**캡스톤 디자인 I**

**종합설계 프로젝트**

|  |  |
| --- | --- |
| **프로젝트 명** | *FeelingU, Chillin’ with me, ChillinChillin* |
| **팀 명** | *Fortune Teller* |
| **문서 제목** | 계획서 |

|  |  |
| --- | --- |
| **Version** | 1.0 |
| **Date** | 2019-03-08 |

|  |  |
| --- | --- |
| **팀원** | 오윤정 (조장) |
| 박정은 |
| 원수빈 |
| 홍자현 |
| 강은채 |

|  |
| --- |
| **CONFIDENTIALITY/SECURITY WARNING**  이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인I 수강 학생 중 프로젝트 “xxxx xxxx”를 수행하는 팀 “Fortune Teller”의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 “Fortune Teller”의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다. |

**문서 정보 / 수정 내역**

|  |  |
| --- | --- |
| **Filename** | 계획서-프로젝트명.doc |
| **원안작성자** | 오윤정, 박정은, 원수빈, 홍자현, 강은채 |
| **수정작업자** | 원수빈, 홍자현, 오윤정 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 수정날짜 | 대표수정자 | Revision | 추가/수정 항목 | 내 용 |
| 2019-03-08 | 전원 | 1.0 | 최초 작성 | 공란으로 되어있는 내용작성 및 수정 |
| 2019-03-10 | 오윤정 | 1.0 | 내용 수정 | 공란으로되어있는 내용작성 및 수정 |
| 2019-03-10 | 홍자현 | 1.0 | 내용 수정 | 공란으로되어있는 내용작성 및 수정 |
| 2019-03-11 | 전원 | 1.0 | 내용 수정 | 연구 개발 및 주제에 대한 내용 수정 |
| 2019-03-14 | 전원 | 1.0 | 내용 수정 및 추가 | 연구 개발 및 주제에 대한 내용 수정 및 추가 |
|  |  |  |  |  |
|  |  |  |  |  |

**목 차**

[**1**](#_gjdgxs) **개요** 4

[1.1 프로젝트 개요 4](#_30j0zll)

[1.2 추진 배경 및 필요성 4](#_1fob9te)

[**2**](#_3znysh7) **개발 목표 및 내용** 5

[2.1 목표 5](#_2et92p0)

[2.2 연구/개발 내용 6](#_tyjcwt)

[2.3 개발 결과 7](#_3dy6vkm)

[2.3.1 결과물 목록 및 상세 사양 7](#_1t3h5sf)

[2.3.2 시스템 기능 및 구조 7](#_1t3h5sf)

[2.4 기대효과 및 활용방안 7](#_2s8eyo1)

[**3**](#_17dp8vu) **배경 기술** 8

[3.1 기술적 요구사항 8](#_3rdcrjn)

[3.2 현실적 제한 요소 및 그 해결 방안 9](#_26in1rg)

[3.2.1 하드웨어 9](#_lnxbz9)

[3.2.2 소프트웨어 9](#_35nkun2)

[3.2.3 기타 9](#_1ksv4uv)

[**4**](#_44sinio) **프로젝트 팀 구성 및 역할 분담** 10

[**5**](#_qsh70q) **프로젝트 비용** 10

[**6**](#_3as4poj) **개발 일정 및 자원 관리** 11

[6.1 개발 일정 11](#_1pxezwc)

[6.2 일정별 주요 산출물 12](#_49x2ik5)

[6.3 인력자원 투입계획 13](#_2p2csry)

[6.4 비 인적자원 투입계획 14](#_46r0co2)

[**7**](#_2lwamvv) **참고 문헌** 15

# **개요**

## 프로젝트 개요

본 프로젝트에서는 사용자의 얼굴(표정)과 음성(억양)을 인식하여 감정을 분석한 후 분석한 정보를 간단한 코멘트와 함께 보여주고, 이에 적절한 노래를 추천해주는 어플리케이션을 개발할 것이다.

이 때, 사용자가 어플리케이션을 실행하면 카메라 또는 갤러리로 얼굴 이미지를 api로 입력해 표정에 관한 감정정보와 추정 나이 정보를 얻는다. 어플리케이션 안의 캐릭터가 질문을 하면 사용자가 직접 대답하고 사용자의 음성을 인식하여 학습한 모델로 억양에 관한 감정정보를 분석해낸다.

이렇게 나온 3가지 정보(표정에 관한 감정정보(70%), 억양에 관한 감정정보(30%), 추정 나이 정보)를 조합하여, 최종적인 결과를 도출해내고 어플리케이션 안의 캐릭터가 감정분석 결과와 함께 적절한 노래를 추천해준다.

# 

## 추진 배경 및 시장 동향 분석

**1.2.1 추진 배경 및 필요성**

감정이란 어떤 현상이나 일에 대하여 일어나는 마음이나 느끼는 기분으로, 감정을 파악하기 위해서는 감정을 정교하게 표현하는 언어적 수단인 감정단어가 있다. 감정단어의 효과적인 의미 전달을 위해 표정, 억양, 강세, 빠르기, 등의 준언어적 요소가 말소리에 반영되어야 표현의 강도를 극대화할 수 있음을 강조하였다(Owens, 2005; Van Rheenen & Russell, 2013). 이런 특성을 가진 감정은, 개인의 사회적 상호작용과 결정을 내리기 위한 판단력 뿐만 아니라 생존과 성장에 영향을 주므로 자신의 감정을 이해하는게 중요하다. 우리 시스템은 그것을 돕기위해 표정과 음성을 통해 감정을 분석하고, 더 나아가 그 결과를 활용해 사용자에게 유용한 음악을 추천해준다.

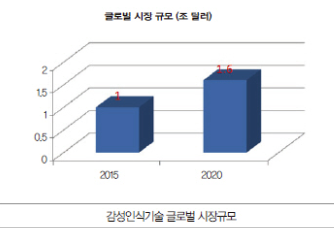
감성 인식 기술이란 사람이 느끼는 감성을 얼굴 표정이나 목소리, 생체 리듬 등을 통해 인지하는 기술이다. IT 기술이 발달하면서 인간의 감성을 읽기 위해 여러 센서부터 인공지능 알고리즘까지 접근 방법이 다양해지고 있다. 이를 ‘감성 컴퓨팅'이라고 하며, 인공지능을 활용하면 이미지 속의 인물을 식별해 분노, 슬픔, 놀라움, 공포 등 여러 감정 요인을 확률적으로 표현해서 보여준다.

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.

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





# **개발 목표 및 내용**

## 목표

본 프로젝트는 사용자에게 감정 분석 결과를 알려주고 이를 바탕으로 노래를 추천해주는 어플리케이션 제작을 목표로 한다. 감정 분석 결과는 얼굴인식을 통한 감정분석, 음성인식을 통한 억양분석, 그리고 앞선 얼굴인식을 통해 알아낸 추정나이를 종합해 도출해낸다.

어플리케이션 구성도는 다음과 같다.

**(어플리케이션 구성도 넣기)**

## 연구/개발 내용

**본 프로젝트의 수행의 내용을 구체적으로 기술한다.**

**목표를 세분화하여 세부 목표를 정하고 그에 따른 결과물을 제시한다.**

**연구/개발 방법을 기술한다. 연구/개발 방법은 단계별 수행 방법을 기술한다.**

핵심적인 개발 내용은 다음과 같다.

1. 사용자의 얼굴 이미지를 API에 입력해 감정과 나이를 분석한다.
2. 음성인식을 통한 억양 파악의 경우에는 API가 아닌 확보된 data set을 통해 직접 학습시키고, 학습된 데이터를 감정 분석에 이용한다.
3. 분석한 감정에 맞는 노래를 추천해주는 알고리즘을 구현한다.
4. 3D 캐릭터는 분석된 감정에 따라 적절한 애니메이션을 보여준다.
5. api와 통신하고, 최종결과를 도출하고, 노래를 추천해주는 역할은 API server에서 담당한다.

\*다 쓰고 밑에거 지우기\*

(전체적인 방향 ->2.1의 구성도로 대체)

1. 스플래시 이미지 (Splash Image)
   1. 어플을 시작하면 로고가 나오고 사라진다.
2. 실행이 안 된 모델이 나타나있다.
3. 더블클릭을 하면 모델이 활성화된다. 어플에 대한 설명을 한다.
4. 얼굴 인식을 위해 카메라를 띄울 건지, 갤러리의 이미지를 사용할 건지 선택하게 한다.
   1. 카메라를 선택했을 경우, 카메라가 떠서 사진을 찍어 api 서버에 전달한다.
   2. 갤러리를 선택했을 경우, 본인이 원하는 사진 하나를 결정하면 api 서버에 전달한다.
5. 음성인식을 위한 질문을 몇 가지 한다.
   1. 오늘 기분이 어때? 오늘 너의 하루는 어땠니?
   2. 음성 데이터가 너무 짧다고 판단될 경우, 재요청을 한다.
6. api서버는 학습된 모델의 결과와 api의 결과를 종합해 최종 결과를 만든다.
7. 최종결과를 추천 알고리즘을 사용해 추천해줄 노래를 결정한다.
   1. 노래 추천할 때는 최종 감정 결과와 사용자의 나이 정보를 이용한다.
8. 어플은 코멘트와 함께 노래 정보와 감정 분석된 결과를 받아 사용자에게 보여준다.

얼굴인식을 통한 감정 분석을 위해 필요한 정보는 이미지 속 얼굴의 추정 나이와 얼굴에서 분석된 감정이다. 이 조건에 부합하는 api인 Microsoft Face API을 선택했다. Face API는 입력된 비전 데이터를 통해 얼굴을 인식하거나 얼굴 감지를 이용한 애플리케이션을 만들 때 유용한 API다. Google Cloud Vision와 Clova Face Recognition은 얼굴 이미지에 분석되는 감정이 한 가지 value로 정의 되지만 Microsoft Face API는 다양한 감정에 대한 정도를 분석해주어 좀 더 정밀한 결과를 도출할 수 있을 것이다.

**2.2.1 얼굴 인식 단계**

어플을 처음 실행하게 되면 캐릭터가 나타난다. 캐릭터는 더블클릭으로 활성화되고, 어플의 사용흐름을 안내한다. 첫 번째 단계인 얼굴 인식 단계에서, 우선 입력 이미지를 얻기 위해서 카메라 혹은 갤러리를 사용한다. 카메라를 사용할 경우 사용자는 자연스러운 상태로 얼굴을 찍게 되고, 갤러리를 사용할 경우 사용자가 원하는 이미지를 선택하게 된다. 이렇게 결정된 얼굴 이미지는 API 서버로 보내지고, 다시 API 서버에서 Face API에 입력되어, 얼굴에서 분석된 감정 데이터를 API로부터 얻는다. 이 데이터는, MySQL DB의 ~를 PK로 한 사용자 정보 테이블 저장이 된다.

\* (기능 추가안할때)사용자 정보 테이블에 저장할 정보인 CFR에서 오는 ‘감정, 나이’를 api server를 거쳐야하나?

\* + (사용자 정보 테이블가 의미 있기 위해 거기에 사용자가 선택한 음악을 넣을 경우 2.2.3.2 에 적을 말 -> 단, 로그인 기능이 추가됨 )사용자가 어플리케이션을 사용할 때마다 생성되는 데이터들이 DB에 저장되면 (1) 사용자는 분석된 감정, 추천받은 순위별 음악 리스트, 그리고 자신이 선택한 음악에 대한 누적 기록을 어플리케이션에서 리스트로 한눈에 확인할 수 있다. (2)그리고 사용자가 선택한 음악은 DB의 사용자 정보 테이블에 저장이 되어 추천 알고리즘에 적용할 수 있다. (예를 들어, 좀 더 밝은 곡의 3순위 음악을 선택해서 기분이 우울할 때 꼭 우울한 음악을 듣는건 아니다 -> 사용자에게 맞는 음악을 추천할 수 있다)

**2.2.2 음성 인식 단계**

얼굴인식이 끝난 후 캐릭터는 사용자에게 질문을 한다. 사용자는 대답하고 대답음성이 인식 되면 미리 학습해둔 모델에서 감정분석 결과를 얻게 된다. 이 결과는 얼굴 인식 결과와 70대 30의 비율로 감정 분석 최종결과에 기여한다.

직접 학습 모델을 만들기 위해서 data를 수집한다. IEMOCAP에서 감정분석을 위한 음성 dataset를 사용하여 학습시킨다. IEMOCAP에서 제공받은 data는 그 길이가 다양하다. 총 12시간 분량에 해당하는 data로 기존의 2명의 배우가 연기한 음성data에서 24명의 배우가 더 정교한 목소리연기와 텍스트를 제공한다.

학습 모델은 tensorflow로 구현하게 된다. 입력받은 data의 패턴을 분석하여 감정의 종류별로 분류한다. 음성신호 ->전처리 ->특징값 추출(MFCC) ->분류기 ->결과 순서로 이루어 진다. data의 전처리 과정은 매우 중요하다. 일반적으로 사용되는 전처리 방법에는 Normalization, Whitening, PCA-Principle Component Analysis 등이 있다. 신호가 가지고 있는 정보를 최대한 보존하면서 전체 data의 크기를 최소화 해야 효율적으로 연산이 가능하다.

특징값의 대표적인 예로는 MFCC가 있다. 음색이 다르다는건 주파수 에너지 비율이 다르기 때문이고 이것을 배음 구조가 다르다 라고 할 수 있다. MFCC는 바로 이 배음구조의 차이를 표현하는 숫자다. MFCC의 특징이 음정이 변해도 일정하게 유지된다는 점이라서 톤이 높은사람이 안녕하세요 했을 때와, 톤이 낮은사람이 안녕하세요 한것은 같게 인식이 되어야 하고 음성인식, 감정 인식 등 다양한 분야에서 MFCC가 활용된다. 감정 분석을 할 때 우리가 사용할 특징값은 무엇이 될 것인가? 생각해 보아야 한다.

분류기는 y=f(x)라 했을 때 y는 결과 f는 분류기라고 할 수 있다. 0이면 기쁨, 1이면 슬픔이라 설정해 두었을 때, 그 둘중 하나를 고르는걸로 설정해 놓고 우리는 학습을 시키게 된다.

이렇게 학습된 모델은 api서버상에 올려지게 된다.

**2.2.3 분석 결과 및 노래 추천 단계**

**2.2.3.1감정 분석 결과 열람 단계**

**2.2.3.2 노래 추천 및 선택 단계**

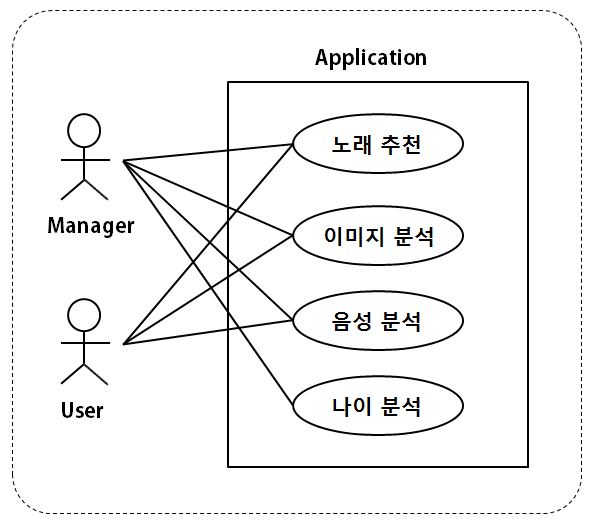
얼굴인식 단계에서 사용자의 얼굴 이미지로부터 추정나이를 얻을 수 있다. 이 정보와 앞선 두 단계의 정보를 api server로 보낸다.

전체적인 어플리케이션 개발

애니메이션이 주를 이루는 어플리케이션이기 때문에 개발하면서도 바로바로 결과를 볼 수 있도록 unity로 개발한다. 시작화면은 스플래시 이미지로 한다. 이미지가 사라지고 나면 아직 깨어나지 않은 캐릭터가 화면 상에 나타나있을 것이다. 배경은 포켓몬고처럼 카메라 배경 혹은 집 안 같은 3d 모델로 한다. 잠들어있는 캐릭터는 더블클릭을 통해 깨울 수 있다. 캐릭터가 깨어나면 캐릭터는 인사하는 애니메이션과 함께 어플에 대한 소개와 흐름을 이끌어가주는 역할을 한다. 어플 소개 후에는 얼굴 인식을 하기 위해서 카메라로 이미지를 찍을 것인지 혹은 내 갤러리에 있는 사진을 사용할 것인지 선택할 수 있다. 갤러리 접근은 안드로이드로 해야하기 때문에 유니티 에셋스토어에 존재하는 native android asset을 사용한다. 이렇게 이미지를 정하고나면, 캐릭터가 사용자에게 몇가지 질문을 건넨다. 예를 들어, “오늘 기분은 어땠어?”, “오늘의 하루는 어땠어?” 가 될 수 있다. 만약 들어온 음성 정보의 데이터가 너무 작다면 더 자세하게 이야기를 해줄 수 있냐는 식의 대화를 통해 더 나은 음성을 얻어 음성 인식을 할 수 있게 할 것이다. 그렇게 일련의 정보 수집 단계가 끝나고, 만약 정보를 처리함에 있어서 시간이 어느정도 소요된다면 캐릭터가 지금 분석을 하는 중이라는 모션을 취하게 한다. 그게 아니라면 팝업으로 감정 분석에 대한 결과를 표시해주고, 해당 결과에 대한 작은 코멘트를 해준다. 예를 들어, 우울이 크게 나왔다면 “너의 감정 분석 결과에 우울이 큰 비중을 차지하더라. 오늘 하루도 힘내!”라는 가 될 수 있다. 그리고 나서 감정 분석 정보에 맞춘 노래를 추천해줄 것이다. 노래 또한 팝업으로 제공하고, 유튜브 영상을 띄운다. 유튜브 영상을 띄우는 것 또한 유니티에 에셋으로 존재하기 떄문에 에셋을 활용한다.

## 개발 결과

### 시스템 기능 요구사항

****

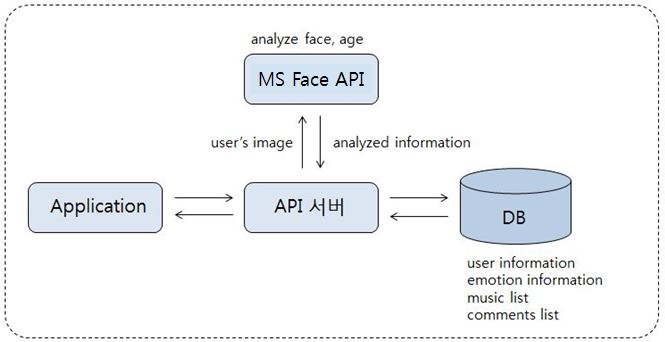
* 어플리케이션

1. 핸드폰 내의 카메라로 사용자의 얼굴을 인식한다.
2. 어플리케이션은 입력된 사용자의 얼굴을 분석하고 그에 맞는 정보(감정, 연령)를 데이타베이스에 업데이트한다.
3. 어플리케이션은 질문에 대한 사용자의 대답 음성을 분석하고 그에 맞는 정보(감정)를 데이타베이스에 업데이트한다.
4. 서버에서 업데이트 된 정보에 데이터베이스 내의 적절한 노래를 매칭한다.
5. 어플리케이션의 화면에 감정분석 결과와 선별된 노래의 유투브를 띄워준다.

### 시스템 비기능(품질) 요구사항

1. 사용자가 어플을 사용하기 시작하면 10초 내에 이미지 분석이 되어야 한다.
2. 어플리케이션을 100번 수행하였을 때 95번은 막힘없이 결과가 나와야 한다.
3. 선별된 노래는 정식으로 등록된 유튜브 영상을 제공해야 한다.
4. 데이터베이스는 관리자만 접근이 가능해야 한다.

### 시스템 구조



어플리케이션은 사용자에게서 입력받은 얼굴 이미지를 API서버에 넘겨주고 API서버는 Face API에 사용자의 이미지를 넘겨준다. Face API에서는 이미지를 통해 감정과 나이를 분석하고 API서버에 넘겨준다. 이 결과를 API서버에서 분석한 음성 정보와 조합하여 DB내의 노래와 매칭한다. DB내에는 사용자의 정보 테이블(나이, 감정), 감정 정보 테이블, 음악 리스트 테이블, 코멘트 리스트 테이블, 관계 테이블이 존재한다.

### 결과물 목록 및 상세 사양

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **대분류** | **소분류** | **기능** | **형식** | **비고** |
| 어플리케이션 | *얼굴 분석* | 얼굴을 인식하고 분석한다 | API |  |
|  | *음성 인식* | 음성을 인식하고 분석한다 | 학습 |  |
|  | *노래 추천* | .분석 결과에따라 노래를 추천한다 | 추천 알고리즘 |  |
|  | *코멘트* | 분석 결과에따라 캐릭터가 사용자에게 코멘트한다. | 매칭 알고리즘 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## 기대효과 및 활용방안

활용방안 = 감성기술을 전망?활용방안? + 우리 어플은 가벼운(쓰진말고) 요소도 추가함(아래내용)

2.4.1 감성 기술 관점 (기대효과)

2.4.2 사용자 관점 (기대효과)

심적인 공허함과 병을 알지 못하고, 자신의 감정이 어떤지 제대로 돌아보지도 못한 채로 살아가고 있는 현대인들로 하여금 자신의 감정을 되돌아보고 스스로도 모르고 있었던 자신의 감정을 이해할 수 있게 한다. 자신의 감정을 실시간으로 파악하고 이를 토대로 노래를 추천하는 우리의 소프트웨어는 바쁘게 삶을 달려나가는 현대인들에게 조금이나마 흥미를 선사할 휴식처가 될 수 있을 것이다.

기존에 존재하는 감정에 따른 노래 추천은 단순히 사람들의 기분을 몇 개로 분류해놓고, 그에 맞는 노래를 정리해놓았을 뿐이다. 하지만 사람의 감정이란 것은 한 가지로만 정의되기에는 부족하다. 한 순간에도 여러 가지의 감정을 느끼는 경우가 많기 때문이다. 이렇듯 자신이 느끼는 복합적인 순간의 감정을 분석해보고 이를 토대로 노래를 추천 해주는 프로그램은 사용자들에게 기존의 것과는 다른 새로움과 다양함을 느끼게 해 줄 것이다.

대중적으로 많이 사용되는 기존 음악어플에서 제공하는 단순한 상황별, 장르별 추천 노래는 타인에의해 선정되거나 단순한 분류의 데이터 분석을통해 제공된다. 따라서 사용자가 새로운 노래를 추천 받고자 할 때, 기존 플랫폼의 단순한 노래 추천 서비스는 사용자들의 흥미를 끌기 어렵다.

본 프로젝트에서 개발할 어플은 사용자의 감정을 좀 더 세부적으로 판단한다. 입력된 이미지와 음성데이터를 함께 고려하여 사용자의 현재 감정을 분석하고, 이를 토대로 노래를 추천해주는 것이다. 더 나아가 얼굴 인식을 통해 사용자의 나이를 분석할 것이다. 이는 단순히 최신 가요만을 추천하는 것이 아닌 사용자의 연령대에 맞는 노래를 추천해주어 여러 연령대의 사용자들이 즐길 수 있게 해준다. 따라서 본 프로젝트에서는 이 어플을 통해 사용자들에게 현재 자신의 감정을 파악해보고 이에 적절한 노래를 추천받아 현대인들의 바쁜 삶 속에서 소소한 재미를 느끼고 힐링할 수 있는 경험을 선사하는 것을 기대해 볼 수 있다.

어플리케이션의 구현이 완성되면 Google play store에 출시하여 여러 사용자가 사용할 수 있게 할 것이다. 더나아가 본 어플리케이션의 노래 추천에 랭킹 알고리즘을 도입하면 상업적으로 발전 시키는 것도 가능할 것이고 기존 대중적인 음악 어플리케이션과 제휴를 맺으면 완곡 스트리밍 서비스 제공 또한 가능해 질 것이다.

# **배경 기술**

## 기술적 요구사항

1. 개발하는 데 필요한 개발 환경

* 운영체제 환경 : Windows10
* 컴파일 환경: 터미널, Anaconda prompt
* 개발 언어 : Java, C#, python, My SQL
* 언어의 문법적 요구사항 :

2. 프로젝트 결과물을 확인할 수 있는 환경

* 동작시킬 수 있는 운영체제 환경 : Windows10, Android
* 미리 설치되어 있어야 하는 소프트웨어 및 라이브러리 : Nox
* 서버 환경 : AWS

## 현실적 제한 요소 및 그 해결 방안

**프로젝트를 수행하기 이전에 시스템 개발시 발생할 가능성이 있는 제한 요소를 미리 예측하여 나열한다. 또한 그 제한 요소를 피해갈 수 있는 해결 방안에 대해서도 나열한다. 예를 들어, GNU 라이선스가 있는 소프트웨어 라이브러리를 사용하는 경우에 이를 사용하는 소프트웨어의 소스를 공개하여야 한다. 만약 개발할 시스템이 상용화 제품일 경우에는 문제가 발생할 수 있다. 이를 어떻게 해결할 것인가? 하는 점 등이다. 또한 시스템의 성능(속도, 처리할 수 있는 데이터의 양 등등)이 어느 정도 이상이 되어야 한다든지 혹은 안정성을 어느 정도 확보를 하여야 하는 점도 현실적 제한 요소가 될 수 있다. 이를 하드웨어 측면 혹은 소프트웨어적인 측면에 대하여 기술한다.**

**이러한 현실적 제한요소를 팀원들과 토의한 내용과 지도 교수님과 토의한 내용은 반드시 회의록에 남기도록 한다.**

### 하드웨어

* 어플 용량이 커서 공기계에서 잘 돌아가지 않을 경우

-> 어플의 용량을 최소화한다.

* 안드로이드 스튜디오 에뮬레이터가 잘 돌아가지 않을 경우

-> cpu intel i5 6세대 이상, RAM 8GB이상 되는 사양의 노트북 구비한다.

-> 안드로이드 스튜디오 기본사양에 대한 문서를 참고한다.

### 소프트웨어

* API의 성능이 떨어질 경우

-> 대체 API를 마련해놓고, 각 API의 성능을 측정하여 더 좋은 API를 사용한다.

* Unity와 Maya의 튕김 현상

-> 백업파일을 중간중간에 계속 생성한다.

* 쌓아둔 추천노래들의 data의 양이 굉장히 방대해졌을 경우

-> 카테고리를 세분화하여 저장한다.

* 추천 노래의 저작권 문제 (스트리밍을 할 경우 저작권에 문제가 있다.)

-> 가급적 소속사에서 공식으로 올린 youtube를 연결한다.

### 기타

* 어플리케이션 개발 과정에서 특허를 등록할 경우 오픈소스 라이브러리를 사용했다면 소스를 공개해야 한다.

# **프로젝트 팀 구성 및 역할 분담**

|  |  |
| --- | --- |
| **이름** | **역할** |
| 오윤정 | Software Project Leader  API 서버 구축 및 관리  API를 이용한 감정 분석과 음성 정보를 통합한 결과 도출 |
| 박정은 | Maya 모델링 및 애니메이션, 어플 개발 |
| 원수빈 | 데이터 관리 |
| 홍자현 | 추천 알고리즘 정의 |
| 강은채 | 음성 인식을 통한 감정분석 모델 생성 및 학습 |

# **프로젝트 비용**

**프로젝트에 투입될 비용을 예상하여 기입한다. 여기서 Man-Days 라 함은 인적 비용을 계산하기 위한 단위로 한 사람이 하루 8시간 정도 일을 해야 하는 양을 1 MD라고 한다. 즉, 한 사람이 하루 4시간씩 일을 하게 되면 이틀 정도 일을 했을 때 1 MD 정도의 비용이 들어갔다고 한다.**

|  |  |
| --- | --- |
| **항목** | **예상치 (MD)** |
| API서버 구축 | 30 |
| API서버 관리 | 40 |
| 어플리케이션 개발 | 40 |
| 데이터 관리 | 30 |
| Maya 모델링 및 애니메이션 | 60 |
| 음성인식 모델 학습 | 70 |
| 추천 알고리즘 구현 | 40 |
|  |  |
| 합 | 310 |

# **개발 일정 및 자원 관리**

## 개발 일정

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **항목** | **세부내용** | **3월** | | **4월** | | **5월** | | **비고** |
| 요구사항분석 | 요구 분석 |  | |  | |  | |  |
| SRS 작성 |  | |  | |  | |  |
| 관련분야연구 | 주요 기술 연구 |  | |  | |  | |  |
| 관련 시스템 분석 |  | |  | |  | |  |
| 설계 | 시스템 설계 |  | |  | |  | |  |
| 구현 | 코딩 및 모듈 테스트 |  | |  | |  | |  |
| 테스트 | 시스템 테스트 |  | |  | |  | |  |

## 일정별 주요 산출물

|  |  |  |  |
| --- | --- | --- | --- |
| **마일스톤** | **개요** | **시작일** | **종료일** |
| 계획서 발표 | 개발 환경 완성 (GCC 설치, 기본 응용 작성 및 테스트 완료)  **산출물 :**   1. 프로젝트 수행 계획서 2. 프로젝트 기능 일람표 | 2019-03-08 | 2019-03-15 |
| 설계 완료 | 시스템 설계 완료  **산출물 :**   1. 시스템 설계 사양서 | 2019-03-15 | 2019-03-31 |
| 중간 보고 | API서버, DB구축, 어플리케이션 UI, 모델링 구현 완료  **산출물 :**   1. 프로젝트 1차 중간 보고서 2. 프로젝트 진도 점검표 3. 1차분 구현 소스 코드 | 2019-04-01 | 2019-04-19 |
| 구현 완료 | 시스템 구현 완료  **산출물: 어플리케이션** | 2019-04-01 | 2019-04-30 |
| 테스트 | 시스템 통합 테스트  **산출물: 어플리케이션 베타버전** | 2019-05-01 | 2019-05-18 |
| 최종 보고서 | 최종 보고  **산출물: 프로젝트 최종 보고서, 포스터 및 책자** | 2019-05-20 | 2019-05-31 |

## 인력자원 투입계획

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **이름** | **개발항목** | **시작일** | **종료일** | **총개발일(MD)** |
| 오윤정 | *API서버 구축, API 사용한 감정분석* | 2019-03-08 | 2019-05-31 | 60 |
| 박정은 | *3d 캐릭터 애니메이션 구현, 어플 개발* | 2019-03-08 | 2019-05-31 | 60 |
| 원수빈 | *데이터 관리* | 2019-03-08 | 2019-05-31 | 60 |
| 홍자현 | *추천 알고리즘 개발* | 2019-03-08 | 2019-05-31 | 60 |
| 강은채 | *음성인식 학습모델 개발* | 2019-03-08 | 2019-05-31 | 60 |

## 비 인적자원 투입계획

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **항목** | **Provider** | **시작일** | **종료일** | **Required Options** |
| 컴파일러 | Microsoft, Oracle | 2019-03-08 | 2019-05-31 |  |
| 개발용 PC 5대 | Dell,Lenovo, Mac | 2019-03-08 | 2019-05-31 |  |
| 안드로이드 공기계 | 미정 | 2019-03-08 | 2019-05-31 |  |
| gpu서버 |  | 2019-03-18 | 2019-04-30 |  |
| aws서버 | Amazon | 2019-03-18 | 2019-05-31 |  |

# **참고 문헌**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **번호** | **종류** | **제목** | **출처** | **발행년도** | **저자** | **기타** |
| 1 | 서적 | 텐서플로 첫걸음 |  | 2016 | 박해선 |  |
| 2 | 사이트 | django REST framework | <https://www.django-rest-framework.org/> |  |  |  |
|  |  |  | <http://dreamusn.kr/Home/board/bbs/board.php?bo_table=IT_News&wr_id=18332&page=82> |  |  |  |