



국민대학교  
소프트웨어융합대학  
소프트웨어학부

# 캡스톤 디자인 I

## 종합설계 프로젝트

프로젝트 명	<i>My Mood Music #마무무</i>
팀 명	<i>포춘텔러</i>
문서 제목	결과보고서

Version	1.2
Date	2019-MAY-28

팀원	오윤정(조장)
	박정은
	원수빈
	홍자현
	강은채

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

#### CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인 I 수강 학생 중 프로젝트 "My Mood Music"을 수행하는 팀 "포춘텔러"의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 "포춘텔러"의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

## 문서 정보 / 수정 내역


<b>Filename</b>	중간보고서-My Mood Music.doc
<b>원안작성자</b>	오윤정, 박정은, 원수빈, 홍자현, 강은채
<b>수정작업자</b>	오윤정, 박정은, 원수빈, 홍자현, 강은채

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2019-05-28	강은채	1.1	최초 작성	양식 및 개요, 추진배경, 음성인식관련 기술
2019-05-28	오윤정	1.2	내용 수정	백엔드 및 서버관련 내용 및 관련 기술
2019-05-28	박정은	1.2	내용 수정	3D 모델 및 어플리케이션 관련 기술과 내용
2019-05-28	원수빈	1.2	내용 수정	도표디자인 및 데이터베이스관련 기술
2019-05-28	홍자현	1.2	내용 수정	추천알고리즘관련 기술과 내용
2019-05-28	전체	1.2	내용 수정	사용자 메뉴얼, 자기평가 및 연구개발 내용
2019-05-28	전체	1.2	내용 수정	기술적/비기술적 요구사항, 제한요소 등

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

## 목 차

1	개요	4
1.1	프로젝트 개요	4
1.2	추진 배경 및 5	5
1.2.1	추진 배경 및 필요성	
1.2.2	기술 상황	6
1.2.3	감성 인식 시장 동향	6
1.2.4	유사 어플리케이션 분석과 시스템 경쟁력	7
2	개발 내용 및 결과물	8
2.1	목표	9
2.2	연구/개발 내용 및 결과물	9
2.2.1	연구/개발 내용	
	1)Back-End(server)	10
	2)Front-End	22
2.2.2	시스템 요구사항	31
2.2.3	시스템 비기능(품질) 요구사항	32
2.2.4	시스템 구조 및 설계도	33
2.2.5	활용/ 개발된 기술	35
2.2.6	현실적 제한요소 및 그 해결방안	36
2.2.7	결과물 목록	37
2.3	기대효과 및 활용방안	39
3	자기평가	40
4	참고 문헌	43
5	부록	46
5.1	사용자 매뉴얼	46
5.2	운영자 매뉴얼	46
5.3	테스트케이스	48

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28


# 1 개요

## 1.1 프로젝트 개요

현대인의 우울증 문제가 날로 심각해지고 있다. 과도한 경쟁 사회 속에서 살고 있는 현대인들은 각박한 사회 속에서 살아남기 위해 자신의 감정을 돌보지 못하고 있는 실정이다. 이런 자기 인식의 부재로 인해 요즘 현대인들은 감정 표현에 서툴 뿐 아니라 또한 몹시 어색해 한다.

이런 우울증, 소통 어려움 등의 문제가 더 심화되기 전에, 자신의 감정을 표출할 수 있는 기회가 있어야 할 것이다. 하지만 갑자기 자기 자신을 돌아보고 감정에 대해 인식하게 되는 것은 쉽지 않은 일이다.

본 프로젝트는 위 상황을 해결하기 위한 첫번째 방안으로써 이미지(사용자 표정)와 음성 데이터(사용자 음성)를 통해 사용자의 감정을 분석하고 분석 결과를 이용하여 간단한 설명과 함께 적절한 음악을 추천하는 서비스이다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

## 1.2 추진 배경 및 기술 상황과 시장 동향 분석

### 1.2.1 추진 배경 및 필요성

과도한 긴장과 경쟁에 사로잡힌 사회 속에서 현대인들이 자신의 감정은 잊은 채 바쁘게 살아가고 있다. 이로 인해 우울증, 분노 조절, 소통의 어려움 등의 문제가 나타나기도 한다. 여러 기사들에서는 다음과 같이 표현한다.

“요즘 현대인들은 감정표현에 서투르고 또 몹시 어색해 한다. 이로 인해 회사생활, 인간관계 부분에서 정확한 감정표현과 소통이 되지 않아 어려움을 겪는 사람들이 늘고 있다. 자신의 감정에 신경 쓸 여력이 없어 보인다.”

‘감정표현 잘하는 사람이 행복해집니다’,

<GOODNEWS> , 2015.04.24 19:07, <http://www.igoodnews.or.kr/news/articleView.html?idxno=4363>

“보건복지부의 2016 년 정신건강 실태조사에 따르면 현재 우리나라의 우울증 환자는 60 만명을 넘는 것으로 나타났다. 우울증 환자 수 또한 매년 증가하고 있는 실정이다. 이는 현대사회가 해결해 나가야 할 심각한 사회문제 중 하나이다. 현대인들은 각박한 사회 속에서 살아남기 위해 자신의 감정을 돌볼 시간이 없다.”

‘국내 우울증 환자 60 만명 넘었다’

<한겨레 신문>, 2017-04-06 13:39 ,

<http://www.hani.co.kr/arti/society/health/789572.html>

“우리가 뭔가를 조절하려고 하면 그것에 대해 잘 알고 있어야한다. 자신의 감정에 대해 잘 알지 못할 때 우리는 감정을 조절하기 힘들어진다. 자신의 감정을 관찰하고 이해할 기회가 없었던 사람은 감정을 조절하는 것이 무척 힘들게 된다. 이해가 수용을 만들고 수용이 개방성을 만들어 내듯이 자신의 감정에 대한 이해 없이는 감정을 조절하는 것은 불가능하다. 감정의 조절은 나의 감정을 관찰하고 읽는 것부터 시작되어야 한다. 내 감정을 관찰하고 읽겠다는 의지는 자기존중의 바탕이 되고 자신을 용서하고 위로할 수 있는 힘이 된다.

‘감정 조절을 위한 이해와 연습’, <brunch>, 2016.04.16,

<https://brunch.co.kr/@hesse24/35>

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

이처럼 자신의 감정을 잘 인식하고 있으면 자기 기준이 설정되어 외부세계를 인식하고 상호작용하는데 혼란이 적고 적극적이다. 또한 자신의 기준으로 세상을 볼 때 명확하게 인식하고 판단하고 행동할 수 있다. 이 뿐만 아니라 자기인식은 스스로 무엇을 어떻게 맞추고 조절해야 하는지 알기 때문에 자기존중감과 자신감을 높인다.

따라서 현대인들이 자신의 감정을 좀 더 명확하게 인식하기 위한 시스템 개발의 필요성이 대두되었다.

### 1.2.2 기술 상황


감성 인식 기술이란 사람이 느끼는 감성을 얼굴 표정이나 목소리, 생체 리듬 등을 통해 인지하는 기술이다. IT 기술이 발달하면서 인간의 감성을 읽기 위해 여러 센서부터 인공지능 알고리즘까지 접근 방법이 다양해지고 있다. 이를 '감성 컴퓨팅'이라고 하며, 인공지능을 활용하면 이미지 속의 인물을 식별해 분노, 슬픔, 놀라움, 공포 등 여러 감정 요인을 확률적으로 표현해서 보여준다.

현재 산업에서 감정 인식에 딥러닝을 가장 활발하게 연구하고 적용하는 분야는 사람의 표정을 인식하는 분야이다. 이미지는 데이터의 크기가 크고 복잡도가 높아 딥러닝을 우선 적용할 수 있는 분야이므로 상대적으로 알고리즘의 발전이 빠르다. 하지만 사람이 감정을 표출하는 방식은 성별, 연령, 심지어 시간이나 수면 정도에 따라 다르기 때문에 얼굴 인식만으로는 한계가 있다. 그래서 효과적인 감정 표현을 위해 중요하게 활용되는 다른 준언어적 특성인 억양(말의 높낮이 변화에 의해 나타나는 것)을 함께 활용하기로 한다.

### 1.2.3 감성 인식 시장 동향

감성 ICT 산업은 얼굴, 표정, 음성, 생체 반응 등을 통해 감성을 인지하여 사용자와의 다양한 상호작용에 활용되면서 모바일 기기 및 PC, 게임·자동차·항공 등 여러 산업분야에서 제품의 감성화가 빠르게 진행되고 있다.

감성인식기술 분야의 글로벌 시장은 2015 년 1 조 달러에서 연평균 10%의 높은 성장을 지속해 2020 년에는 1 조 6,000 억 달러에 이르는 초대형 파이를 형성할 것으로 예상하고 있다. 국내 시장도 2015 년 27 조원에서 2020 년에는 43 조원 규모로 성장할 것이란 예측이다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28



\* 자료 : "감성 ICT 산업규모 및 경제효과보고서", 감성ICT산업협회, 2018. 3.  
신원순 외, "감성 ICT 기술 및 산업동향", 전자통신동향분석 제29권 제5호, 한국전자통신연구원, 2014. 10 / 재구성


#### 1.2.4 유사 어플리케이션 분석과 시스템 경쟁력



위의 어플리케이션들은 얼굴 인식, 감정 일기 또는 노래 추천을 주제로 한 소프트웨어들이다. 사용자의 얼굴 사진을 통해 나이, 성별과 감정을 분석해 제공하거나 자신의 감정을 글로 기록하는 등의 유용한 기능을 가지고 있다.

하지만 단순히 얼굴을 분석해 사용자에게 결과를 제공하고 사용자의 감정을 기록하는 기능이 가졌다는 점에서 한계점을 발견하고 본 프로젝트를 진행하게 되었다. 본 시스템은 얼굴 인식을 통한 감정 분석과 음성 인식을 통한 감정 분석을 통해서 사용자의 감정에 맞는 노래를 추천하는 어플리케이션이다. 기존의 어플리케이션들과는 다르게 사용자가 지금 어떤 감정인지 알게 할 수 있고 상황에 맞는 적절한 노래를 추천 받을 수 있다.


우리 어플리케이션은, 감성 인식 기술측면에서 현대인들이 자신의 감정을 제대로 알지도 못하고 바쁘게 살아가는 사회 속에서 본인의 감정을 분석해 바르게 인식하게 해준다. 또한 사람들이 가장

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

친숙하게 접하는 '노래'라는 소재를 활용해서, 분석된 '자신'의 감정에 따른 노래 추천이라는 주제를 가지고있다. 이러한 점에서 경쟁력을 가진다고 본다.

'스노우'라는 어플은 사람들의 얼굴을 인식해 재미있는 스티커를 붙여주는 단순한 기능으로 선풍적인 인기를 누렸다. 우리 프로젝트도 노래 추천하는 도중, 연령을 반영해서 향수를 불러일으킬 노래를 추천하거나 웃는 얼굴로 슬픈 말을 할 때 모순적인 상황에서 거짓말에 대한 노래를 추천하는 것 등 재미있는 요소를 가미하는 것도 또한 어플의 경쟁력을 높일 것이라고 생각한다.



 <div> <p>국민대학교</p> <p>소프트웨어학부</p> <p>캡스톤 디자인 I</p> </div>	결과보고서		
	프로젝트 명	My Mood Music#마무무	
	팀 명	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

## 2 개발 내용 및 결과물

### 2.1 목표

사용자에게 감정 분석 결과를 알려주고 이를 바탕으로 노래를 추천해주는 어플리케이션 제작을 목표로 한다. 감정 분석 결과는 얼굴인식을 통한 감정분석, 음성인식을 통한 억양분석 결과를 종합해 도출해낸다.

사용자가 어플리케이션을 실행하면 카메라 또는 갤러리로 얼굴 이미지를 api 로 전달하여 표정에 관한 감정정보와 추정 나이 정보를 얻는다. 어플리케이션 안의 캐릭터가 질문을 하면 사용자가 직접 대답하고 사용자의 음성을 인식하여 학습된 모델로 억양에 관한 감정정보를 분석해낸다.

이렇게 나온 2 가지 정보(표정에 관한 감정정보, 억양에 관한 감정정보)를 조합하여 최종적인 결과를 도출해내고, 어플리케이션 안의 캐릭터가 감정분석 결과와 함께 적절한 노래를 추천해준다. 노래추천 알고리즘에 사용할 데이터는 앞서 말한 최종적인 결과와 api 에서 전달받은 추정나이가 될 것이다.

어플리케이션의 예상 시나리오는 다음과 같다.



 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

## 2.2 연구/개발 내용 및 결과물

### 2.2.1 연구/개발 내용

#### 1) Back-End(server)

##### 1-1) 서버 구축

##### (1) MySQL 연동

Ubuntu18.04 에서 mysqlclient 를 다운받아 Django Project 의 settings.py 에서 DATABASE host 와 username, password 를 세팅해주었다. mysql 을 사용하여 database 를 생성하고, root 계정으로 들어간 후 user 를 생성하여 grant 권한을 주었다. 그리고 admin 사이트에 로그인 할 수 있는 superuser 를 생성하여 사이트에 접근 가능하게 하였다.

어플 개발환경이 유니티이기 때문에 손쉽게 테스트를 해보려면 Windows OS 환경이 필요하여 Windows 에서도 MySQL 을 다운받아 직접 연동하였다.

그리고 Django shell 과 dbshell 을 이용하여 데이터베이스 쿼리문을 날리며 테스트를 해보았다. 한글 인코딩 문제 때문에 character set utf8 로 세팅하였다.

##### (2) 기능 연결

함수형으로 작성한 음성인식 코드와 추천 알고리즘을 모두 클래스형 뷰로 바꾸었다. 클래스형 뷰를 사용하는 것이 Django 가 제공하는 제네릭 뷰를 사용할 수 있고 재사용 및 확장성 측면에서 유리하다고 판단되었기 때문이다.

이미지 인식을 통한 결과와 음성 인식을 통한 결과를 추천 알고리즘에서 필요로 하기 때문에 그것을 파일에 각각 저장한다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

global 변수가 아니라 파일로 저장하는 이유가 있다. 각 클래스형 뷰는 지정된 url 로 request 가 왔을 때 실행되는데, 네트워크 통신 오류로 인해 이미지 인식, 음성 인식을 한 후에 서버가 reloading 되면 global 변수에 넣은 값들은 다 날아가 버리기 때문이다. 그래서 'face\_api\_emotion.txt', 'face\_api\_age.txt', 'speech\_api\_emotion.txt' 를 사용하여 저장된 값을 불러오는 방법을 사용했다.

### (3) 어플과의 통신 흐름을 고려한 URL 설계 및 기능 구현

어플과의 통신 흐름에 따라 url 을 설명하겠다.


사용자가 어플에서 로그인했을 때, log in 버튼을 누르면 "host:8000/getauthtoken/" url 을 통해 GetAuthToken view 가 실행되고, resT\_framework.authtoken.models.Token 을 이용해 사용자에게 토큰을 보내게 된다. 그리고 사용자의 데이터를 CreateUserSerializer 를 이용하여 DB 에 저장한다.

그 이후 어플에서 사용자의 감정을 분석하기 위한 사진을 찍는다. 사진을 찍어서 POST Method 를 "host:8000/face/" 로 보내게 된다. 그러면 RequestFaceAPI 클래스의 post 함수로 연결되어 이미지를 통한 감정분석을 실행한다.

이후에는 어플에서 사용자의 감정을 분석하기 위한 음성을 녹음한다. 음성을 녹음한 후 POST Method 가 "host:8000/speech/" 로 보내지고, Call 클래스의 post 함수로 연결되어 음성을 통한 감정분석을 실행한다.

그리고 어플에서 "host:8000/recommend" 로 GET Method 를 보내면 RecommendationMusic 클래스의 get 함수로 연결되어 추천알고리즘에 따라 music 정보와 그에 따른 url 정보, 또 tag 정보들을 3 쌍을 받게 된다.

Host IP Address 에 8000 포트로 접근하면 기본적으로 rest\_framework.router 를 이용해 등록한 UserViewSet 이 나오게 된다. 거기에서 "host:8000/users" 링크를 누르게 되면 지금까지 로그인한 사용자의 데이터를 불러온다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

#### (4) 데이터베이스 접근(create, filter), queryset

Django 의 Model Manager 를 통해 queryset 을 작성하고, 별도로 SQL 을 작성하지 데이터베이스로부터 데이터를 가져오고 추가하는 기능을 구현하였다.

Model 이 테이블이라면 Model.objects.filter()로 그 테이블에 있는 조건에 부합하는 엔트리들을 필터링할 수 있고, queryset = Model.objects.create(), queryset.save()를 이용하여 테이블에 추가할 수 있다.


RecommendationMusic class 에서 get 함수가 호출되면, 사용자의 age 를 받아서 age 가 10 보다 작으면 Child 테이블에서 랜덤으로 값을 3 개 가져온다. 10 세 이상이라면 추천 알고리즘을 거쳐서 노래제목과 가수, youtube url, tag 정보를 가져오는 queryset 을 작성했다. 그리고 어플에서 받은 user 의 id 값과 추천 알고리즘으로부터 받은 노래 제목 3 가지를 Analysis\_Result 테이블에 추가하는 queryset 을 작성했다.

데이터베이스 테이블 레코드를 리스트로 받아올 때 pk 값을 랜덤으로 (1 부터 그 테이블의 사이즈까지) 가져오는데, 아예 랜덤으로 가져오면 나중에 pk 값이 겹칠 수 있기 때문에 pk\_list 를 사용하여 쌓아둔 후 중복되지 않은 값들로만 가져오는 방법을 사용하였다.

#### (5) Django REST Framework

RESTful API 구현을 위해 Django REST Framework 를 사용했다. Serializer 를 이용하여 쿼리셋에 해당하는 정보를 JSON 으로 매핑했다. 어플에서 원하는 정보를 queryset 으로 받아온 다음, 그에 해당하는 값을 JSON 데이터로 직렬화 하는 것이다. 쿼리셋을 작성한 Serializer 에 입력으로 주면, 그 값의 data 를 받아 어플에 Response 해주는 기능을 구현하였다.

그리고 Django REST Framework 에서는 인증방식을 여럿 제공하는데, 그 중 코튼 인증 방식을 사용하기 위해 settings.py 에 rest\_framework.authtoken 을 추가하였다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

또, 커스텀 미들웨어를 구현하여 Django REST Framework 를 위한 http response formatting 을 하였다. middleware.py 파일에 있는 Unity3DMiddleware 에 해당한다. 미들웨어란 http request/ response 처리 중간에 작동하는 시스템이라고 볼 수 있다. process\_response 메소드를 구현하여 http 응답을 포매팅 해주는 로직을 구현했다.

## 1-2) 이미지 인식을 통한 감정분석

어플에서 이미지를 WWWForm UnityWebRequest 로 보낸 것을 RequestFaceAPI Class View 를 이용해 post 메소드에서 request.FILES 객체를 이용하여 Binary Data 로 받는다. Microsoft Cognitive Services Face API 의 key 를 받아오고, 이 API 는 input 으로 Image url, Binary data, Json data 를 허용하므로 받은 binary data 자체를 input 으로 넣는다. API 결과는 string 형태로 나오고, 서버가 reloading 됐을 때 결과들이 사라지는 것을 방지하여 그 결과값을 'face\_api\_emotion.txt' 파일에 저장한다.

얼굴이 인식되지 않은 경우에는, API 결과값이 None 이 되므로 그 때는 어플에게 HttpResponse 를 준다. ("please try again")

그렇지 않고 얼굴이 잘 인식되었을 경우에는, get\_data\_from\_faces 메소드를 부른다. 이 메소드에서는 string 형태의 결과값을 json 모듈을 이용하여 json 데이터로 만든 후 key 값이 'faceAttributes' 인 것에서 'emotion', 그리고 'age' value 들을 찾아서 따로 저장해 놓는다.

```
emotions = dict['faceAttributes']['emotion']
age = dict['faceAttributes']['age']

print('emotions : ', emotions)
print('age : ', age)
```

저장한 후 value 가 0.0 이면 버리고 0.0 보다 큰 값들만을 list 에 저장한다. 그리고 이 age

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

값을 추천 알고리즘에 적용해야 하기 때문에 또 파일에 저장한다. 그리고 어플의 태깅 기능을 위하여 0 보다 크게 나온 감정 값들을 list of string 형태로 Response 를 보낸다.

```
emotions : {'anger': 0.0, 'contempt': 0.099, 'disgust': 0.073, 'fear': 0.0,
'happiness': 0.0, 'neutral': 0.012, 'sadness': 0.816, 'surprise': 0.0}
age : 25.0
[('anger', 0.0), ('fear', 0.0), ('happiness', 0.0), ('surprise', 0.0), ('neutral', 0.012), ('disgust', 0.073), ('contempt', 0.099), ('sadness', 0.816)]
[('disgust', 0.073), ('contempt', 0.099), ('sadness', 0.816)]
```

### 1-3) 음성인식을 통한 감정분석

이 단계에서는 음성데이터를 이용한 감정분석을 수행함으로써, 얼굴 사진을 이용한 감정분석의 정확도를 좀 더 높이는 것을 목표로 한다. 캐릭터가 사용자에게 질문을 하면 사용자는 대답하고, 이 대답음성이 인식 되면 미리 학습해둔 모델에서 감정분석 결과를 얻게 된다. 이 결과는 얼굴 인식 결과와 종합되어 감정 분석 최종결과에 기여한다.

CNN(합성곱 신경망)으로 구현하여 히든 레이어 3 개(300-200-100)을 시그모이드, 렐루, 하이퍼볼릭탄젠트의 성능을 적절히 비교하여 배치하고 더 나아가 히든 레이어 개수를 조정하여 정확률을 높인다.


학습을 시키기 위한 데이터는 IEMOCAP 과 RAVNESS 에서 얻는다. 12시간의 분량과 1500개의 wav 데이터를 활용하여 구현한 모델에 학습을 시킨다.

```
#1차 hidden layer
W_1 = tf.Variable(tf.random_normal([n_dim, n_hidden1], mean = 0, stddev = sd), name="w1")
b_1 = tf.Variable(tf.random_normal([n_hidden1], mean = 0, stddev = sd), name="b1")
h_1 = tf.nn.relu(tf.matmul(X, W_1)+b_1)
#시그모이드함수, 탄젠트하이퍼볼릭과 비교
```

<그림 1 1차 hidden layer 의 예시>

하지만 데이터의 부족과 모델의 정확률이 어플리케이션의 완성도에 미칠 영향을 고려하여 [github\(https://github.com/MITESHPUTHRANNEU/Speech-Emotion-Analyzer\)](https://github.com/MITESHPUTHRANNEU/Speech-Emotion-Analyzer)

에 올라온 Speech-Emotion-Analyzer 를 사용하기로 한다. SEA 는 wav 파일의 파형과

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

스펙트럼을 분석하여 감정을 도출하는데 CNN 알고리즘을 사용하며 다층 퍼셉트론 신경망 구조를 가진다. 70%의 정확률을 가지는 이 모델의 자체 성능평가에서 EngDataTest 35.1%, KorDataTest 14.3%의 일치율을 보였다. 영어 음성 데이터로 학습된 모델이라 한국어에서 정확률을 얻기는 힘들어 이미지 인식 API 와 최종적으로 결과를 도출할 때 weight 값을 적게주어 '거짓말' 판별을 할 때 쓰일 수 있도록 독립적으로 추천 알고리즘에 반영되도록 하였다.

그 다음 이 모델이 어플리케이션에서 서버로 음성 데이터를 보내왔을 때 작동될 수 있도록 인식 결과가 웹에서 보여지도록 테스트 한다. 모델을 실행하는 함수 labelfrommodel 로 정의하고 그 인수로 파일이름을 지정한다. 'file.wav'파일로 입력이 되었을 때 emotion label 의 값을 HttpResponse()로 화면에 표시한다.

visual studio 에서 기본으로 제공하는 django 환경을 이용하여 모델테스트를 마치고 pyDjango 프로젝트에서도 모델이 작동할 수 있도록 코드를 이식한다. unity 를 설치하여 application branch 의 파일로 어플을 실행하면서 파일이 서버에서 받아져 모델이 결과값을 내는지 확인한다.

어플리케이션에서 audioRecordFile 을 byte array 형식으로 보낸다. 이 file 을 가져와 .wav 형식으로 저장하여 모델에 넣어주는 것이 바람직하다.

```
audio_file = request.FILES.get('audio','')

if os.path.isfile('./media/file.wav'):
    os.remove('./media/file.wav')
else:
    path = default_storage.save('file.wav', ContentFile(audio_file.read()))
```

이 때 이전에 테스트 했던 file.wav 가 이미 존재하면 다른 이름으로 파일이 생성된 후 실행되므로 모델에 파일의 path 를 넣어주어야 한다. 모델은 filename 을 인수로 작동함으로 해당 폴더에 file.wav 의 존재여부를 확인 한 후 저장 코드를 실행 한다.

#### 1-4) database

본 프로젝트의 어플리케이션이 작동하기 위해 설계된 데이터베이스 테이블은 다음과 같다.

: USER\_INFORMATION, HAPPINESS, SADNESS, ANGER, FEAR, DISGUST, SURPRISE,

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28


SUBCLASS\_SAD, ANALYSIS\_RESULT, LIE, CHILD

USER\_INFORMATION 테이블에는 회원가입 시 입력 받은 사용자의 이메일과 패스워드가 저장된다. 더불어 어플리케이션의 마지막 단계인 사용자 만족도 조사에 대한 결과값도 저장된다. emotion 테이블에는 HAPPINESS, SADNESS, ANGER, FEAR, DISGUST, SURPRISE 와 같이 6 개의 테이블이 있는데, 이는 심리학자들이 정의한 인간의 6 가지 감정을 기준으로 하였다. SADNESS TABLE 에 대해서는 노래 추천 알고리즘에 따라 SUBCLASS\_SAD TABLE 을 이용하여 노래들을 다음과 같이 3 가지로 분류하였다.

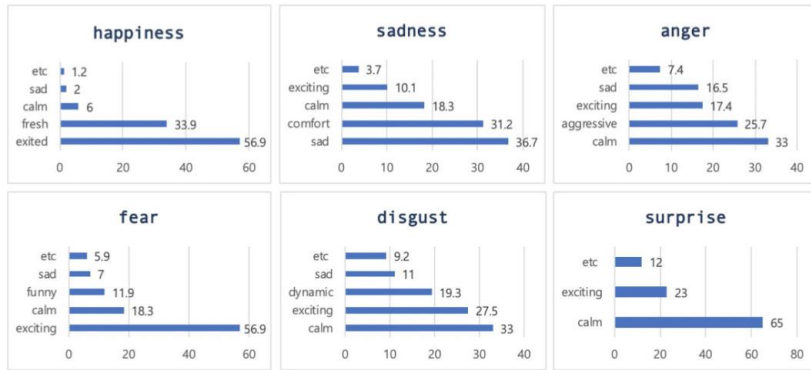
→ (1) 잔잔한 노래 (2) 슬픈 노래 (3) 기분전환 노래

ANALYSIS\_RESULT 에는 사용자의 이메일과 함께 이전 분석 결과 값에 따라 선정된 노래들을 담아두는데, 이는 사용자들이 자신이 이전에 추천 받은 노래들을 다시 확인하고 싶을 때 사용된다. LIE TABLE 은 얼굴과 음성이 상반된 결과로 분석될 때 사용되는 테이블로, 거짓말과 관련된 노래들이 담겨있다. CHILD TABLE 은 입력 받은 사용자의 얼굴(이미지)의 분석된 결과가 10 세 이전으로 판단될 때 사용된다. 데이터베이스는 Django REST Framework 를 사용하여 설계하였다. 감정 별로 노래 데이터를 리스트화 하기 위해 노래를 분류하는 기준을 정해야 했다. 따라서 "감정 별로 어떠한 노래를 듣는지" 에 대한 설문을 진행하였고, 설문 결과와 음원 스트리밍 사이트(벅스, 멜론, 지니)의 태그 서비스를 통해 노래를 분류하기 위한 정책을 정하였다.



 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

#### RESEARCH - “당신은 감정 별 상황일 때 어떠한 노래를 듣습니까?”



<표> 설문조사 결과

- Happiness : 신나는 노래, 발랄하고 상큼한 노래
- Sadness : 슬픈 노래, 위로가 되는 노래
- Anger : 진정이 되는 노래, 공격적인 노래(화를 표출할 수 있는 노래)
- Fear : 상황을 반전시켜주는 밝은 노래, 진정이 되는 노래
- Disgust : 잔잔한 노래, 신나는 노래
- Surprise : 진정이 되는 노래

내부적으로 정한 정책에 맞게 감정 별로 나눈 노래는 csv 파일에 리스트화 해놓고, python 코드를 사용하여 Django server 에 한꺼번에 올렸다. 다음은 HAPPINESS TABLE 에 신나는 분위기의 노래가 들어있는 모습이다.

<input type="checkbox"/>	ID	MUSIC	AGE	LINK	TAG 1	TAG 2
<input type="checkbox"/>	54	방탄소년단 - 불타오르네	20	<a href="https://youtu.be/ALj5MKjy2BU">https://youtu.be/ALj5MKjy2BU</a>	내적댄스	아이돌
<input type="checkbox"/>	53	T-ARA(티아라) - Roly Poly	20	<a href="https://youtu.be/afwK0Mv0IsY">https://youtu.be/afwK0Mv0IsY</a>	중독성	비트
<input type="checkbox"/>	52	박진영 - 어머니가 누구니	20	<a href="https://youtu.be/kUGQ7Tz4os0">https://youtu.be/kUGQ7Tz4os0</a>	댄스곡	홍폭팔
<input type="checkbox"/>	51	임창정-문을 여시오	30	<a href="https://youtu.be/FLPLgJqeZJw">https://youtu.be/FLPLgJqeZJw</a>	댄스댄스	신나는
<input type="checkbox"/>	50	PSY - RIGHT NOW	20	<a href="https://youtu.be/UmHdefsaL6l">https://youtu.be/UmHdefsaL6l</a>	댄스	신나는
<input type="checkbox"/>	49	PSY - HANGOVER (feat. Snoop Dogg)	20	<a href="https://youtu.be/HkMN0IYcpHg">https://youtu.be/HkMN0IYcpHg</a>	홍겨운	비트

#### 1-5)추천알고리즘

Microsoft Face API 의 얼굴 이미지를 이용한 감정 분석 결과와 Speech Emotion Analyzer 의 음성(억양)을 이용한 감정 분석 결과를 추천 알고리즘의 입력으로 넣는다. 본 시스템의 추천 알고리즘에 따라 노래를 1 순위에서 3 순위까지 추천한다.

추천 알고리즘 설계와 알고리즘 구성도에 대한 자세한 내용은 다음과 같다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

- (1) 얼굴 이미지를 이용한 감정 분석 결과는 다음과 같은 json 형식이다. 추천 알고리즘의 첫 번째 인자가 된다. 알고리즘 내부에서는 "Attributes"속성의 "emotion"속성에 대한 값들을 사용한다. 이들은 8 개의 감정(anger, contempt, disgust, fear, happiness, neutral, sadness, surprise) 과 그에 대한 비율이다.

```
{
  "userId" : "f70c4331-f391",
  "Attributes" : {
    "age" : 13.0,
    "emotion" : {
      "anger" : 0.0,
      "contempt" : 0.001,
      "disgust" : 0.0,
      "fear" : 0.0,
      "happiness" : 0.206,
      "neutral" : 0.793,
      "sadness" : 0.0,
      "surprise" : 0.0
    }
  }
}
```

- (2) 음성(억양)을 이용한 감정 분석 결과는 angry, calm, fearful, happy, sad 중 하나이다. 추천 알고리즘의 두 번째 인자가 된다. Speech Emotion Analyzer 의 성능이 70%임을 고려해 Face API 의 결과를 보조하는 역할로 사용하기로 한다.
- (3) 인간의 기본 감정을 분류해놓은 지침이 있다. 우리의 행동을 지시하고 신속하게 움직일 수 있게 하는 등 중요한 역할을 하는 감정들 중, 가장 중요한 감정인 '놀라움, 혐오, 공포, 행복, 슬픔, 그리고 화'를 기본 감정이라고 부른다.[\(reference\)](#) 이 지침에 따라서 Face API 의 결과인 감정 8 개(anger, contempt, disgust, fear, happiness, neutral, sadness, surprise) 를 'surprise, contempt+disgust, fear, happiness, sadness, anger'의 6 개의 감정으로 새로이 분류한다. 따라서 이 6 개의 감정 상황에 따라 알고리즘을 수행하면 된다. 8 개의 감정이 어떻게 6 개로 추려지는지는 (4)의 정책에서 다시 기술하도록 한다.


 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

(4) Database 에 존재하는 테이블 목록 중 고려해야하는 테이블은'HAPPINESS, SADNESS, ANGER, FEAR, DISGUST, SURPRISE, LIE, CHILD' 테이블이다. 전자인 6 개의 테이블들은 각각 6 개의 감정인 happiness, sadness, anger, fear, contempt\_disgust, surprise 에 대한 테이블이고,'LIE' 테이블은 얼굴인식을 통한 감정분석결과와 음성(억양)인식을 통한 감정분석결과가 서로 모순적인 양상일 때 접근하는 테이블이며,'CHILD' 테이블은 영유아를 위한 동요테이블이다. 이를 기반으로 한 추천 알고리즘의 정책은 다음과 같다.

ㄱ. Database 의 테이블 중 'CHILD'테이블에는 0~9 세 영유아를 위한 동요가 들어있다. 우리 정책에서는 동요의 경우, 감정을 고려하는 추천 알고리즘을 통하지 않기로 한다. Server 에서 Face API 의 결과 중 연령을 파악해 0~9 세에 해당하면 추천 알고리즘을 거치지 않고 'CHILD' 테이블에 접근한다. 따라서 추천 알고리즘에서는 이에 대한 처리를 고려하지 않는다.

ㄴ. 추천 알고리즘의 성능을 높이기 위해서는 몇 가지 전처리를 거쳐야한다. 첫 번째로, (3)에서 기술한 8 개의 감정(화, 경멸, 역겨움, 공포, 행복, 중립, 슬픔, 놀람 / anger, contempt, disgust, fear, happiness, neutral, sadness, surprise)을 6 개(surprise, contempt+disgust, fear, happiness, sadness, anger)로 추려야한다. 이를 위해서 Face API 의 결과 중'neutral'이 0.8 이상이면 3 개의 랜덤 노래를 추천하고, 그 외의 경우는 이 항목을 무시한다. 그리고 Face API 테스트에서 'contempt'와 'disgust'가 함께 등장할 확률이 75%임을 고려해 이 둘의 비율을 합쳐 'contempt\_disgust'항목으로 사용한다. 두 번째 전처리는 특정 값을 넘지 않은 비율에 대해 0.0 으로 스무딩하는 것이다. 낮은 값에 의해 알고리즘의 성능이 저하되는 현상을 방지할 수 있을 것이다. 스무딩하는 기준값은 0.05 로 지정해두고 테스트를 통해 알고리즘의 성능을 높이는 방향으로 수정해갈 것이다.

ㄷ. 모순적인 상황에서'LIE' 테이블에 접근하도록 해야한다. 이 상황에는 '슬픈 얼굴을 하고 행복한 목소리로 말하기', '행복한 얼굴을 하고 슬픈 목소리로 말하기'등이 포함된다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

ㄹ. 전처리를 마친 후에는 1 순위 감정을 중심으로 알고리즘을 처리한다. 사용자가 특정 감정일 때 선호하는 노래는 각기 다르기 때문에, 다양성을 고려해 여러 분위기의 노래를 3 가지로 고루 추천하기로 하였다. 예를들어, '슬픔' 감정일 때, 추천할 노래 분위기의 종류를 3 가지로 만든다. 각각 잔잔한 노래인 '슬픔(1)음악', 너무 슬퍼서 울고싶은 노래인 '슬픔(2)음악', 기분전환을 할 수 있는 밝은 노래인 '슬픔(3)음악'이다.

ㅁ. 이때 '슬픔(1)음악'인 잔잔한 노래는 다른 감정 결과일 때도 사용될 수 있으므로 추천 상황에 따라서 활용한다. 예를 들어, '공포'감정에서 잔잔한 노래를 추천해줄 때 Database 의 슬픔테이블에 접근해 노래를 가져올 수 있다. (Database 의 슬픔 테이블에는 앞에서 기술한 3 종류의 분위기에 따라 노래들이 소분류되어 있다.)

ㅂ. 분석된 감정이 복합적으로 나올 수 있으므로 경우에 따라 2 순위 감정과 3 순위 감정까지 고려할 수 있다. 만약 감정이 3 순위까지 나오지 않아 노래추천에 문제가 생길 시, 1 순위 감정을 다시 활용한다.

ㅅ. 다양한 추천노래 중 사용자의 선택에 도움을 주기 위해, 추천된 노래 옆에 노래의 분위기를 알 수 있는 해시태그(#)를 넣기로 한다.

(5) 위의 사항을 고려해 추천 알고리즘을 설계한다면, 감정마다의 추천 노래는 다음과 같다. 아래에 첨부된 알고리즘 구성도에서 부연 설명이 필요한 것만 기술하였다.

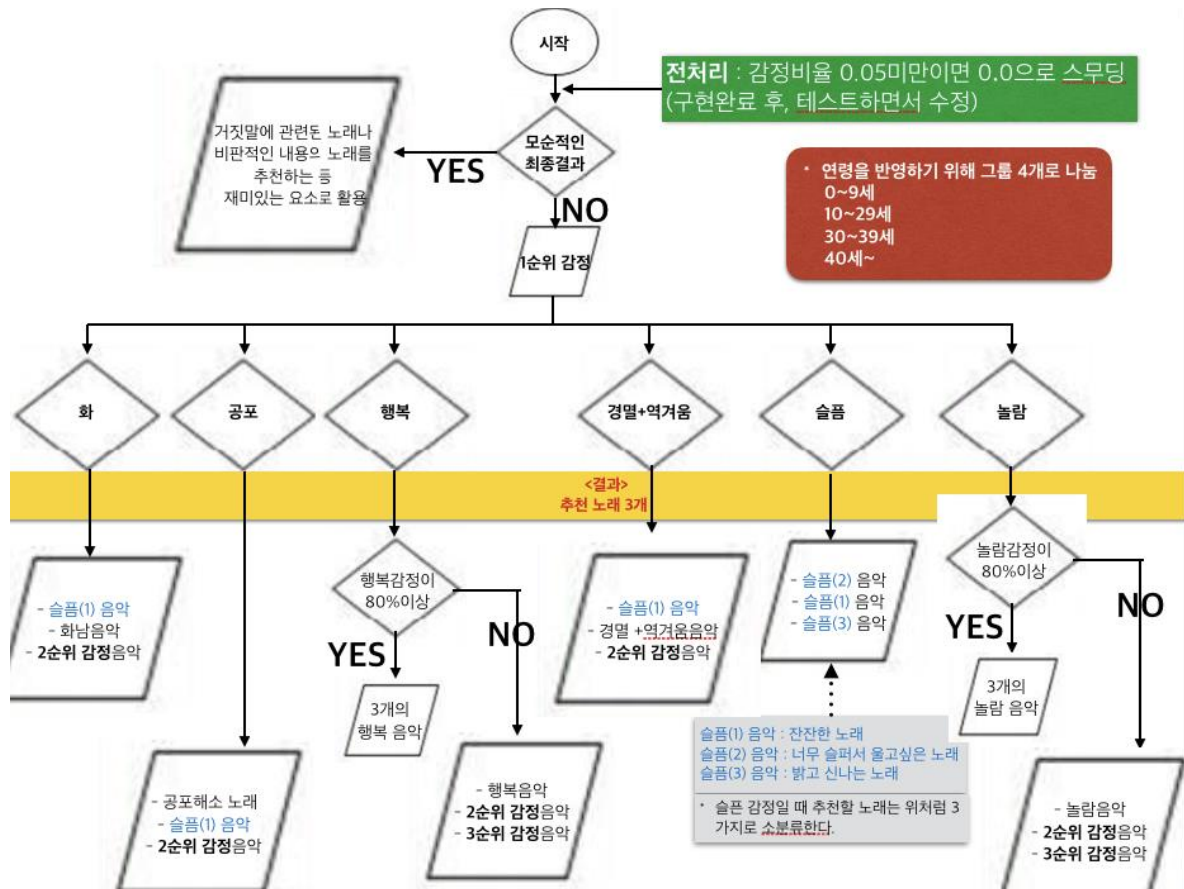
ㄱ.'화'감정일 때 추천 노래들은, [1]화남을 진정시킬 수 있는 Database 의 슬픔 테이블의 잔잔한 노래(이하 '슬픔(1)음악'), [2]Database 의 화남 테이블의 웃긴 분위기의 노래(이하 '화남음악'), [3]2 순위 감정음악

ㄴ.'공포' 감정일 때 추천 노래들은, [1]공포를 해소하는 노래, [2]슬픔(1)음악(잔잔), [3]2 순위 감정음악

다음은 추천 알고리즘 정책에 따른 알고리즘 구성도이다.



프로젝트 명	My Mood Music#마무무	
팀 명	포춘텔러	
Confidential Restricted	Version 1.2	2019-MAY-28



 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

## 2) Front-End

### 2-1) 3D animation

어플리케이션에 3D 캐릭터를 띄워서 사용자와 대화를 주고 받는 듯한 어플을 제작하기로 기획했다. 3D 캐릭터 제작을 위해 Maya 를 사용하여 3D 모델링과 해당 캐릭터의 애니메이션을 제작했다.

- (1) 3D 모델은 Low Poly 로 제작한다. 그리고 나서 최종 렌더를 하기 전, smooth 를 거는 방식이다. 같은 방식으로 Low Poly 로 제작한 후 smooth 를 걸어 rendering 을 진행했다.


3D 모델의 애니메이션은 토폴로지에 영향을 많이 받는다. 토폴로지가 잘못 되어있다면 애니메이션을 진행할 때 모델이 깨지기 쉽상이다. 그래서 후에 발생할 수 있는 현상을 막기 위해서 처음 제작부터 토폴로지를 최대한 고려하면서 제작을 진행했다.

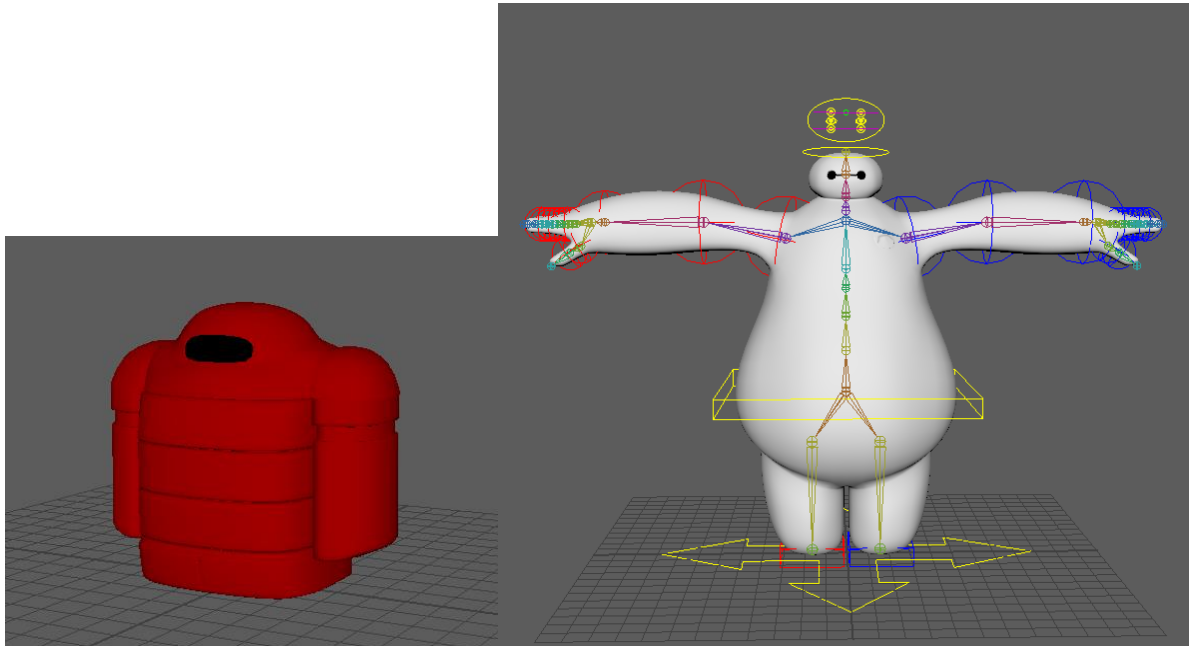
- (2) 모델에 어떤 이미지나 색을 입히기 위해서는 uv 를 펴는 작업이 필요하다. 내가 자른 라인을 기준으로 unfold 를 하기 때문에 후에 입힐 텍스처를 생각했을 때 이어져야 하는 텍스처가 필요하면 해당 부분은 잘라지지 않아야 한다는 걸 생각하면서 uv 를 폈다. 그리고 해당 값을 포토샵으로 create 해서 텍스처링 작업을 한 후 해당 모델의 material 로 생성 후 적용해줬다.

- (3) 모델에게 움직임을 주기 위해서는 먼저 rigging 이라고 하는 모델에 뼈를 박는 작업을 해야한다. rigging 을 시작하기 전 먼저 모델의 뼈 구조를 정해야 한다. 사람과 마찬가지로 뼈가 어떻게 들어가 있느냐에 따라서 모델의 움직임을 달리 정할 수 있다. 전체적으로 모델이 사람과는 크게 다른 형태가 아니라서 사람 rigging 과 비슷하게 진행했다. 하지만 제작한 모델은 무릎이 없어 다리가 굽어지지 않게 진행했다. 다리는 humanIK 를 사용했다. rigging 에는 IK 와 FK 있는데 무게중심, 중심축이 어디냐에 따라서 이어진 나머지 뼈들의 움직임이 달라진다.

	결과보고서		
	프로젝트 명	My Mood Music#마무무	
	팀 명	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

- (4) 뼈 구조를 정의해줬다면 해당 관절들을 움직이게 할 수 있는 컨트롤러를 만들어줘야 한다. 컨트롤러는 뼈와 뼈가 이어지는 조인트에만 만들 수 있다. 또한 컨트롤러는 어떤 관절에 따라서 제한을 둘 수 있는데, 예를 들어 어깨는 이동은 불가하고 회전만 가능하게, 하는 식으로 모든 컨트롤러들을 알맞은 제한들과 함께 만들어줬다.
- (5) 캐릭터 애니메이션에서 빠질 수 없는 부분은 얼굴의 움직임이다. 제작한 모델은 눈만 있는 캐릭터라 얼굴 애니메이션은 눈을 가지고 할 수 있는 애니메이션을 제작했다. 얼굴 움직임은 facial rigging 으로 진행했다. 여러 얼굴 표정을 만들어놓고 Blend shape 를 이용해서 설정을 해주었다. blend shape 는 토폴로지가 같은 모델이면 서로의 모양으로 mapping 을 해주는 기능이다. 그래서 모델의 눈 움직임을 구현하기 위해 blend shape 와 driven key 를 이용해서 제작했다. 좀 더 수월한 컨트롤을 위해 얼굴 움직임에도 컨트롤러를 만들어주었다.
- (6) 모델에 어떠한 변형을 주는 것을 보통 deformer 라고 하는데 해당 변형의 값의 정도를 조절해줄 수 있다. 컨트롤러 또한 내가 어느 정도로 변형을 줄 지 setting 을 해주어야 한다. 해당 setting 은 rigging 한 것과 model 을 skin bind 를 하면 skin weight 를 이용해서 해당 컨트롤러가 어디까지 영향을 미치는지 조절할 수 있다. weight 설정을 잘못하면 변형이 의도와는 다르게 생기게 때문에 많은 수정을 거쳐서 setting 을 완료했다.
- (7) connection editor 와 driven key 를 사용해서 애니메이션 제작을 진행했다. 애니메이션을 만들기 전 미리 조절할 attribute 들을 channel box 에 추가했다.
- (8) 초반 제작 기획으로 케이스에서 깨어나는 애니메이션을 제작하기로 했다. 그래서 추가적인 모델링으로 케이스를 제작했다. 케이스에도 필요한 deformer 와 컨트롤러를 달고 channel box 의 attribute 들을 이용해서 케이스가 열리는 애니메이션을 제작했다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28



## 2-2) application

Unity 를 사용하여 application 제작을 진행했다.


### (1) Login/Signup

로그인과 회원가입은 switch 되는 화면으로 구현이 되어있다.

login 이 불가하다면 회원가입을 진행한 후 로그인을 할 수 있다. 회원가입은 서버에 username 과 email, password 를 보내 db 에 저장하는 방식으로 진행이 되고 성공적으로 저장이 되었다면 해당 값 들로 로그인을 할 수 있다.

로그인을 하지 않고도 어플 진행은 가능하지만 로그인을 했을 경우, 지금까지 내가 진행했었던 감정 분석 결과들에 대한 전체 값을 다시 볼 수 있다. 해당 값은 login toggle 에 표시되고 버튼들의 scroll view 다. 진행한 감정 분석의 값을 사용자 id 에 따른 값들로 저장이 되어있기 때문에 서버에 id 로 요청을



 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

해 전체 값을 json 형태로 받아와 prefab 으로 만들어져 있는 button 을 clone 해서 생성한다.

## (2) 어플의 코드 구성

어플에는 Base Menu 와 Base Game 이라는 스크립트가 존재한다. 기본적인 게임의 서비스는 base game 에 구현이 되어있고 전체적인 기능들에 대한 코드는 Base Game 을 상속받은 Baymax Game Script 에 구현이 되어있다.

Base Menu 에는 기본적인 Menu 들을 위한 기본 코드가 구현이 되어있고 delegate 가 존재해서 Base Menu 를 상속받은 여러 Menu script 들에서 사용한다. 서버로 요청을 하는 코드는 backend manager script 와 example backend 에 구현이 되어있다. backend manager 는 서버에 요청과 response 를 하는 기본 구조 코드로 구현이 되어있고 example backend 는 backend manager 의 partial class 로 필요한 여러 요청 별로 구현이 되어있다.

## (3) 어플 진행 흐름

이런 전체적인 어플의 진행은 대다수가 delegate 로 구현이 되어있다.

보통의 진행 흐름은 어떤 기능이 실행되면 해당 결과 값을 담아 서버로 요청한다. 서버는 같이 온 command 별로 기능을 수행하고 해당 값을 response 로 보내준다. 어플에서는 해당 response 를 받은 후 downloadhandler.text 를 이용해 값을 받아온다. photo 와 audio 는 string 값으로 받아와 split 을 진행해 해당 값을 사용하고 감정 분석 결과 요청은 json 으로 받아와 사용한다.

서버의 response 를 기다리는 함수들은 coroutine 으로 구현을 했고 해당 코루틴을 시작하는 함수들을 구현해서 delegate 로 실행을 하는 형식이다.

사진과 음성이 값을 도출하지 못했을 경우 어플에서 여러번 요청을 하기

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

때문에 bool 변수로 성공했을 때 true 가 되게 구현했고 다른 곳에서도 접근이 가능하도록 signal tone 을 이용해 구현했다. 해당 값이 true 가 되는 시점을 잡는 코드는 코루틴의 yield return 의 waituntil 을 이용하여 람다식으로 해당 값이 true 가 되었을 때 다음으로 넘어갈 수 있게 구현했다.

enum 으로 각각의 state 를 정의해 놓았고 switch 문을 사용해서 해당 state 가 불리면 각각 알맞은 기능을 호출하도록 구현했다.

대화창 클릭을 통해 다음 state 로 넘어갈 수 있는데 대화창을 button 으로 구현하여 button 이 클릭되면 click 변수가 true 가 된다. 하지만 다른 기능을 실행하고 있을 때 클릭을 한다고 해서 다음 state 로 넘어가선 안되기에 locked 변수를 하나 더 만들어 어떤 state 에 들어가 있다면 lock 이 걸려 다음 state 로 넘어갈 수 없게 구현했다.

#### (4) 사진 촬영

사용자의 얼굴이 화면상에 표시된다면 본인의 얼굴을 보면서 표정을 가다듬을 가능성이 있다. 그래서 카메라 화면을 어플 상에 띄우지 않기로 했다. 사진 촬영 시 카메라는 실행되어서 사진을 찍되 사용자가 해당 카메라 화면을 볼 수는 없다. 해당 이미지 정보는 바이트 배열로 변환되어서 서버로 전송이 되고 microsoft 의 face api 로 전해져 감정 분석을 진행한다. 만약 원하는 감정 분석 결과값이 나오지 않는다면 3 번까지 재촬영을 해 서버에 재요청을 한다.

안드로이드 카메라를 실행했을 때 안드로이드 카메라는 default 로 90 도 회전이 되어있는 상태로 카메라가 켜진다. 안드로이드는 자체로 회전 값을 가지고 있어 이미지를 저장할 때 해당 회전을 처리한다고 한다. 그래서 유니티로 코드를 짜서 안드로이드로 실행을 해보면 사진이 회전이 되어있어 face api 에 그 사진 그대로 전송을 하면 얼굴 인식이 되지 않는 에러가 발생했다. 그래서 해당 에러를 고치기 위해 실시간으로 카메라 화면을 rotation 시키는 건 overhead 가 너무 크기 때문에 먼저 사진을 찍고 해당 이미지를

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

rotation 해주는 코드를 추가했다.


#### (5) 음성 녹음

음성 녹음은 처음에는 누르고 있으면 음성이 녹음이 되고 떼면 녹음이 멈추고 파일이 저장되는 형식으로 구현을 했었다. 하지만 그렇게 누르고 있는게 불편하다는 말이 있어 toggle 형식으로 변경하였다. 최소 3 초 이상 녹음을 해야하고 3 초 이하일 때는 다시 녹음을 시도해달라고 안내가 뜬다. 음성을 녹음 하는 방법은 내가 버튼을 눌렀을 때부터 시간을 쟀다. 그리고 다시 또 한번 눌렀을 때의 현재 시간을 가져와서 해당 값을 뺀 만큼의 음성파일을 저장해준다. 해당 녹음을 .wav 파일로 변환해 저장을 하고 서버에 byte 배열로 전송을 한다. 녹음 또한 인공지능 모델에 들어가 감정 정보가 분석이 되지 않을 경우를 고려하여 제대로 된 값을 받지 못했다는 response 가 올 경우 3 번까지 다시 재요청을 한다.

#### (6) 데이터 통신

서버와는 HTTP 통신을 한다. 서버에 값을 보낼 때는 Post method 를 이용하고 서버에 값을 요구할 때는 Get method 를 사용한다. post 에는 보낼 정보인 byte 배열 값이랑 서버 url 에 command 를 추가로 붙여서 원하는 요청에 맞게 보낸다. Post 를 보내고 나면 ACK 나 NAK 를 받을 때까지 yield return 으로 기다린다. ACK 를 받으면 내가 보냈던 request 의 downloadhandler 로 서버가 보낸 값이 들어오는데 해당 값을 받아서 처리를 한다. NAK 를 받으면 재요청을 하거나 해당 response type 에 따라서 exception 처리를 한다.

ACK 를 받았을 때 data form 이 string 이라면 string 변수에 값을 저장을 하고 지정해 놓은 delimiter character 에 따라서 string 값을 split 해주고 string 배열에 순서대로 저장을 해준다. 받는 값이 json 형식인 경우를 처리하기 위해, c#에서는 Newtonsoft.Json 이 필요하다. json 처리를 위해 해당 파일을

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

다운로드를 해놓았다. 그래서 들어온 값이 json 형식이라면 jtoken 으로 parsing 을 하는데 서버에서 보낸 json 이 encoding 이 한 번 더 거쳐서 오기 때문에 backslash 가 추가되는 오류가 있었다. 해당 오류를 잡기 위해 파싱을 하기 전 받은 값에 replace 를 거쳐서 Jarray 라면 jarray 로 parse 를 진행하고 jobject 라면 jobject 로 parse 를 진행한다. 저장한 jtoken 은 요청한 response 에 따라서 json convert 로 deserialize 를 한다. deserialize 를 하는 object 의 형식을 미리 class 로 정의를 해놨다. 서버에서 json 이 dictionary 형태이기 때문에 서버에서 보내는 key 값과 class 의 변수의 이름을 맞춰서 정의해 놓은 class 에 맞춰서 deserialize 를 진행한다.

```

{
  "music_1": "헬로카봇 엔지니어 크루송",
  "link_1": "https://youtu.be/7abTBSDdN4Y",
  "music_2": "시크릿 쥬쥬 - 시크릿 플라워 '봄봄봄'",
  "link_2": "https://youtu.be/zHD5gN9q-8s",
  "music_3": "카드캡터체리 오프닝송",
  "link_3": "https://youtu.be/QsGZFva3eQ8"
}
UnityEngine.Debug.Log(Object)
<HandleFileRequest>d__11:MoveNext() (at Asse

```


헬로카봇 엔지니어 크루송

시크릿 쥬쥬 - 시크릿 플라워 '봄봄봄'

카드캡터체리 오프닝송

## (7) 감정 분석 결과 표시

제일 마지막 state 에 도달하면 서버에 result 로 Get 요청을 해서 분석 결과 json 값을 받으면 해당 값을 정리해서 팝업으로 보여준다. 노래 추천에 대한 정보는 총 세 개를 받게 구현했다. 버튼이 세 개가 존재해서 각각의 노래 정보 별로 버튼의 text 에 '노래 제목 - 가수'를 표시해주고 가져오려는 유튜브의 썸네일이 존재할 시에는 버튼 위에 썸네일 image 도 같이 표시를 해준다. 노래 제목이 적혀있는 해당 버튼을 클릭했을 시에는 application.openURL 을 사용하여 내 어플에 존재하는 유튜브 어플을 실행시켜서 정보로 받은 url 을 실행시켜준다. 감정 결과가 끝나면 재시작 버튼을 통해 state 변수를 초기화해 처음부터 재실행이 가능하다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

#### (8) 동영상 재생

동영상 재생에는 많은 이슈가 존재 했다. 원래의 목표는 노래를 클릭했을 때 동영상 사이즈의 창이 떠서 해당 창에 유튜브 영상이 뜨게 하는 것이었다. 이렇게 동영상 재생을 하기 위해서는 .mp4 처럼 영상 확장자로 끝나는 url 을 이용해서 재생을 할 수 있는데 유튜브는 그저 공유를 위한 url 이 때문에 그냥 url 로는 원하는 대로 영상 재생이 불가능 했다. 그래서 youtube api 도 나 youtube android player api 를 다운 받아 시도해보았다. 하지만 youtube api 를 처음 사용하기 위한 방법이 인터넷이나 youtube api 사이트에 적혀있지 않아서 애를 먹었다. 그러다 하이퍼링크라는 차선택을 선택하게 됐다. openURL 로 유튜브 영상을 띄우는 것에 성공해서 동영상 재생은 application.openURL 을 사용해서 하이퍼링크로 노래를 재생해주기로 했다.

#### (9) 마야 모델 파일 유니티에 임포트하는 방법

마야에서는 모델에 다양한 rendering 을 걸 수가 있다. 그런데 제작한 모델을 export 할 때 마야에서는 마야 파일 확장자 말고 다른 확장자로 export 를 하면 해당 랜더링 값을 같이 포함해서 뽑아주지 않는다고 한다. 기본 텍스처 값만 export 를 해줘서 다른 것과 설정한 값들과 같이 export 를 하려하니 export error 가 계속 났다. 유니티에 import 하기 위한 .fbx 확장자로 export 할 때 다른 텍스처 사용의 제공 자체가 안되는 문제가 발생했다. 그래서 다른 render 값을 제외한 기본 texture 만 가지고 model 을 export 를 해서 유니티에 import 시키니 오류없이 모델이 import 되었다.

#### (10) permission

unity 에는 기본으로 자동으로 permission 을 요청하는 기능이 있다. 그래서 권한이 필요한 기능을 사용할 때마다 permission 이 요청되는데 어플을

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

실행함에 있어 뭔가를 하려할 때마다 멈춰서 permission 이 요청되면 불편할 것이다. 해당 문제를 해결하기 위해서 unity 를 android 로 build 를 하면 생기는 androidmanifest.xml 에 자동으로 permission 이 불리는 기능을 해제 해놓았다. 그리고 unity 2018.3 버전부터는 유니티에서 자체적으로 permission 을 요청할 수 있는 클래스가 추가 되었다고 한다. 해당 클래스를 이용해서 permission 을 요청하는 코드를 구현하고 script 가 불리는 순서를 가장 먼저로 수정해서 먼저 불릴 수 있도록 했다.

#### (11) exception 처리


서버에 요청을 했는데, 서버를 찾을 수 없을 때, 돌아온 에러 값이 null 이거나 empty 면 client error 로 예외처리를 해준다.

status code 가 2xx 면 success 이므로 요청했던 request 에 맞는 response delegate 에 들어있는 함수를 불러 실행시킨다.

서버를 통해 받은 값을 파싱하지 못할 때는 parse error 로 예외처리가 된다.

response 가 successful 이 아닌데 status code 가 404 면 page not found 로 예외처리를 해준다.

사진이나 음성은 실패해도 3 번 다시 서버에 재요청을 해본다. 그래도 실패한다면 서버의 상태가 좋지 못한 것으로 간주하고 대화창에 해당 메시지를 적어준다.


 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

## 2.2.2 시스템 기능 요구사항

대분류	소분류	기능	형식	완료여부
Server	Microsoft Face API	이미지를 이용한 감정 분석	Class	완료
	Speech-Emotion-Analyzer	음성을 이용한 감정 분석	Class	변경
	Database	사용자 정보, 코멘트, 추천할 노래, 이에 따른 태그를 저장하는 공간	Module	완료
	Recommendation Algorithm	분석된 감정 결과에 추천되는 노래를 매칭하는 알고리즘	Module	완료
Application	3D Model Animation	어플리케이션 진행 및 디스플레이해줄 모델	Model	완료
	Analyzed Result	서버에 데이터를 보내고, 받은 정보를 디스플레이	Class	완료

### - 변경사항


피드백을 받은 결과 데이터셋 약 2000 개를 가지고 직접 학습을 하였을 때 얻게되는 정확률이 부족할 것으로 예상된다. 예상되는 정확률이 약 3-40%정도 되어 최종 감정 결과를 도출할 때에 이미지 인식 결과를 무디게할 수 있다고 생각하여 직접 학습시킨 모델을 빼고, API 를 통한 이미지 인식과 Speech\_Emotion\_Anlyzer 의 활용성을 더욱 높일 수 있는 방향으로 한다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

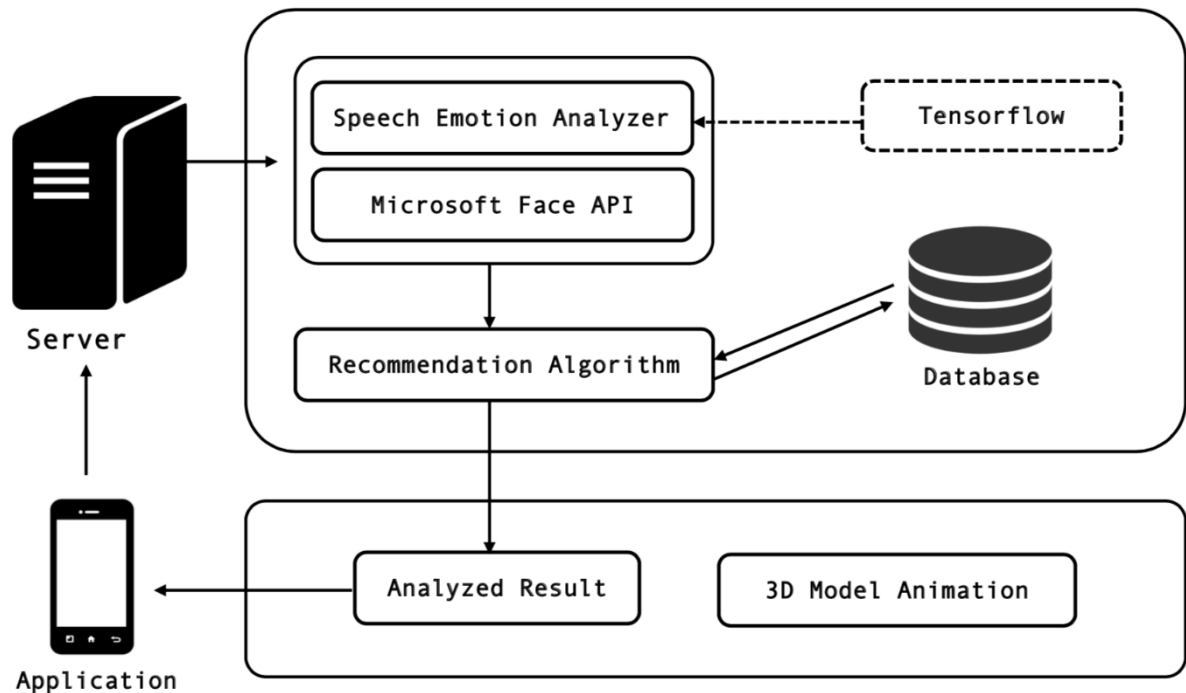
### 2.2.3 시스템 비기능(품질) 요구사항

분류	요구사항	달성여부
Performance	어플리케이션은 사용자의 감정에 따른 적절한 노래를 추천 한다	달성
	어플리케이션에서 서버로 이미지와 음성이 빠르고 적절히 전달된다	달성
	어플리케이션은 100 번중 70 번이상의 비율로 막힘없이 진행된다	달성
Availability	사용자는 별도의 기술을 익히지 않고 어플리케이션을 사용할 수 있다	달성
	안드로이드 사용자라면 누구나 다운받아 사용할 수 있다	달성
Competitiveness	감정을 분석하는 다른 어플리케이션과는 다르게 음성-감정 인식 기능과 노래추천 기능이 더해진다	달성



 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

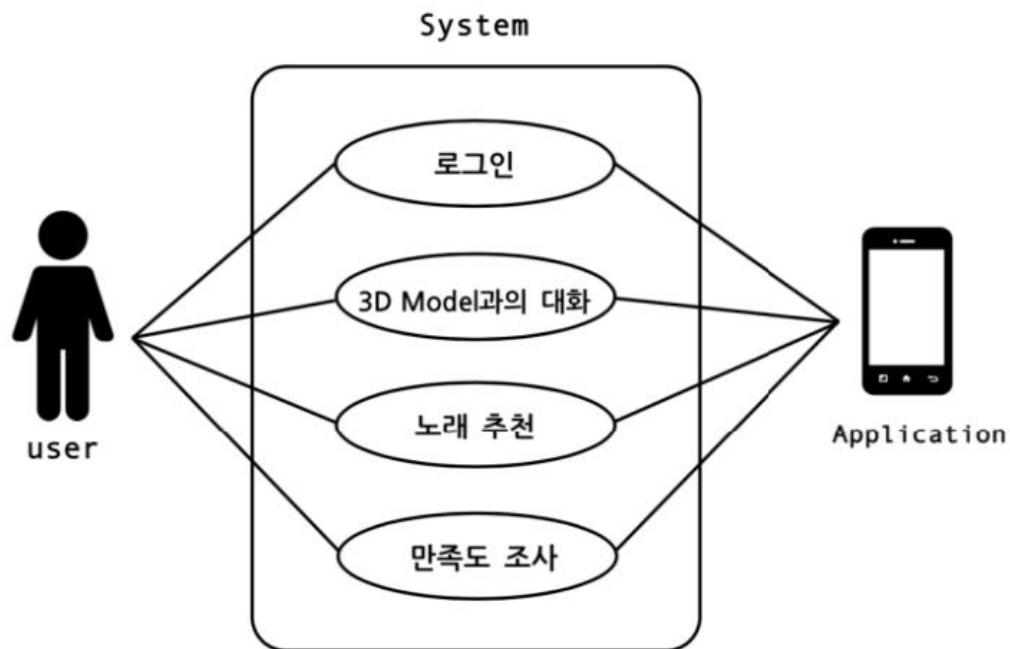
## 2.2.4 시스템 구조 및 설계도



어플리케이션은 사용자에게서 입력받은 이미지와 음성 데이터를 서버에 넘겨준다. 데이터를 입력받은 서버는 Microsoft Face API 를 사용하여 이미지를 분석하고 Speech Emotion Analyzer 를 통해 음성을 분석한다. 분석된 이미지와 음성의 결과를 조합한 뒤, Recommendation Algorithm 을 통해 DB 내의 알맞은 노래와 매칭한다. 선정된 노래는 Application 의 3D Model 의 animation 을 통해 알려준다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

## Usecase Diagram



- 사용자는 어플을 실행시켜 회원가입을 원할 시 로그인 기능을 제공 받을 수 있다.
- 사용자는 화면을 터치해서 3D 모델을 활성화 시킬 수 있어야 한다.
- 사용자는 어플리케이션의 3D 모델과 대화를 주고 받을 수 있다.
- 사용자는 두가지 데이터를 분석한 감정결과를 어플 화면을 통해 볼 수 있다.
- 사용자는 어플을 통해 추천된 노래의 태그와 유튜브를 볼 수 있다.
- 사용자는 자신의 추천 받은 노래에 대한 만족도를 선택할 수 있다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28


## 2.2.5 활용/개발된 기술

### 1) Back-End(server)

- Django, MySQL, AWS framework 사용
- Microsoft Face API
- 딥러닝을 활용한 Speech-Emotion-Analyzer
- 감정 분석결과를 통해 노래를 매칭하는 Recommendation Algorithm
- HTTP 를 활용한 I/O

### 2) Front-End

- Unity, Maya framework 사용
- HTTP 를 활용한 I/O

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

## 2.2.6 현실적 제한 요소 및 그 해결 방안

### 1) 음성-감정 모델을 학습시키기 위한 데이터셋이 부족한 경우

IEMOCAP 과 RAVNESS 에서 얻을 수 있는 데이터셋의 양은 약 2000 와 12 시간의 분량이다. IEMOCAP 같은 경우에는 script 에대한 감정 Attribute 정보가 txt 로 제공되어 정확한 타이밍에 의도한 감정을 캐치하기 힘든 점과 RAVNESS 에서 제공한 2000 개의 데이터를 더해도 훈련데이터셋과 테스트데이터셋으로 분류해서 사용하기 매우 부족함이 있어, github 에 올라온 정확률이 검증된 모델을 사용하여 어플리케이션의 완성도를 높일 수 있도록 한다. 한국어 데이터셋으로 훈련한 결과는 감정이 섞인 한국어 음성데이터셋이 충분히 확보될 경우 파악할 수 있도록 한다.


### 2) Face recognition 과 Speech-Emotion 의 결과가 상충되는 경우

사용자의 표정과 음성을 인식한 결과가 반대되는 경우가 나올 수 있다. 이때에 이미지 인식결과로 나온 3 개의 감정결과와 음성인식 결과를 비교하여, 공통된 감정이 없으면 예외처리를 통해 '거짓말' 카테고리의 노래를 추천한다. 모순될 수 있는 상황을 어플리케이션의 재미요소로 풀어나간다.


 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

### 2.2.7 결과물 목록

대분류	소분류	기능	기술문서 유/무
Server	Microsoft Face API	이미지 분석을 통한 감정 분석	유 <a href="https://azure.microsoft.com/ko-kr/services/cognitive-services/face/">https://azure.microsoft.com/ko-kr/services/cognitive-services/face/</a>
	Speech-Emotion-Analyzer	음성 분석을 통한 감정 분석	유 <a href="https://github.com/MITESHPU/THRANNEU/Speech-Emotion-Analyzer">https://github.com/MITESHPU/THRANNEU/Speech-Emotion-Analyzer</a>
	Database	사용자 정보, 코멘트, 추천할 노래를 저장하는 공간	유 2.1.1 연구개발내용 참조

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

	Recommendation Algorithm	분석된 감정 결과에 추천되는 노래를 매칭하는 알고리즘	유  2.1.1 연구개발내용 참조
Application	3D Model Animation	어플리케이션에 디스플레이해줄 모델	유  2.1.1 연구개발내용 참조
	Analyzed Result	서버에 데이터를 보내고, 받은 정보를 디스플레이	유  2.1.1 연구개발내용 참조

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

## 2.3 기대효과 및 활용방안

본 프로젝트를 통해 다음과 같은 긍정적인 효과가 발생할 것을 기대할 수 있다.


### 1) 감정 분석 기술에 대한 기술적 향상과 상업적 요소

감성을 정보기술로 인식하는 ‘감성 컴퓨팅’ 분야에 대한 연구는 아직 일상에서 자연스럽게 사용할 정도로 상용화하지는 못하였다. 하지만 기계학습 등 AI 고도화 과정을 단축할 수 있는 기술들이 나오면서 최근 빠르게 발전하고 있다. 이에 따라 모바일, 헬스케어, 신 ICT 산업 분야 등 다양한 산업 분야에 감정 분석 기술이 이용되고 있다. 우리 프로젝트를 통해, 감정 분석 시스템을 구현하고 활용함으로써 감정 분석 기술에 발 맞추어 나갈 수 있을 것이다.

나아가 본 어플리케이션의 노래 추천에 랭킹 알고리즘을 도입하면 상업적으로 발전 시키는 것도 가능할 것이고 기존 대중적인 음악 어플리케이션과 제휴를 맺으면 완곡 스트리밍 서비스 제공 또한 기대할 수 있다.

### 2) 사용자 관점에서의 자기 인식 고취와 재미 요소


AI 스피커가 유행하는 이유는 혼자 사는 사람들 또는 바쁜 현대인들이 간단한 대화를 할 수 있는 재미를 느낄 수 있기 때문이다. ‘말하는 고양이 토미’도 휴대폰 화면에서 고양이와 대화할 수 있다는 기능 만으로도 선풍적인 인기를 끌었다. 마찬가지로 My Mood Music도 화면상에 등장하는 베이맥스 3D 모델과 간단한 대화를 진행할 수 있고, 기존 어플의 이미지를 인식하여 포토샵 적용, 심지어는 이미지를 인식하여 감정을 보여주는 어플리케이션과는 다르게 감정을 인지하지 못한 채 살아가는 현대인들의 감정을 이미지 뿐 아니라 음성을 통해서도 분석해주는 것에서 그치지 않고 일상생활에서 친숙하게 접할 수 있는 ‘노래’라는 소재를 접목한다. 이때에 사용자의 연령이 파악되어 노래 추천에 향수를 불러 일으킬 수 있는 요소와 이미지와 감정이 맞지 않았을 때의 상황 등 재미요소까지 더해 어플을 이용하는 사용자는 지루할 틈이 없게 된다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

### 3 자기평가

이름/역할	자기 평가 내용
오윤정/ 서버개발 및 이미지인식	<p>초반에 계획했던 AWS Elastic Beanstalk 와 Django 를 연동하는 목표를 이루지 못했다. 하지만 AWS 의 RDS 인스턴스를 활용하여 연동하는 것은 마쳤지만, 시연할 때 따로 컴퓨터에 서버를 돌리기로 하여서 소용이 없게 돼서 아쉬웠다. 하지만 서버를 구축하고 데이터베이스를 연동하고 Class view 및 URLconf 설계 등의 목표를 모두 이루었다.</p> <p>어플과의 Http 통신 또한 구현하였다. 그리고 여러 함수들을 병합하여 하나의 flow 를 만들어낸 것에 굉장히 만족한다. 초반에 가상환경을 세팅하고, Django 프로젝트를 생성하고, 테이블을 구현하고 MySQL 과 연동해보는 과정에서 시행착오가 무척 많았다. 그 와중에 팀원들의 세팅을 돕기 위해 우분투 환경에서 윈도우 환경으로, 또 MacOS 환경으로 세팅을 다시 하는 등의 힘든 과정을 겪었다. 그래도 그런 시행착오를 겪으면서 에러에 많이 주눅들지 않는 자세를 가지게 된 것 같다.</p>
박정은/어플리케이션 개발 및 3D 모델	<p>전체적인 3D 모델 애니메이션 제작과 어플리케이션 개발을 완성했다는 것에 한 단계 성장한 느낌이다. 어플리케이션 개발에 생각보다 더 많은 시간이 투자되어서 좀 더 퀄리티 높은 3D 관련 제작을 하지 못한 것이 아쉬움으로 남는다. 어플의 서버 통신을 위한 backend 와 사용자에게 정보를 display 해주는 frontend 를 다 개발하고 delegate 와 coroutine, singleton, observer pattern 과 같은 디자인 패턴 기능들을 이해하고</p>




 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

	<p>사용한 것에 상당히 만족스럽다. 처음으로 어플의 전체적인 개발을 해본 거라 정말 많은 시행착오가 필요했지만 오류의 detection 실력이 늘기도 했고 완성이라는 목표를 이루어 뿌듯하다. 이 어플을 좀 더 다듬어서 완성도 있는 어플로 만들고 싶은 바람이 있다.</p>
원수빈/database	<p>초반의 어플리케이션 시스템을 확립하는 과정에서는 데이터베이스 설계를 계속 수정해야하는 시행착오가 있었다. 하지만 시스템이 확립된 이후부터는 데이터베이스의 큰 틀을 잡아놓고, 세부 디테일만 수정하면 되었다. 노래를 추천해주는 어플리케이션의 특성상 노래 데이터가 많이 쌓여 있을수록 사용자는 더 만족하는 결과를 얻게 된다. 따라서 노래 데이터를 최대한 많이 쌓으려고 노력했다. 이번 프로젝트에서는 설문과 태그를 통해 자체적으로 정한 정책에 의해 노래를 분류하였지만, 노래 분류에 대한 기계적인 알고리즘을 설계한다면 더 확장성 있는 어플리케이션이 될 것으로 판단된다.</p>
홍자현/추천 알고리즘 설계 및 구현	<p>우리 시스템만의 추천 알고리즘 정책을 만들어나가는 과정에서, 결국 노래 추천 결과가 알고리즘의 성능을 입증하는 것이라고 판단했다. 그래서 사소한 요소라도 놓치지 않기 위해 팀원들과 많이 소통하고 데이터 수집을 위해 설문을 돌리는 등, 더 나은 결과물에 대한 욕심을 냈다. c++로 알고리즘을 작성해 속도를 최대한 보장하고 싶었지만 리눅스 서버로 옮기는 문제때문에 python 으로 다시 작성했다는 점이 아쉽다. 추후 사용자 만족도를 조사해 추천 알고리즘의 성능을 보강하면 더 나은 서비스를 제공할 수 있을 것이라고 판단된다.</p>
강은채/ 음성 데이터 모델	<p>딥러닝을 활용해 직접 학습을 시킬 수 있는 기회를 놓쳐 아쉽지만 어플리케이션이 잘 진행되기 위해서 최선의 선택을 한 것이라 생각한다.</p>

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

	<p>그리고 github 에 올라온 모델을 로컬에서 직접 수집한 나의 데이터를 가지고 결과를 확인하며 성능을 평가하고, 어플리케이션에서 결과를 확인할 수 있도록 시도했던 점이 유익했다. 기회가 된다면 한국어 데이터를 수집하여 이 모델의 정확률을 한국 음성데이터에서도 높일 수 있도록 하는 것을 목표로 하고있다.</p>
--	---

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

## 4 참고 문헌

번호	종류	제목	출처	발행년도	저자
1	서적	텐서플로 첫걸음		2016	박해선
2	사이트	Django REST framework	<a href="https://www.django-rest-framework.org/">https://www.django-rest-framework.org/</a>		
3			<a href="http://dreamusn.kr/Home/board/bbs/board.php?bo_table=IT_News&amp;wr_id=18332&amp;page=82">http://dreamusn.kr/Home/board/bbs/board.php?bo_table=IT_News&amp;wr_id=18332&amp;page=82</a>		
4			<a href="http://keunwoochoi.blogspot.com/2016/03/2.html">http://keunwoochoi.blogspot.com/2016/03/2.html</a>		
5			<a href="https://engineering.linecorp.com/ko/blog/voice-waveform-arbitrary-signal-to-noise-ratio-python/">https://engineering.linecorp.com/ko/blog/voice-waveform-arbitrary-signal-to-noise-ratio-python/</a>		

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>				
	<b>프로젝트 명</b>		My Mood Music#마무무		
	<b>팀 명</b>		포춘텔러		
	Confidential Restricted		Version 1.2	2019-MAY-28	

6		국내 우울증 환자 60 만명 넘었다.	<a href="http://www.hani.co.kr/arti/PRINT/789572.html">http://www.hani.co.kr/arti/PRINT/789572.html</a>		
7		Speech-Emotion-Analyzer	<a href="https://github.com/MI-TESHPUTHRANNEU/Speech-Emotion-Analyzer">https://github.com/MI-TESHPUTHRANNEU/Speech-Emotion-Analyzer</a>		
8		IEMOCAP	<a href="https://sail.usc.edu/iemocap/iemocap_release.htm">https://sail.usc.edu/iemocap/iemocap_release.htm</a>		
9		wav dataset	<a href="https://tspace.library.utoronto.ca/handle/1807/24492">https://tspace.library.utoronto.ca/handle/1807/24492</a>		
10		유튜브 썸네일	<a href="https://yeahvely.tistory.com/30">https://yeahvely.tistory.com/30</a>		
11	기사	김성훈 - 딥러닝 강좌 (유튜브)	<a href="https://youtu.be/BS6O0zOGX4E">https://youtu.be/BS6O0zOGX4E</a>	2017-04-06 13:39	김양중
12	포스팅	자기인식 능력의 비밀, 감성지능	<a href="https://brunch.co.kr/@hesse24/46">https://brunch.co.kr/@hesse24/46</a>	2016-08-10	김권수

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

13	논문	감성 ICT 기술 및 산업동향	<a href="https://ettrends.etri.re.kr/ettrends/149/0905001979/29-5_30-39.pdf">https://ettrends.etri.re.kr/ettrends/149/0905001979/29-5_30-39.pdf</a>	2014-10-01	신현순, 함찬영, 엄남경, 김미경, 이석희, 김용선
----	----	------------------	---	------------	------------------------------------

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

## 5 부록

### 5.1 사용자 매뉴얼

- 1) 어플리케이션 시작한다
- 2) 팝업으로 어플리케이션의 간단한 소개와 사용법을 읽고 닫는다
- 3) 자고있는 3D 모델 베이맥스를 더블 터치하여 깨운다
- 4) 베이맥스의 간단한 인사를 듣고 눈을 맞추면 대화로 이어간다
- 5) 녹음버튼을 누른 뒤 베이맥스의 물음에 3 초이상 대답을 한뒤 다시 버튼을 누른다
- 6) 결과 창이 뜨면 태그와 추천된 노래의 썸네일을 확인하고 링크를 눌러 유튜브로 이동한다
- 7) 재시작 버튼을 누르면 감정 분석 재실행이 가능하다

### 5.2 운영자 매뉴얼

- 1) Microsoft Face API 를 사용해 ID/KEY 발급받는다
- 2) python3.5(64bit) 이상의 버전에서 파이썬 모듈 설치(tensorflow, keras, librosa)

pip 을 사용하여 Django 프로젝트를 수행할 가상환경을 세팅해놓아야 한다

github 에 있는 2019-cap1-2019/src/pyDjango/mmm\_project 로 들어가서 requirements.txt 를 다운받아 <pip3.6 install -r requirements.txt> 명령어를 사용하여 세팅을 완료한다

Windows 환경이라면 MySQL 을 미리 설치하여야 한다. 연동하는 것은 <https://jamanbbo.tistory.com/28> 이 링크를 참조한다

start project 를 하고 나서 settings.py 를 고쳐야 한다. 한국이라면 아래와 같다



```
# Internationalization
# https://docs.djangoproject.com/en/2.1/topics/i18n/

LANGUAGE_CODE = 'ko-kr'

TIME_ZONE = 'Asia/Seoul'

USE_I18N = True

USE_L10N = True

USE_TZ = True

MEDIA_URL = '/media/'

MEDIA_ROOT = os.path.join(BASE_DIR, 'media')

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/2.1/howto/static-files/

STATIC_URL = '/static/'

STATICFILES_DIRS = [os.path.join(BASE_DIR, 'static')] # 추가
```


그 이후 Django 를 이용해 프로젝트를 만드는 방법은 다음 링크를 참고한다

[https://tutorial.djangogirls.org/ko/django\\_start\\_project/](https://tutorial.djangogirls.org/ko/django_start_project/)

<https://github.com/MITESHPUTHRANNEU/Speech-Emotion-Analyzer>

에서 Emotion\_Voice\_Detection\_Model.h5, model.json 을 다운받아 실행파일 디렉토리에 둔다

- 3) github 의 unity 폴더에 있는 폴더들을 clone 한다. unity 에 새 project 를 생성하고 해당 파일들을 붙여 넣고 실행을 시킨다. scene 에 있는 start 를 실행한다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

### 5.3 테스트 케이스

대분류	소분류	기능	테스트방법	기대 결과	결과
Back-End	server	GET 기능	어플에서 또는 웹에서 GET 메소드를 요청했을 때 테스트한다.	request.GET request.FILES request.headers 출력	성공
		POST 기능	어플에서 또는 웹에서 POST 메소드를 요청했을 때 데이터를 출력한다.	request.POST request.FILES request.headers 출력	성공
	Microsoft Face API	감정분석 정확률	어플에서 받은 사진을 분석하는 것뿐만 아니라 구글에서 사람 표정에 관련된 데이터셋을 모은 후에 API 를 사용하여 결과를 본다.	사람이 파악한 감정과 API 가 분석한 감정이 비슷하기를 기대	성공
		얼굴인식	어두울 때, 얼굴의 측면사진일 때 잘 되는지 확인. 초점이 안 맞고 흐릿한 사진일 때 잘 되는지 확인. 가까울 때나 멀 때나 잘 되는지 확인.	마스크나 가면을 쓰고 있는 경우에 마스크를 코 밑까지는 내려도 얼굴이 인식되기를 기대	성공





			얼굴의 반이 잘렸을 경우, 또는 가면을 쓰고 있을 경우에 얼굴 인식이 잘 되는지 확인.		실패
Speech- Emotion- Analyzer	음성 분석을 통한 감정 분석		유튜브에서 드라마나 영화 클립을 활용하여 영어뿐 아니라 한국어 음성 데이터를 넣었을 때 정확률이 어떻게 나오는지 확인, 그 결과가 영어인식결과와 비슷한 확률로 나오기를 기대	한국어의 angry 감정을 fearful 로 일관되게 분석하는 것 말고는 대부분 정확률이 낮게 보임	실패
		음성 파일 저장	어플리케이션에서 byte array 로 보내준 audio 파일을 file.wav 로 저장하여 모델에 wav 파일 자체를 바로 넣을 수 있도록 한다. 이때, 같은 디렉토리에 이전에 녹음한 file.wav 가 존재하면 그것을 지운다.		성공
	Database	사용자 정보, 추천할 노래를 저장하는 공간	감정 별 노래를 csv 파일에 리스트화 해놓고, Django server 에 python 코드를 통해 한번에 올린 뒤, 테이블에 맞는 노래가 올라갔는지 확인한다.	감정 별 테이블에 적절한 노래가 올라갔다.	성공

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	My Mood Music#마무무	
	<b>팀 명</b>	포춘텔러	
	Confidential Restricted	Version 1.2	2019-MAY-28

	Recommendation Algorithm	분석된 감정 결과에 추천되는 노래를 매칭하는 알고리즘	얼굴을 통한 감정 분석 결과와 음성(억양)을 통한 감정 분석 결과를 알고리즘의 입력으로 넣었을 때 나오는 결과인 노래 테이블 목록과 설계 때 예상한 결과를 비교한다.	추천 알고리즘의 결과와 설계에 의해 예상한 결과가 동일한 모습을 기대할 수 있다.	성공
Application	3D Model	어플리케이션에 디스플레이 해줄 모델	어플리케이션을 실행시켜본다. 3D Model 이 잘 나오는지 확인		성공
	Analyzed Result	서버에 데이터를 보내고, 받은 정보를 디스플레이	어플리케이션에서 사진을 찍고 음성을 녹음한 후 결과가 나오기까지를 기다린 후 정보가 잘 나오는지 확인	노래 세가지가 결과 팝업에 출력 해당 노래를 클릭시 하이퍼링크로 유튜브로 연결	성공