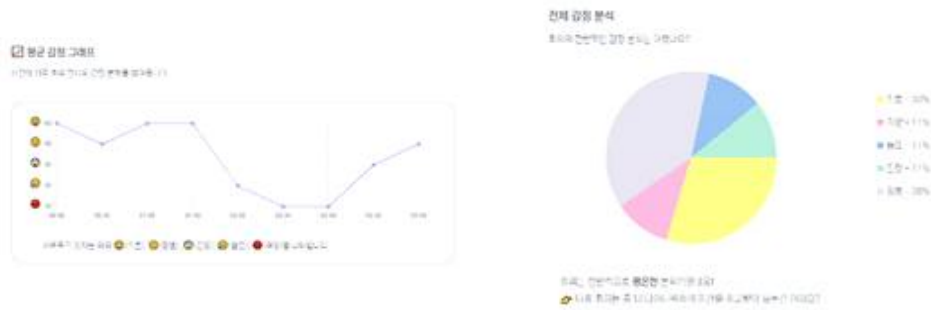


그림8) 평균감정 그래프 및 전체 감정 분포도 화면



그림9) 참여도 랭킹 및 감정 대표자 화면

4.2 완성품 설명

음성 감정 인식 기반 실시간 화상 회의 및 감정 회의록 서비스인 FLOG는 로그인 후 이용할 수 있습니다. 로그인을 마치면 가장 먼저 마이페이지가 활성화됩니다.

마이페이지에서는 프로젝트 폴더를 생성하거나, 회의에 참여할 수 있으며, 하단에는 사용자가 속해 있는 프로젝트 목록을 볼 수 있습니다. 목록에서 원하는 프로젝트 이름을 클릭하면 프로젝트 폴더로 이동하게 됩니다. 프로젝트 폴더에서는 회의 방을 개설하거나, 작성된 감정 회의록을 확인할 수 있으며, 이외에도 좌측 메뉴바를 통해 이슈를 등록하거나 참여자를 관리할 수 있습니다. FLOG에서는 프로젝트 별로 회의록과 참가자 등을 관리할 수 있도록 했습니다. 이를 통해 체계적인 관리가 가능합니다.

실시간 회의 방이 개설되면, 방을 개설한 사용자가 방장이 되고, 회의의 시작과 종료는 방장의 권한이 있어야 가능합니다. 코드 공유는 모든 사용자가 접근할 수 있으며, 버튼 클릭 시 클립보드에 회의 코드가 복사됩니다. 이를 통해 참가자를 초대할 수 있습니다. 방장이 회의 시작 버튼을 클릭하게 되면 회의가 시작되었음을 알리는 알림창이 모든 사용자의 화면에 나타나고, 그때부터 STT가 활성화되고, 각 발언에 따른 감정 분석을 진행하여 그 결과를 사용자의 카메라 화면 상단에 이모티콘으로 시각화하여 표시합니다. 각 이모티콘에 대한 감정은 우측 하단에 위치한 감정 가이드를 통해 쉽게 확인할 수 있습니다. 회의 전체 분위기와 참여도 순위는 회의가 시작된 지 30초 후부터 산정되며, 이후 30초마다 새롭게 갱신합니다. 회의 종료하기를 클릭하면 회의가 종료되었음을 알리는 알림창이 모든 사용자의 화면에 나타난 뒤, 해당 회의를 바탕으로 작성된 회의록 페이지로 이동합니다.

감정 회의록은 실시간 회의를 마친 후 바로 이동하게 되는 화면이며, 이외에도 프로젝트의 회의록 관리 메뉴의 목록을 통해 접근할 수 있습니다. 회의록은 회의가 진행된 날짜와 사용자가 설정한 이름, 참여한 사용자의 이름이 함께 목록으로 나타납니다.

회의록 열람 메뉴에서는 STT를 바탕으로 한 회의록 전문을 확인할 수 있으며, 우측 감정 필터 버튼을 통해 해당하는 감정의 발언들만 필터링 할 수 있습니다. 또한 음성으로 회의 듣기 버튼을 클릭 시, 감정 요소가 포함된 음성으로 회의 내용을 들을 수 있습니다. 또한 좌측 메뉴바를 통해 워드 클라우드, 감정 그래프, 참여도 랭킹을 확인할 수 있으며, 워드 클라우드의 경우 실시간 회의에서 많이 등장한 단어 상위 15개를 추출하고 많이 등장할수록 가중치를 두어 단어가 더 크게 나타납니다. 또한 회의 요약문도 함께 제공합니다. 이를 통해 회의의 주제와 흐름을 빠르게 확인할

수 있습니다. 감정 그래프 분석 화면에서는 실시간 회의에서 30초 마다 산정한 회의 전체 분위기를 모두 취합하여 시간 추이와 함께 보여줍니다. 또한 하단의 전체 감정 분석은, 모든 발언에 대해 분석한 감정이 차지하는 비율을 원그래프로 나타내어 보여줍니다. 이를 통해 회의가 전반적으로 어떤 분위기에서 진행되었는지 확인할 수 있으며, 가장 많이 등장한 감정이 무엇인지와 함께 그에 따른 조언을 통해 다음 회의에서 분위기를 개선하는 데에 도움이 되도록 했습니다. 마지막으로 참여도 랭킹에서는, 실시간 회의를 바탕으로 가장 적극적으로 참여한 사용자를 3위까지 선정하고, 각 감정별로 해당 감정으로 가장 많이 발언한 사용자들을 산출하여 나타냅니다. 이를 통해 가장 적극적으로 참여한 사용자를 판별할 수 있습니다.

4.3 작품 제작 과정 정리

내용	시작 날짜	완료 날짜
데이터베이스 및 백엔드 서버 구축		
MySQL workbench를 통한 테이블 설계 및 생성	21.04.02	21.04.04
Express로 개발 환경 구축	21.04.02	21.04.04
감정 분석 모델 개발		
음성 데이터 가공 및 특징 벡터 추출	21.04.02	21.04.04
특징 벡터 추출 알고리즘 최적화(파라미터 조정)	21.04.04	21.04.16
분류 모델 선정 및 이론 학습	21.04.06	21.04.09
분류 모델 설계 및 학습과 검증 데이터 분류	21.04.12	21.04.14
학습 및 검증 데이터 전처리	21.04.13	21.04.14
모델 학습 및 학습된 모델과 전처리된 저장	21.04.21	21.04.22
웹 UI 및 세부 기능 구현		
전체 화면 설계 및 기획	21.03.29	21.04.01
홈 화면 페이지 구현	21.05.05	21.05.06
로그인 및 회원 가입 화면 구현	21.05.07	21.05.13
마이 페이지 화면 구현		
-프로젝트 생성 기능 구현	21.05.15	21.05.17
-회의 참여 기능 구현	21.05.15	21.05.16
프로젝트 폴더 화면 구현		
-사이드바 생성 및 구현	21.05.17	21.05.20
-이슈 등록 화면 구현 및 DB에 정보 저장	21.05.17	21.05.20

-참여자 관리 화면 구현 및 DB에 정보 저장	21.05.21	21.05.23
감정 회의록 화면 구현		
-회의록 관리 화면 및 감정 필터링 구현	21.05.22	21.05.24
-워드 클라우드 화면 구현 및 Flask서버와 통신	21.05.24	21.05.27
-감정 그래프 화면 구현 (Line-Chart와 Pie-Chart 이용)	21.05.25	21.05.27
-참여도 랭킹 화면 구현	21.05.27	21.05.28
화상 회의 화면 구현		
-Web RTC로 구현한 화상 회의 연동	21.05.29	21.06.02
-회의 메뉴 생성	21.05.25	21.06.01
-STT및 회의 분위기, 참여도 순위 시각화	21.06.01	21.06.02

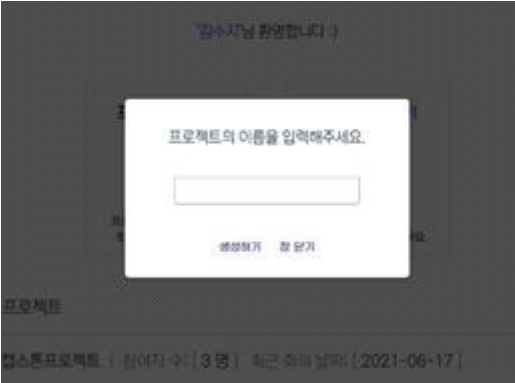
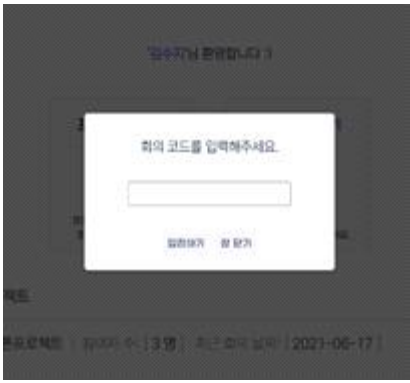
4.4 작품의 특징 및 종합설계 수행 결론

FLOG는 실시간 화상회의에서 음성을 통한 감정 분석을 진행하고 이를 시각화 하여 나타내므로 회의에 참가한 사용자가 보다 쉽게 상대방의 감정을 파악할 수 있기에 회의에 보다 집중할 수 있으며, 참여도에 대해서 정량적으로 평가하여 순위로 나타냈습니다. 또한 비대면 업무에 최적화하여 여러 프로젝트를 동시에 진행하더라도 각 프로젝트 별로 회의록을 체계적으로 관리할 수 있도록 했으며, 프로젝트 별로 참가자를 다르게 설정할 수 있고, 이슈 등록이 가능합니다.



또한, 실시간 회의에서 분석한 감정 결과를 회의록에도 사용하여 텍스트로만 이루어진 일반적인 회의록에 비해 보다 생생하게 회의 분위기를 읽을 수 있습니다. 이외에도 워드 클라우드와 회의 요약문, 감정 그래프와 참여도 랭킹에 대해 제공함으로써 단순히 음성을 텍스트화 한 회의록에서 그치지 않고, 사용자에게 도움이 되는 회의록을 제공합니다.

4.5 완성품의 사용 매뉴얼



기본적으로 FLOG의 모든 기능은 회원가입 후 사용할 수 있습니다. 로그인을 마치면 마이페이지로 이동하게 되는데, 이곳에서 프로젝트를 생성하거나 회의 방에 참가할 수 있습니다.

	
<p>프로젝트 생성 버튼 클릭 시 나타나는 모달창에 프로젝트의 이름을 입력하면 프로젝트가 생성됩니다.</p>	<p>회의 참여하기 버튼 클릭 시 나타나는 모달창에 회의 코드를 입력하면 회의 방에 입장할 수 있습니다.</p>
<p>그림1) 마이페이지 모달</p>	

프로젝트의 참여자를 관리하기 위해서는 해당 프로젝트 폴더로 이동한 후 참여자 관리 메뉴로 이동합니다. 좌측 검색창에 사용자의 이름을 입력하면 해당하는 이름을 가진 사용자의 아이디가 출력됩니다. 따라서 동명이인이 존재할 경우 아이디를 통해 구별할 수 있도록 했습니다. 좌측에서 현재 프로젝트의 참여자 목록을 확인할 수 있습니다. 또한 이슈 관리 메뉴에서 이슈를 등록할 수 있습니다.

	
<p>추가 버튼을 누르면 해당 사용자가 참여자에 추가됩니다.</p>	<p>이슈 등록 버튼을 클릭하면 나타나는 모달창입니다. 이슈 내용을 입력합니다.</p>
<p>그림2) 참가자 등록 및 이슈 등록</p>	

새로운 회의를 개설하기 위해서는 생성하고자 하는 회의에 관련된 프로젝트 폴더로 이동합니다. 방 만들기 버튼을 클릭하고, 나타나는 모달창에 회의의 이름을 입력하면 새로운 회의가 개설됩니다. 이때, 방을 개설한 사용자가 방장의 권한을 가지게 되며, 회의의 시작과 종료를 관리할 수 있습니다.

	
<p>회의 시작 버튼을 클릭하면 그 때부터 STT 및 감정 분석, 참여도 랭킹 산정이 시작됩니다.</p>	<p>좌측 하단에는 30초 마다 갱신되는 회의 분위기와 사용자의 참여도 순위를 아이콘으로 나타냅니다.</p>
<p>그림3) 회의 기능 설명</p>	

회의가 종료되면 해당 실시간 회의를 바탕으로 생성된 회의록 화면으로 이동합니다. 이곳에서 회의록 전문을 확인하고, 음성 TTS를 들을 수 있습니다. 회의록의 경우 우측의 감정 필터 버튼을 통해 해당하는 감정으로 필터링 할 수 있습니다.

	
	
<p>그림4) 회의록 감정 필터링</p>	

워드 클라우드와 회의 요약문은 회의록 좌측의 워드 클라우드 메뉴를 통해 조회 가능하며, 평균 감정 그래프와 전체 감정 분석의 경우는 감정 그래프 분석 메뉴에서 조회 가능합니다. 또한, 참여도 랭킹 메뉴에서 참여도 1, 2, 3순위인 참여자를 확인할 수 있으며, 각 감정 별 대표자를 확인할 수 있습니다.

5. 시험 (시험 결과 기술)

5.1 최종 결과물에 대한 시험 결과

테스트	테스트 방법/절차	테스트 결과
회원가입 테스트	회원가입후 데이터 베이스에 정보가 저장되는지 확인	데이터베이스에 정확하게 회원정보가 저장됨
로그인 테스트	가입된 경우, 가입이 안되어 있는 경우, 비밀번호가 틀린 경우, 비밀번호가 맞는 경우로 나누어 테스트	가입되어 있는 상태의 정확한 아이디와 패스워드를 입력하여야 로그인이 되는 결과 확인
jwt토큰 테스트	로그인이 성공한 경우 jwt토큰을 통하여 안전하게 회원정보를 관리 하는지 테스트	로컬 스토리지에 저장하고 삭제하는 과정을 통해 로그인 로그아웃 시 토큰이 잘 관리되는지 확인했으며 암호화 된 토큰을 해독하여 아이디와 사용자의 이름을 추출했습니다
프로젝트 생성 테스트	프로젝트 생성 확인	사용자가 프로젝트를 생성한 경우 데이터베이스에 저장되며, 웹페이지에서 바로 생성 되는 것이 확인됨
프로젝트 참여인원 테스트	프로젝트 참여자 확인	생성되어 있는 프로젝트에서 사용자가 프로젝트에 참가 시킬 인원을 검색 및 추가 하는 것이 데이터베이스에 반영되는지 확인 및 웹페이지 상에서도 반영되는지 확인됨
회의생성 테스트	회의 생성 확인	회의를 생성하면 화상회의 창으로 넘어가는 것이 확인됨
회의 참여코드 테스트	회의 참여코드 확인	회의창에서 다른 사용자에게 참여코드를 복사해서 알려줄 수 있는 것이 확인됨.
회의록 저장 테스트	화상회의를 진행하는동안 자동으로 회의록이 데이터베이스에 저장되는지 확인	화상회의 동안 STT기능이 구현되어 자동으로 회의록을 생성하여 데이터베이스에 저장 되는 것을 확인.
테스트	테스트 방법/절차	테스트 결과
워드클라우드 생성	명사의 빈도수를 기반으로 워드클라우드 생성 여부 확인	웹페이지에서 워드 클라우드가 생성 되는 것을 확인함.
회의록 자동요약	회의가 끝난후 회의가 요약되는지 확인	웹페이지에서 회의가 요약 되는 것을 확인함.
데이터셋 전처리 테스트	학습시킴 데이터셋이 같은 형식으로 되어있는지 확인	각기 다른 데이터셋을 4초 짜리 파일로 통일화 시킨것을 확인.
데이터셋 특징벡터 추출 테스트	같은 형식의 데이터셋의 음성을 몇 개의 특징벡터로 추출할지 직접 시행	특징벡터를 추출 하는 과정에서 n_mfcc인자를 36으로 하였을 경우 정확도가 가장 높은 것으로 확인함.
모델 학습 테스트	파라미터를 조정하여 정확도를 높임.	일일이 파라미터값과 커널을 조정하던중 girdsearch함수를 사용하여 rbf커널 의 파라미터 최적화를 하는 값을 찾아냄
감정분석 머신러닝 설계	설계된 모델의 test데이터로 결과값이 80% 이상이 일치하는지 확인	85%일치함을 확인
감정분석	결과값으로 감정이 나오는지 확인함	다섯가지 감정이 올바르게 감정분석 출력
감정분석 결과 데이터 연동 확인	감정분석 결과를 저장한 데이터베이스가 웹페이지에서 잘 연동되는지 확인	웹페이지에서 100% 연동됨.
각감정마다 필터링되어 데이터 출력 확인	감정분석 결과가 각 감정별로 웹페이지에 필터링 되는지 확인	웹페이지에서 100%연동됨.
회의록 명사 추출	저장된 회의록에서 명사와 빈도수 추출	회의록에서 나오는 내용중 명사와 각 명사의 빈도수가 추출되어 저장됨.

6. 평가

6.1 작품의 완성도 및 기능 평가

저희가 목표로 하던 감정분석 모델의 정확도는 90% 였습니다. 초기 모델의 정확도는 70% 정도였으나, 수많은 모델 튜닝 시행착오를 거쳐 결과적으로 최종 정확도는 85%까지 향상시킬 수 있었습니다.

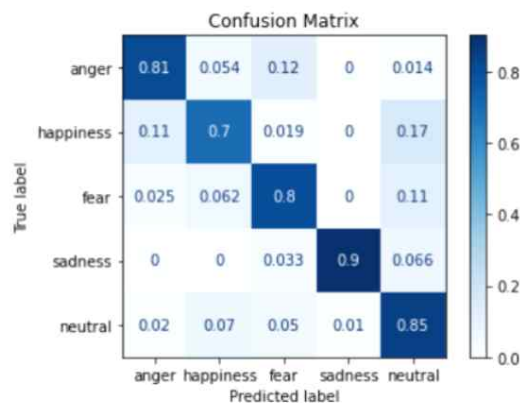


그림1) Confusion Matrix

또한 두 번째 목표인 화상회의 품질 또한 외부 소켓 서버를 사용함으로써 화질은 Ultra-HD 까지 지원하도록 구현하였습니다.

세 번째 목표인 소켓 통신 지연률 또한 여러 테스트를 통해 가장 효율적인 로직을 사용하였고, STT 결과 전송 분리 등 최대한 사용자 경험이 좋을 수 있도록 노력하였습니다.

네 번째 목표인 서비스 배포 또한 성공적으로 마무리 되었습니다. 아래의 주소에서 FLOG를 체험할 수 있도록 하였습니다.

- 웹 사이트 주소 : <https://flog.tk>
- Git Hub 주소 : <https://github.com/Brave-Cookie>

6.2 기대효과 및 영향

6.2.1 기대효과

1) 체계적인 회의 관리 용이성

현업에서 진행되는 회의는 일회성이 아니므로 팀 혹은 프로젝트별로 구분하여 저장하여 관리할 필요성이 있습니다. 사용자는 체계적인 회의 관리 시스템을 원할 것이고, FLOG는 프로젝트별로 회의 및 참가자 관리가 가능하므로 이러한 요구사항을

충족시킬 수 있습니다.

2) 회의 참여도 증진

업무 환경에서 회의 참가자는 자신이 회의에 잘 참여하고 있는지 의문을 가질 수 있습니다. 자신의 참여도를 실시간으로 분석하여 보여주기에 스스로의 피드백이 가능합니다. 또한 모두의 동등한 참여율이 필요한 환경에서 유용하게 활용할 수 있습니다. 전체적으로 모든 참가자의 참여를 유도할 수 있습니다.

3) 감정분석을 통한 집중도 개선

음성을 통한 감정분석 결과를 제공하므로 카메라를 키지 않은 상태에서도 상대방의 감정을 쉽게 알 수 있습니다. 이는 화상회의를 진행할 때 불필요한 화면 주시를 통한 스트레스를 감소시킬 수 있으며, 또한 카메라를 켤 수 없는 상황에서도 효과적입니다. 결과적으로 본질적인 회의 내용에만 집중할 수 있습니다.

4) 빠르고 정확한 회의 내용 파악

단순히 음성을 텍스트화 하는것에 그치지 않고 워드 클라우드 및 회의 요약본을 제공하므로 회의록 전체를 보지 않아도 빠르고 쉽게 핵심을 파악할 수 있습니다. 또한, 회의에 참가하지 못한 팀원이 회의록만을 보고도 회의 당시의 분위기를 생생하게 파악할 수 있도록 감정분석 결과와 음성 TTS를 제공합니다. 감정 TTS를 사용하면 사용자는 긴 회의록을 모두 읽지 않고도 회의 내용을 회의 분위기와 함께 파악할 수 있습니다.

6.2.2 해결방안의 긍정적 및 부정적인 공학적 영향

현재 비대면 협업 사례가 매우 급진적으로 증가하고 있는 와중, 많은 현업 실무자들은 화상회의 협업에 익숙해진 상태이고, 코로나의 종식이 찾아와도 화상회의의 전망은 매우 밝습니다. 하지만 아직까지 올바른 화상회의 문화는 형성되어 있지 않는 상황입니다. 감정분석 기반 화상회의 서비스 FLOG를 사용하면 회의 분위기를 쉽게 파악할 수 있으며 이는 참가자로 하여금 회의 분위기 개선점을 찾기 쉽도록 합니다. 또한 참가자 자신의 참여도 순위를 확인할 수 있어 스스로 빠른 피드백을 찾도록 유도합니다. 이와 같이 FLOG의 주요기능을 잘 활용한다면 올바른 화상회의 문화를 형성하는데 기여할 수 있습니다.

6.3 작품제작 후기

현재 코로나 사회에서 화상회의 같은 비대면 협업 툴의 수요가 생각보다 많이 증가하였습니다. 하지만 수요가 증가한 것에 비해 현재 화상회의 서비스는 여전히 과거와 비교했을 때 큰 발전이 없다는 것을 알게되었습니다. 4차 산업혁명으로 IT 기술이 급진적으로 발전하여 인공지능 및 자동 텍스트화 등 화상회의에 적용할 수 있는 기술이 있지만 이를 적절히 활용하는 서비스는 찾아볼 수 없었습니다.

따라서 저희는 이러한 기술들을 활용하여 화상회의 서비스를 만들고자 하였고, 현재 개발된 오픈소스 기능들의 성능이 매우 뛰어나다는 것을 느꼈습니다. 게다가 오픈 API의 사용법이 간단하여 쉽게 학습 및 적용이 가능했습니다. 이를 통해 오픈소스의 범용성과 편리성을 체감하였고, 이를 적절히 활용하여 어떠한 소프트웨어 서비스도 개발할 수 있다고 생각하는 계기가 되었습니다.

또한 저희는 프론트엔드 및 백엔드 서버를 분리하였습니다. 각 서버간 통신은 Rest API 통신으로 이루어졌고, 이와 같은 구조에서는 프론트엔드 개발자와 백엔드 개발자 간의 소통이 필수적이란 것을 느꼈습니다. 개발 초기엔 별다른 소통 수단 없이 개발을 진행하였으나, 통신 규칙이 생각보다 많다는 것을 알게 되었고, 협업 규칙의 문서화가 필요하다는 것을 느끼게 되었습니다. 따라서 프로젝트를 진행하며 협업 규칙 및 개발 관련 사전을 작성하였고, 이와 같은 협업 문서화의 중요성을 깨닫게 되었습니다.

6.4 팀 개요 및 역할분담

저희 팀은 디지털콘텐츠학과 2명(강전호, 한재원) 및 소프트웨어학과 1명(김수지), 총 3명으로 이루어져 있습니다. 3명의 팀원 모두 웹 관련 개발자를 희망하고 있어 실시간 통신을 다루는 해당 주제를 선택하였습니다. 역할 분담은 자신이 희망하는 분야를 선택하여 진행하였고, 모두가 동일한 성과를 달성하였습니다.

이름	역할
강전호	- 음성 데이터 전처리 및 감정분석 모델 테스트 - 백엔드 Rest API 구축 - 데이터베이스 ORM 구축
김수지	- 감정분석 모델 설계 및 구현 - 프론트엔드 React 구현 - 데이터베이스 설계
한재원	- 개발 총괄 - 실시간 소켓 통신 구현 - 개발 환경 및 배포 인프라 구축

6.5 참고문헌

- 완벽한 IT 인프라 구축을 위한 Docker / Asa Shiho
- Node.js 교과서 / 조현영
- 감정기반 서비스를 위한 통화 음성 감정분석 기법 / 방재훈, 이승룡 2013
- Survey on speech emotion recognition: Features, classification schemes, and databases / Moataz El Ayadi, Mohamed S. Kamel, Fakhri Karay 2010
- Emotional speech recognition: Resources, features, and methods / Aristotle University of Thessaloniki 2006
- 음성신호기반의 감정인식의 특징 벡터 비교 / 신보라, 이석필 2018
- 음성의 감정 인식을 위한 감정 분류기 앙상블 기법 / 서민지, 김명호 2019
- <https://drive.google.com/drive/u/0/folders/14EXlmdxGvrvw-NCvzHw-lmdVJ7lG6pCR> (참고 논문)
- <https://zdnet.co.kr/view/?no=20200413135719> (수요조사 - 라인웍스)
- https://www.e4ds.com/sub_view.asp?ch=23&t=0&idx=11573 (화상회의 수요)
- <https://www.intrado.com/en/cloud-collaboration> (화상회의 단점 기사)
- <https://www.pmi.org/learning/library/successful-virtual-meetings-skills-improvement-6267> (회의록의 중요성 기사)
- <http://www.codingworldnews.com/news/articleView.html?idxno=1403> (empath)
- <http://news.bizwatch.co.kr/article/mobile/2021/04/10/0002> (다글로)
- <https://www.sorizava.co.kr/minute> (소리자바)
- <http://hleecaster.com/ml-svm-concept/> (SVM 이론 1)
- <https://woono.tistory.com/111> (SVM 이론 2)
- http://pigbrain.github.io/math/2015/07/17/LagrangeMultiplier_on_Math(라그랑주 승수법)
- <https://greedywyatt.tistory.com/121> (라그랑주 승수법 2)
- <https://gmlwjd9405.github.io/2018/09/21/rest-and-restful.html> (Rest API)
- <https://withseungryu.tistory.com/129> (Web RTC 관련)
- <https://devji.tistory.com/entry/WebRTC-%EB%9E%80-%EC%86%8C%EA%B0%9C-%EA%B0%9C%EC%9A%94> (Web RTC 이론)
- <https://subicura.com/2017/01/19/docker-guide-for-beginners-1.html> (Docker 이론)
- <https://studyingych.tistory.com/46> (socket의 원리)
- <https://d2.naver.com/helloworld/1336> (Web socket 이론)