



국민대학교
전자정보통신대학
컴퓨터공학부


캡스톤 디자인 I

종합설계 프로젝트

| | |
|--------|----------------|
| 프로젝트 명 | Music Seat |
| 팀 명 | 14wh |
| 문서 제목 | MUSIC SEAT 계획서 |

| | |
|---------|------------|
| Version | 1.0 |
| Date | 2018-03-19 |

| | |
|----|-----------|
| 팀원 | 임 현재 (조장) |
| | 성 정훈 |
| | 조 용현 |
| | 변 승현 |
| | 위 봉우 |

| | | | |
|---|-------------------------|-------------|-------------|
|  국민대학교 컴퓨터공학부 캡스톤 디자인 I | 계획서 | | |
| | 프로젝트 명 | | |
| | 팀 명 | | |
| | Confidential Restricted | Version 1.2 | 20xx-MAR-25 |


CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인I 수강 학생 중 프로젝트 **"Music Seat"**를 수행하는 팀 **"14조"**의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 **"14조"**의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

문서 정보 / 수정 내역


| | |
|-----------------|--------------------|
| Filename | 계획서-Music Seat.doc |
| 원안작성자 | 임현재, 성정훈, 조용현, 변승현 |
| 수정작성자 | 임현재, 조용현, 성정훈, 변승현 |

| 수정날짜 | 대표수정자 | Revision | 추가/수정 항목 | 내 용 |
|----------|-------|----------|----------|---------------|
| 2018-03- | 임현재 | 1.0 | 최초 작성 | |
| 2018-03- | 임현재 | 1.1 | 내용 수정 | 일정 및 역할분담 재조정 |
| 2018-03- | 임현재 | 1.2 | 내용 수정 | 현실적 제한요소 추가 |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

| | | | |
|---|-------------------------|-------------|-------------|
|  국민대학교 컴퓨터공학부 캡스톤 디자인 I | 계획서 | | |
| | 프로젝트 명 | | |
| | 팀 명 | | |
| | Confidential Restricted | Version 1.2 | 20xx-MAR-25 |

목 차

| | | |
|----------|--------------------------|-----------|
| 1 | 개요 | 4 |
| 1.1 | 프로젝트 개요 | 4 |
| 1.2 | 추진 배경 및 필요성 | 4 |
| 2 | 개발 목표 및 내용 | 5 |
| 2.1 | 목표 | 5 |
| 2.2 | 연구/개발 내용 | 5 |
| 2.3 | 개발 결과 | 6 |
| 2.3.1 | 결과물 목록 및 상세 사양 | 6 |
| 2.3.2 | 시스템 기능 및 구조 | 6 |
| 2.4 | 기대효과 및 활용방안 | 8 |
| 3 | 배경 기술 | 8 |
| 3.1 | 기술적 요구사항 | 8 |
| 3.2 | 현실적 제한 요소 및 그 해결 방안 | 8 |
| 3.2.1 | 하드웨어 | 8 |
| 3.2.2 | 소프트웨어 | 8 |
| 3.2.3 | 기타 | 9 |
| 4 | 프로젝트 팀 구성 및 역할 분담 | 9 |
| 5 | 프로젝트 비용 | 9 |
| 6 | 개발 일정 및 자원 관리 | 10 |
| 6.1 | 개발 일정 | 10 |
| 6.2 | 일정별 주요 산출물 | 10 |
| 6.3 | 인력자원 투입계획 | 11 |
| 6.4 | 비 인적자원 투입계획 | 11 |
| 7 | 참고 문헌 | 12 |

| | | | |
|--|-------------------------|-------------|-------------|
|  국민대학교 컴퓨터공학부 캡스톤 디자인 I | 계획서 | | |
| | 프로젝트 명 | | |
| | 팀 명 | | |
| | Confidential Restricted | Version 1.2 | 20xx-MAR-25 |

1 개요

YouTube의 URL이나 동영상, MP3파일 에서 음원을 따와서 음원을 푸리에 트랜스 폼을 이
용해 악보를 추출해내는 것.

1.1 프로젝트 개요

1. 유튜브에서의 음원 다운로드
2. 음원에서의 음정분리
3. 음정으로의 악보 구성
4. 악보의 플레이
5. 악보의 수정

1.2 추진 배경 및 필요성

이 프로젝트는 악기로 음악을 연주하고 싶을 때 기존의 클래식 악보 외에는 악보가 없는
상황이 항상 발생하여 이를 타개하고자 프로젝트를 추진하게 되었다.


기존의 악기를 연주하는 사람들이라면 새롭게 연주를 하고싶은 곡을 찾는 곳 은 대개 유
튜브나 음원파일, 또는 동영상 파일을 보고 그러한 마음을 가지게 된다. 이러한 상황에서
해당 음원의 악보를 구하는 것은 하늘의 별 따기와 같다. 대부분 음원이 MP3파일이나 동영
상 형태로만 존재하지 악보로는 존재하지 않는다. 또한 악보를 찾기 위해 많은 공을 들여
야한다.

이러해도 악보를 찾지 못하였다면 두가지 방법이 있다. 청음과 시장의 프로그램 사용이다.
첫번째가 청음을 하여 한음 씩 음을 청음 하여 악보에 적 는 것이다. 이는 매우 전문적인
지식과 경험을 필요로 한다. 하지만 이러한 경험을 가지고 있는 사람일지라도 화음이 겹치
게 되면 실제로 하기 어려운 일이 된다.

다른 두번째는 시중에 나와있는 AudioScore Ultimate, 코디 파이 등 악보추출 프로그램을 이
용하는 것이다.

먼저 예로 AudioScore Ultimate 이 프로그램은 기존에 악보를 추출하는 프로그램으로 며
\$249.95에 판매되는 프로그램이다. 이는 단지 취미를 위해 또는 여흥거리로 악보를 구하려
는 사람에게는 쉽지않은 지출이다.

또한 다른 프로그램들인 코디 파이는 URL을 이용할 수 있으나 일정이상을 이용하려면 유
료로 결제를 해야 하는 결제방식을 가지고있다. 실제로 코디 파이에서 실험을 해본결과를
예로 들자면 유명 드라마인 별에서 온 그대의 OST인 My Destiny의 부분을 넣어보았다 추
출된 코드는 Am, D, G, C, B, E 하지만 악보의 음은 Am, D, G, F#m, B7, EM7이며 3분의 1
을 틀린 것을 확인 할 수 있다. 다른 무료 프로그램들도 이와 유사하며 심지어 더 음정이나
코드의 정확성이 떨어지는 것을 확인 할 수 있었다.

| | | | |
|---|-------------------------|-------------|-------------|
|  국민대학교 컴퓨터공학부 캡스톤 디자인 I | 계획서 | | |
| | 프로젝트 명 | | |
| | 팀 명 | | |
| | Confidential Restricted | Version 1.2 | 20xx-MAR-25 |

이러한 상황에서 프로젝트인 Music Seat는 음원을 푸리에 트랜스폼을 통해 변환함으로써 기존의 제품들 보다 향상된 악보의 정확성을 끌어올리는 것이 주목적이며, 비 직관적이던 제품들을 좀더 직관적으로 만들어 제품의 직관성을 높일 것이다. 이러한 부분만 고쳐도 상당한 경쟁력을 가 질것으로 판단된다.

이러한 프로그램들의 시장동향은 시장진입하기에 쉬울 것으로 판단된다. 고급의 기능을 가진 프로그램들은 AudioScore Ultimate와 같이 대략 \$200 ~ \$250 사이에 거래되고 있으며 기능이 부족하거나 악보의 정확성이 떨어지는 제품들은 사용자에게 낮은 평점을 받아 중간에 공백이 생겨 이러한 중간층을 노리는 전략을 세워 두고있다. 즉 가격도 소비자가 부담되지 않으며 가격대비 성능과 정확성도 만족을 시키는 것이 핵심이라 할 수 있다.

2 개발 목표 및 내용

2.1 목표

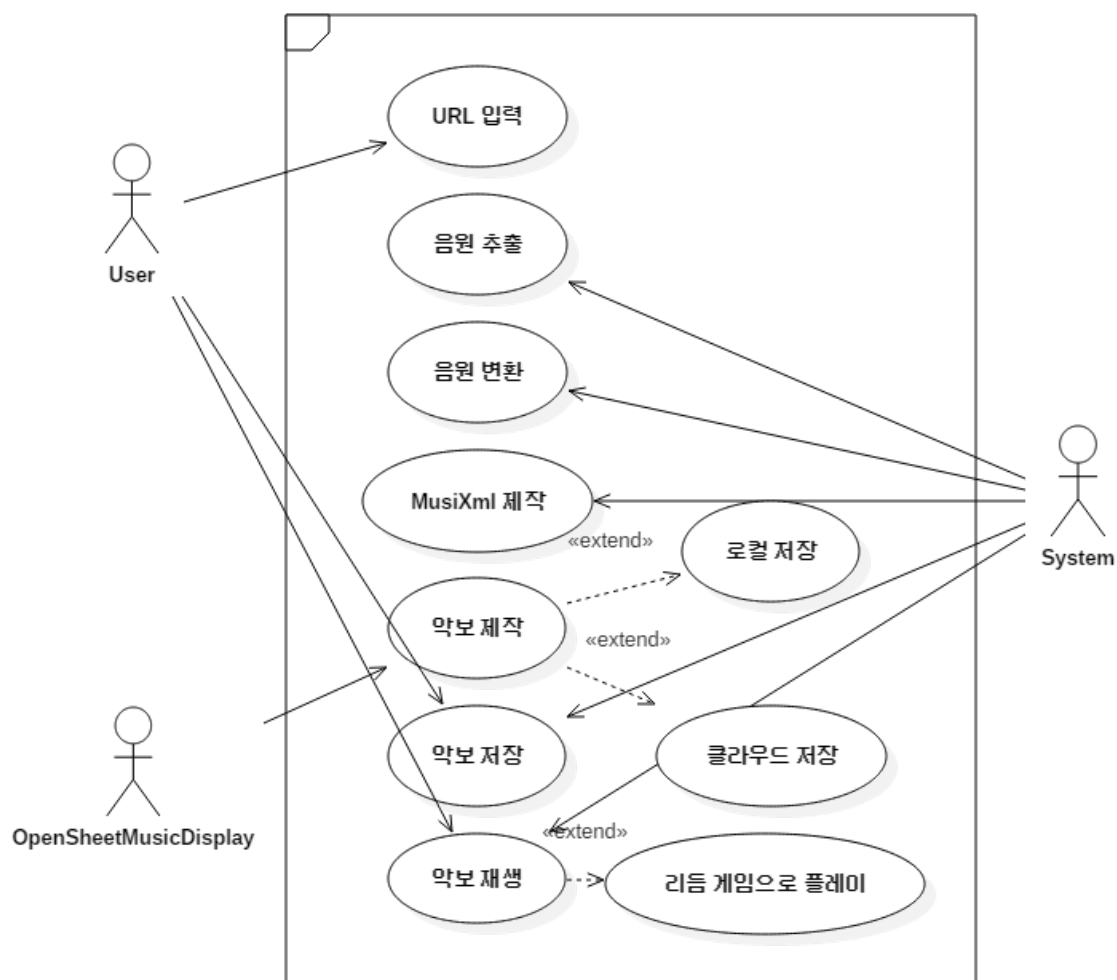
동영상이나 유튜브의 URL에서 주소를 따와 음원에서 악보를 추출한다, 음원 파일에서도 악보를 추출한다.

2.2 연구/개발 내용

1. 유튜브 url에서 mp4 동영상을 추출한뒤 wav 음성 파일으로 변환
2. wav 파일을 읽어서 초당 44k 샘플링 음성 데이터를 4.4k 씩 분할
3. 4.4k씩 분할된 음성 데이터를 각각 fourier transform을 진행
4. fourier transform 결과에서 amplitude 와 frequency를 추출
5. amplitude를 이용하여 유의미한 frequency만 추출
6. frequency를 음역으로 변환 한뒤 music xml 생성
7. xslt를 제작하여 xml을 html 형태로 변경
8. 클라이언트에서 html을 출력

2.3 개발 결과

2.3.1 시스템 기능 요구사항



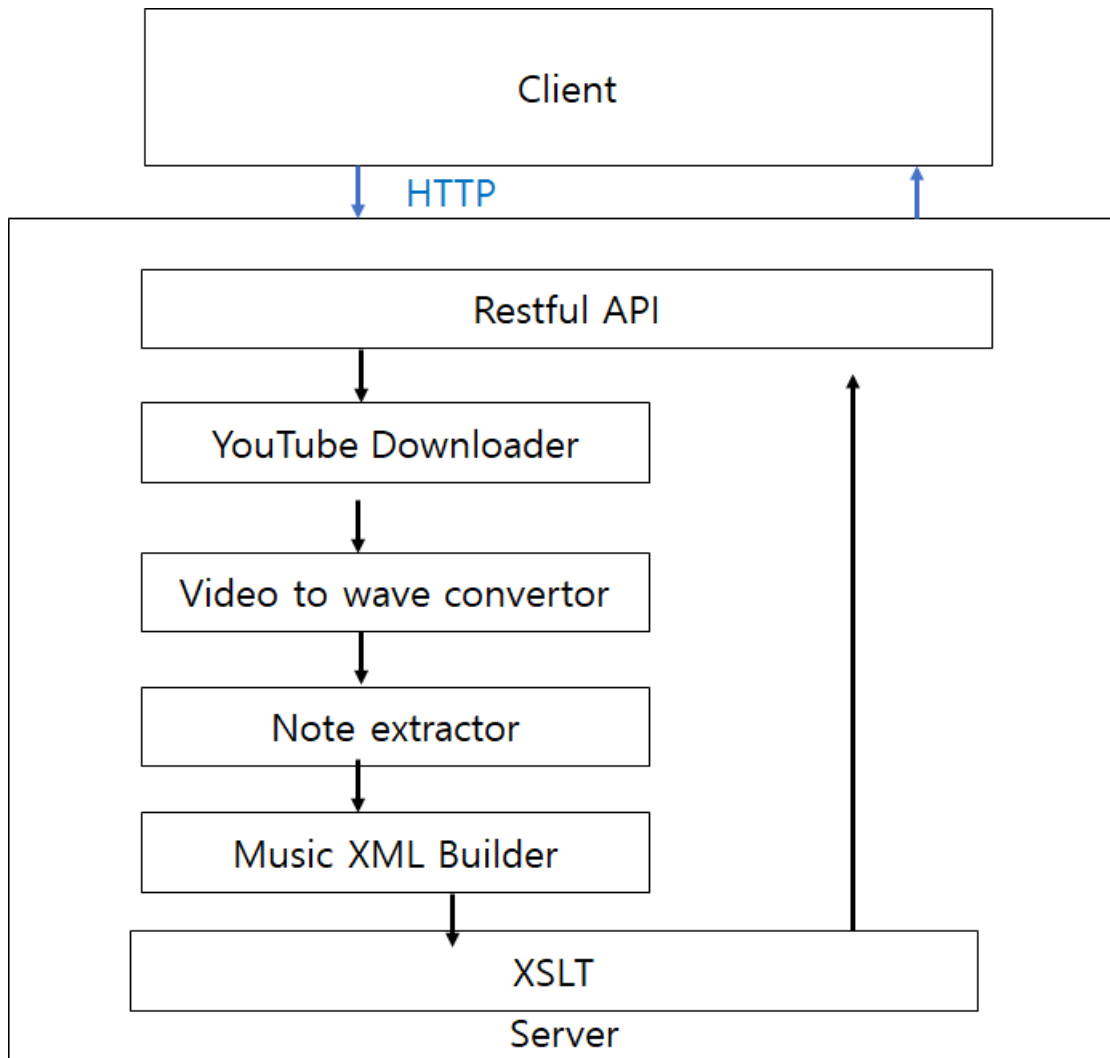
2.3.2 시스템 비기능(품질) 요구사항

URL의 음원에서 악보로 변환시에 기존시장의 프로그램들의 정확도 50%를 요구함.

예시 드라마 별 그대의 오프닝 곡 별에서 온 그대의 경우 기존 시장의 무료 제품들은 정확도가 50~60%로 심한 경우 음 율이 달라지는 경우 발생.


사유로는 악보의 3분의 1이 기존 곡과 다르면 연주 했을 시에 연주자와 청음자가 음원을 매우 다르게 느끼기에 기존 곡의 음 율을 악보로 정확히 구현하는 것이 최고 우선순위사항

2.3.3 시스템 구조



2.3.4 결과물 목록 및 상세 사양

| 대분류 | 소분류 | 기능 | 형식 | 비고 |
|-----|-------|--------------------------|----|----|
| 변환 | 파일 변환 | URL 의 동영상상을 오디오로 변환 | | |
| | | | | |
| | | | | |
| 출력 | 웹 페이지 | 음원의 운율을 웹의 화면에 악보로 출력 | | |
| 모양 | 악보 | 스크린에 출력 | | |

| | | | |
|---|-------------------------|-------------|-------------|
|  국민대학교 컴퓨터공학부 캡스톤 디자인 I | 계획서 | | |
| | 프로젝트 명 | | |
| | 팀 명 | | |
| | Confidential Restricted | Version 1.2 | 20xx-MAR-25 |

| | | | | |
|--|--|--|--|--|
| | | | | |
| | | | | |

2.4 기대효과 및 활용방안

당초 기획한 성능을 만족한다면, 동영상, 음원, URL 에서의 음원에서 악보를 추출 할 수 있을 것이며,

3 배경 기술

3.1 기술적 요구사항

- 서버
 - 운영체제
- Linux (ubuntu)
 - 사용언어
- python3.6
 - 프레임 워크
- flask
 - 관련 라이브러리
 - ◆ blueprint
 - ◆ pymysql
 - ◆ pytube
- 2. 클라이언트
- 플랫폼
- 웹
- 사용언어
- html, CSS, JavaScript
- 프레임워크
- angular js


3.2 현실적 제한 요소 및 그 해결 방안

3.2.1 하드웨어

1. 인터넷 여부

3.2.2 소프트웨어

1. 2개 이상의 악기는 구분이 힘들 수가 있음

| | | | |
|---|-------------------------|-------------|-------------|
|  국민대학교 컴퓨터공학부 캡스톤 디자인 I | 계획서 | | |
| | 프로젝트 명 | | |
| | 팀 명 | | |
| | Confidential Restricted | Version 1.2 | 20xx-MAR-25 |

- 악기별로 주파수 대역이 있으므로 그것으로 구분
- 간단한 멜로디 악기 솔로곡만 대상으로 선정
- 2. 사람의 목소리가 들어가 있을 경우 구분이 힘들다.
 - MR 구분 오픈 소스를 사용
 - 1번 문제와 마찬가지로 주파수 대역을 검사하여 구분한다.
- 3. 주파수 계산 및 악보 변환 속도가 20분 내외로 해야 한다.
 - 서버에서 계산을 수행하므로 서버의 개수를 증가
 - 주파수 분할 간격을 적절하게 정하여 계산 최소화

3.2.3 기타


1. 음악적 지식의 부족으로 악보 제작 시 어려움
 - 관련 서적 참고
 - 전문인에게 질문

4 프로젝트 팀 구성 및 역할 분담

| 이름 | 역할 |
|-----|------------------|
| 성정훈 | - Server Side 개발 |
| 조용현 | - Client Side 개발 |
| 변승현 | - Xslt 제작 |
| 임현재 | - 데이터 베이스 설계 |
| 위봉우 | - 프로젝트 테스트 |

5 프로젝트 비용

| 항목 | 예산치 (MD) |
|-------------------------------|----------|
| 클라이언트 개발 | 30 |
| 서버 개발 | 60 |
| xslt 제작 | 30 |
| Convert -YouTube URL to audio | 10 |
| Music XML 제작 | 20 |

| | | | |
|---|-------------------------|-------------|-------------|
|  국민대학교 컴퓨터공학부 캡스톤 디자인 I | 계획서 | | |
| | 프로젝트 명 | | |
| | 팀 명 | | |
| | Confidential Restricted | Version 1.2 | 20xx-MAR-25 |

| | |
|---|-----|
| 합 | 150 |
|---|-----|


6 개발 일정 및 자원 관리

6.1 개발 일정

| 항목 | 세부내용 | 1 월 | 2 월 | 3 월 | 4 월 | 5 월 | 6 월 | 비고 |
|--------|-------------|-----|-----|-----|-----|-----|-----|----|
| 요구사항분석 | 요구 분석 | | | | | | | |
| | SRS 작성 | | | | | | | |
| 관련분야연구 | 주요 기술 연구 | | | | | | | |
| | 관련 시스템 분석 | | | | | | | |
| 설계 | 시스템 설계 | | | | | | | |
| 구현 | 코딩 및 모듈 테스트 | | | | | | | |
| 테스트 | 시스템 테스트 | | | | | | | |

6.2 일정별 주요 산출물

| 마일스톤 | 개요 | 시작일 | 종료일 |
|-------------------|---|------------|------------|
| 계획서 발표 발표자료 준비 | 개발 환경 완성 및 발표 자료 준비 산출물 : 1. 프로젝트 수행 계획서 2. 프로젝트 기능 일람표 | 2018-03-02 | 2018-03-09 |
| 설계 완료 | 시스템 설계 완료 산출물 : 1. 시스템 설계 사양서 | 2018-03/09 | 2018-03-30 |
| 1 차 중간 보고 | 기능 Convert -YouTube URL audio 구현 완료 산출물 : 1. 프로젝트 1 차 중간 보고서 2. 프로젝트 진도 점검표 3. 1 차분 구현 소스 코드 | 03-09 | 04/12 |
| 2 차 중간 보고 | 기능 동영상 변환~ 클라이언트 구현 완료 산출물 : 1. 프로젝트 2 차 중간 보고서 | 2018-03/09 | 2018-04/13 |

| | | | |
|---|-------------------------|-------------|-------------|
|  국민대학교 컴퓨터공학부 캡스톤 디자인 I | 계획서 | | |
| | 프로젝트 명 | | |
| | 팀 명 | | |
| | Confidential Restricted | Version 1.2 | 20xx-MAR-25 |

| | | | |
|--------|--|--|--|
| | 2. ... | | |
| 구현 완료 | YouTube URL 음원으로부터 WAV 파일로 변환 완료 산출물: https://github.com/kookmin-sw/2018-cap1-14/tree/convert-youtube-url-to-audio 클라이언트 구현 완료 산출물: https://github.com/kookmin-sw/2018-cap1-14/tree/client 서버제작 구현완료 산출물: https://github.com/kookmin-sw/2018-cap1-14/tree/server FFT 1 차목표 완료 산출물: https://github.com/kookmin-sw/2018-cap1-14/tree/fft_test | | |
| 테스트 | YouTube URL 음원으로부터 WAV 파일로 변환 테스트 산출물: https://github.com/kookmin-sw/2018-cap1-14/tree/convert-youtube-url-to-audio-test | | |
| 최종 보고서 | 최종 보고 산출물: | | |


6.3 인력자원 투입계획

| 이름 | 개발항목 | 시작일 | 종료일 | 총개발일(MD) |
|-----|----------------------------|-----|-----|----------|
| 성정훈 | Server side | | | |
| 성정훈 | Convert -YouTube URL audio | | | |
| 조용현 | Client side | | | |
| 변승현 | Xslt | | | |
| 임현재 | 데이터 베이스 설계 | | | |
| 위봉우 | | | | |
| | | | | |

6.4 비 인적자원 투입계획

개발 환경 등 비 인적 자원의 투입 계획을 명시한다.

| 항목 | Provider | 시작일 | 종료일 | Required Options |
|----|----------|-----|-----|------------------|
|----|----------|-----|-----|------------------|

| | | | |
|---|-------------------------|-------------|-------------|
|  국민대학교 컴퓨터공학부 캡스톤 디자인 I | 계획서 | | |
| | 프로젝트 명 | | |
| | 팀 명 | | |
| | Confidential Restricted | Version 1.2 | 20xx-MAR-25 |

| | | | | |
|------------|--|------------|------------|--|
| 개발용 PC 5 대 | | 2018-03-02 | 2018-06-15 | |
| | | | | |
| | | | | |
| | | | | |

7 참고 문헌

| 번호 | 종류 | 제목 | 출처 | 발행년도 | 저자 | 기타 |
|----|-------|-------------------------------|---|-----------|----------------------|----|
| 1 | 웹페이지 | Audio signals in Python | http://myinspirationinformation.com/ | 2017.4.17 | <u>James Thomson</u> | |
| 2 | 논문 | 피아노 소리를 악보로 표현하는 시스템의 설계 및 구현 | 한국교육학술정보원 | 2012 | 도효환 | |
| 4 | 소프트웨어 | XML 을 악보로 변환시켜주는 소프트웨어 | Musescore.org | | | |
| 4 | 기술문서 | Aws Ec2 참조 문서 | https://docs.aws.amazon.com/ | | | |
| 5 | 기술문서 | AngularJS 참조 문서 | https://docs.angularjs.org/api | | | |
| 3 | 기술문서 | Flask 참조 문서 | http://flask-docs-kr.readthedocs.io/ko/latest/ | | | |