



# Music pet

## Software Requirement Specification

2022.04.30

**소프트웨어 공학 TEAM 3**

팀장 송홍빈

조원 임형선

조원 김지현

조원 장준

## CONTENTS

<b>1. Introduction.....</b>	<b>7</b>
1.1. Purpose.....	7
1.2. Scope.....	7
1.3. Definitions, Acronyms, and Abbreviation.....	7
1.4. References.....	8
1.5. Overview.....	9
<b>2. Overall Description.....</b>	<b>9</b>
2.1. Product Perspective.....	9
2.1.1. System Interface.....	9
2.1.2. User Interface.....	9
2.1.3. Hardware Interface.....	9
2.1.4. Software Interface.....	10
2.1.5. Memory Constraints.....	10
2.1.6. Operation.....	10
2.2. Product functions.....	10
2.2.1. 회원 가입.....	10
2.2.2. 연결 및 기기 작동 확인.....	11
2.2.3. 사용자 설정.....	11
2.2.4. 음악 재생.....	11
2.2.5. 음악 추천.....	11
2.3. User Characteristics.....	11
2.3.1. System Administrator.....	11
2.3.2. User.....	12

2.3.3. Vendor.....	12
2.4. Constraints.....	12
2.5. Assumptions and Dependencies.....	13
<b>3. Specific Requirements.....</b>	<b>13</b>
3.1. External Interface Requirements.....	13
3.1.1. User Interfaces.....	13
3.1.2. Hardware Interfaces.....	20
3.1.3. Software Interfaces.....	21
3.1.4. Communication Interfaces.....	21
3.2. Functional Requirements.....	21
3.2.1. Use Case.....	21
3.3. Performance Requirements.....	27
3.3.1. Static numerical requirement.....	27
3.3.2. Dynamic numerical requirement.....	28
3.4. Design Constraints.....	28
3.5. Standards Compliance.....	28
3.6. Software System Characteristics.....	28
3.6.1. Product Requirements.....	28
3.6.1.1. Usability Requirements.....	28
3.6.1.2. Efficiency Requirements.....	28
3.6.1.3. Dependability Requirements.....	29
3.6.1.4. Security Requirements.....	29
3.6.2. Organizational Requirements.....	29
3.6.2.1. Organizational Requirements.....	29
3.6.2.2. Environmental Requirements.....	29

3.6.3. External Requirements.....	29
3.6.3.1. safety/security Requirements.....	29
3.6.3.2. Regulatory Requirements.....	29
3.7. Organizing the specific requirement.....	29
3.7.1. Context Model.....	30
3.7.2. Process Model.....	30
3.7.3. Interaction Model.....	30
3.8. System Architecture.....	31
3.9. System Evolution.....	31
3.9.1. Assumption.....	31
3.9.2. Anticipated Changes.....	31
<b>4. Supporting Information.....</b>	<b>32</b>
4.1. Software Requirement Specification.....	32
4.2. Document History.....	32

## LIST OF TABLES

[Table 1] 용어 설명.....	7
[Table 2] 약어 설명.....	8
[Table 3] Main Interface.....	13
[Table 4] Play Interface.....	14
[Table 5] Playlist Interface.....	16
[Table 6] Play Situation Interface.....	17
[Table 7] Music Recommend Interface.....	19
[Table 8] 회원가입의 Use Case.....	21
[Table 9] 로그인의 Use Case.....	23
[Table 10] 기본 설정의 Use Case.....	23
[Table 11] 상황 설정의 Use Case.....	24
[Table 12] 음악 재생의 Use Case.....	25
[Table 13] 음악 추천의 Use Case.....	26
[Table 14] Document History.....	32

## LIST OF FIGURES

[Figure 1] Use Case Diagram.....	27
[Figure 2] Context Model.....	34
[Figure 3] Process Model.....	35
[Figure 4] System Architecture.....	36

## 1. introduction

### 1.1. Purpose

이 문서는 ‘Music pet’의 요구사항 명세서이다. ‘Music pet’은 기존 음악프로그램과는 다르게 사용자가 집에 들어오는 순간부터 매순간 현재 기분과 날씨를 측정해 그에 맞는 음악을 재생하는 서비스이다. 이 서비스는 소프트웨어 공학 3조에 의해 디자인되고 구현되었다. 이 문서의 목적은 ‘Music pet’의 요구사항을 명세하기 위함이다.

### 1.2. scope

‘Music pet’은 코로나블루로 인해 우울해하는 사람들이 음악을 통해 자신의 감정을 추스릴 수 있도록 현재 감정에 맞는 음악을 재생한다. ‘Music pet’은 사용자의 감정을 파악하기 위해 두 가지 디바이스를 사용한다. Raspberry Pi는 사용자의 혼잣말을 음성인식하여 사용자의 감정을 알아낸다. 또한 현재 두피열을 측정하는 아두이노 적외선 온도센서를 통해 기분을 파악하여 음악을 재생한다.

### 1.3. Definition, Acronyms, and Abbreviation

아래표는 이 문서에서 사용된 용어에 대한 설명이다.

[Table 1] 용어 설명

용어	설명
사용자	시스템을 이용하는 사람
‘Music pet’	설계하고자 하는 인공지능 음악추천 시스템.
소프트웨어	‘Music pet’의 앱
하드웨어	‘Music pet’의 기기

시스템	‘Music pet’
Vendor	공급업체

아래표는 이 문서에서 사용된 약어에 대한 설명이다.

[Table 2] 약어 설명

용어	정의
UI	User Interface
DB	Database
ID	Identification
AI	Artificial Intelligence
N/A	Not Applicable

## 1.4. references

- IEEE Recommendation (IEEE Recommended Practice for Software Requirements Specification, IEEE-Std-830)
- 2022spring\_AAI3007class\_team3(github.com)
- Software Engineering. Ian Sommerville, 2016.



## 1.5. Overview

두 번째 장은 ‘Music pet’의 전반적인 구조를 인터페이스, 시스템, 타 시스템 간의 상호작용을 포함한 다각적인 관점으로 분석한다. 이 챕터에서는 사용자, 시스템 매니저, 밴더로 이루어진 이해관계자를 소개하고 그들이 프로그램과 어떠한 관계로 얹혀 있는지 살펴볼 예정이다. 해당 어플리케이션의 전체적인 디자인을 명세하기에 앞서, 구동과정에 있을 제약사항과 가정, 타 시스템 간의 의존성에 대해 기술한다.

세 번째 장은 해당 소프트웨어 구현에 있어 필요한 구체적인 요구사항에 대해 설명한다. 해당 소프트웨어가 구동되는 시스템 환경 및 구조(데이터베이스/시스템 구조), 내/외부 인터페이스 묘사를 통해 소프트웨어 요구사항을 더욱더 구체화한다. 아울러 다양한 사용자가 개입되어 있는 Use Case들을 시각화하여 해당 프로그램의 구현방향과 요구사항 충족여부를 확인할 것이다.

## 2. Overall Description

### 2.1. Product Perspective

이 소프트웨어는 집에서 음악 감상을 즐겨하는 사람들을 위한 자동 음악 재생 프로그램이다. 이 프로그램은 사용자의 집 안에서 사용자의 기분에 맞는 음악을 자동으로 재생해주는 기능을 제공한다. 사용자가 직접 음악을 재생하기 전에 프로그램이 알아서 미리 설정된 상황이 되면 음악을 재생하며, 사용자가 평소 즐겨 듣던 음악 혹은 날씨를 반영한 음악 등 사용자의 기호에 맞는 음악을 재생한다.

#### 2.1.1. System Interface

사용자는 모바일 앱을 통해 시스템에 접근할 수 있다. 모바일 앱을 통해 자신이 듣고 싶은 음악을 설정하거나 수동으로 음악을 재생 또는 중지시킬 수 있으며, 유튜브 뮤직이나 멜론 등의 기존 음악 앱과 연동하여 사용자의 정보와 플레이리스트 등을 받아들 수 있다.

#### 2.1.2. User Interface

스마트폰의 화면을 통해 인터페이스가 출력되며, 사용자는 스마트폰을 통해 음악 재생을 제어하거나 플레이리스트를 생성, 수정할 수 있다. 3장에서 서술될 것처럼 음악을 재생하는 상황 등을 설정할 때도 각각의 기능에 맞는 인터페이스를 제공한다.

#### 2.1.3. Hardware Interface

이 소프트웨어는 Android 또는 iOS 기반의 스마트폰에서 작동 가능하다. 스마트폰에는 최소 1GB RAM과 1.0GHz 단일 프로세서가 있어야 한다.

#### **2.1.4. Software Interface**

이 소프트웨어는 Android 6.0 이상의 Android OS 버전, Android 10, iOS 13.0 이상의 iOS 버전을 대상으로 한다.

#### **2.1.5. Memory Constraints**

이 소프트웨어를 설치하기 위해서는 최소 344.8MB의 저장공간이 필요하다.

#### **2.1.6. Operations**

##### **■ 음악 재생/중지**

사용자는 수동으로 음악을 재생하거나 재생 중인 음악을 중지할 수 있다.

##### **■ 플레이리스트 추가/수정/삭제**

사용자가 듣고 싶은, 혹은 테마에 맞추어 플레이리스트를 작성하고 수정할 수 있다.

##### **■ 상황 설정**

사용자가 음악을 나오도록 하고 싶은 상황을 설정할 수 있다.

##### **■ 음악 볼륨 조절**

사용자가 재생되는 음악의 볼륨을 조절할 수 있다.

##### **■ 음악 추천**

현재 날씨 혹은 테마에 따라 음악을 추천하고 그 음악을 재생하거나 플레이리스트에 추가할 수 있다.

### **2.2 Product Functions**

#### **2.2.1. 회원가입**

사용자가 애플리케이션을 사용하기 위해선 음원 스트리밍 서비스의 계정이 필요하다. 스트

리밍 서비스 계정이 존재한다면 회원가입을 진행한다. 가입은 약관 및 개인정보 수집 동의와 사용자 정보 입력으로 이루어진다. 사용자 정보로는 이름과 이메일 주소, 비밀번호, 생년월일 등이 요구되며, 이메일 인증을 진행한다. 마지막으로 스트리밍 서비스의 계정과 연동시키면 가입이 완료된다.

#### **2.2.2. 연결 및 기기 작동 확인**

애플리케이션 설치 및 가입이 완료된 후, 사용자는 스피커를 애플리케이션과 연결할 수 있다. 스피커는 스마트폰과 블루투스로 연결한 후, 애플리케이션에 추가한다. 그리고 스피커의 적외선 센서와 음성 인식 센서가 사용자를 잘 인식하는지, 설정된 노래가 제대로 재생되는지 등 기기 작동 여부를 확인한다.

#### **2.2.3. 사용자 설정**

사용자는 애플리케이션을 통해 음악 재생 환경을 커스터마이징 할 수 있다. 원하는 상황에 따른 플레이리스트를 설정할 수 있으며 추가 기능 사용 여부도 결정할 수 있다. 상황은 시간, 위치 및 소리 등을 이용하여 설정 가능하다.

#### **2.2.4. 음악 재생**

사용자는 현재 위치에 따른 날씨, 감정 상태 그리고 사용자 설정에 맞게 음악을 들을 수 있다. 날씨와 감정 상태에 따른 음악 재생은 기본적으로 주어지는 기능이며, 사용자가 음악이 나오도록 하고 싶은 상황을 설정할 수도 있다. 사용자의 감정 상태는 스피커의 적외선 센서와 음성 인식 센서를 통해 파악되며, 분석된 감정을 바탕으로 음악이 재생된다. 사용자는 애플리케이션을 통해 어떤 노래가 재생되는지 확인할 수 있으며, 노래 재생, 정지 및 검색이 가능하다.

#### **2.2.5. 음악 추천**

사용자는 상황에 따른 음악 추천을 받을 수 있다. 각 상황마다 사용자가 설정한 플레이리스트를 기반으로 추천 시스템이 작동된다. 사용자가 음악을 넘길 경우, 넘긴 음악과 당시 상황에 관련된 데이터를 수집, 분석하여 이후 추천할 때 반영한다.

### **2.3. User Characteristics**

#### **2.3.1. System Administrator**

시스템관리자는 소프트웨어 엔지니어, AI 스피커 하드웨어 전문가, 프론트 엔드 개발자 이렇게 3명의 역할을 필요로 한다. 소프트웨어 엔지니어는 AI, Data science, Database에 관한 깊은 이해가 바탕이 되어야 하며, AI 스피커 하드웨어 전문가는 AI Speaker에 대한 이해를 기본으로 하며 추가로 빛과 사운드에 대한 지식 또한 필요로 한다. 프론트 엔드 개발자는 앱의 디자인을 주기적으로 변경할 수 있는 능력을 필요로 한다. 마지막으로 3명 모두 에러가 발생했을 시, 상황에 유연하게 대처할 수 있는 역량을 필요로 하며, 사용자의 요구사항을 재빠르게 파악하고 시스템에 반영할 수 있어야 한다.

### 2.3.2. User

문서에서 언급되는 ‘사용자’는 직장인 1인 가구를 뜻한다. 사용자가 우울증과 외로움, 불안감에 취약할 것으로 전제하였으며, 사람들 간의 소통과 위로를 필요로 할 것이라고 판단하였다. 작년 서울시 청년 1인 가구의 고충으로 외로움(32.5%)이 가장 많았고, 경제적 불안감(20.3%)이 뒤를 이었다. 청년 가구가 주요 사용자인 만큼, 사용자가 스마트폰과 전자기기를 다루는 것에 익숙할 것으로 가정하였으며, AI Speaker가 무엇인지 알고, 구동이 어떤 방식으로 이루어지는지 아는 선 정도의 지식을 필요로 한다.

### 2.3.3. Vendor

공급업체는 KT, 멜론, 네이버이다. KT는 기가지니 스피커에 사용되었던 인공지능, 스피커 등의 부품과 작동원리와 같은 정보를 제공한다. 멜론은 음악 데이터 정보를 네이버는 날씨 정보를 제공한다.

## 2.4. Constraints

이 시스템은 이 문서에 언급된 내용을 기반으로 설계하고 구현하며, 다음 사항을 준수한다.

- 인공지능 스피커는 60dB 소리이상부터 반응하게 한다.
- 두피열의 온도가 34도 이상부터 스트레스를 받았다고 가정한다.
- 오후 6시를 기준으로 이전은 출근, 이후에는 퇴근으로 취급한다.
- 전체 시스템성능을 개선한다.
- 시스템 비용 및 유지관리 비용을 고려한다.

- 본질에 집중하여 단순하고 이해하기 쉬운 코드를 작성한다. 유지보수와 시스템 개선에 용이하게 한다.
- 소스코드를 최적화하여 시스템 리소스 낭비를 방지한다.

## 2.5. Assumptions and Dependencies

이 문서의 소프트웨어는 App 기반에서 설계되고 구현된다는 가정 하에 만들어진다. 앱을 사용하기 위해서는 344.8MB의 공간을 필요로 한다. 또한 시스템은 2022년 4월에 업데이트된 버전을 기반으로 구현되었으며, 안드로이드와 iOS 체제, 2가지 체제를 기반으로 만들어졌기 때문에 다른 운영 체제나 버전에는 적용이 되지 않을 가능성이 있다. 음악 데이터는 멜론 음악 데이터를 따르기 때문에 멜론에 존재하지 않는 음악은 시스템을 통해 들을 수 없다. 날씨 데이터는 네이버 날씨 데이터를 활용하기 때문에 앱의 날씨예측 정확도는 네이버 날씨의 정확도를 따른다.


## 3. Specific Requirements

### 3.1. External Interface Requirements

#### 3.1.1. User Interfaces

[Table 3] Main Interface

	메인 인터페이스
Purpose/ Description	앱을 실행했을 경우 기본적으로 보여지는 인터페이스로, 사용할 기능을 고를 수 있다.
Input Source/ Output destination	사용자/앱을 실행하는 스마트폰 화면
Range/Accuracy /Margin of error	N/A
Unit	Screen
Time/Velocity	N/A

Relationship with other input/outputs	N/A
Format and configuration of screen	<p>1. 메인 화면은 각각의 기능들을 사용할 수 있는 버튼들과 현재 재생 중인 스피커의 위치를 나타내는 화면으로 구성되어 있다.</p> <p>2. 사용자는 각각의 버튼들을 누름으로써 그 기능을 사용할 수 있다.</p> 
Format and configuration of window	N/A
Data type	Button, Text, Image
Instruction type	누르는 버튼에 따른 명령어
Exit message	N/A

[Table 4] Play Interface

	음악 재생 인터페이스
--	-------------


Purpose/ Description	현재 재생 중인 음악을 표시하고 이를 중지하거나 다른 음악을 재생할 수 있는 인터페이스
Input Source/ Output destination	사용자/앱을 실행하는 스마트폰 화면
Range/Accuracy/ Margin of error	N/A
Unit	Screen
Time/Velocity	N/A
Relationship with other input/outputs	N/A
Format and configuration of screen	<p>1. 현재 재생 중인 음악과 재생 중인 스피커가 표시된다.</p> <p>2. 사용자는 버튼을 눌러 재생, 중지, 음악 변경을 할 수 있다.</p> 
Format and configuration of window	N/A

Data type	Button, Text, Image
Instruction type	누르는 버튼에 따른 명령어
Exit message	N/A

[Table 5] Playlist Interface


	플레이리스트 인터페이스
Purpose/ Description	사용자가 만들어 놓은 플레이리스트를 관리하거나 새로운 플레이리스트를 생성할 수 있는 인터페이스
Input Source/ Output destination	사용자/앱을 실행하는 스마트폰 화면
Range/ Accuracy/ Margin of error	N/A
Unit	Screen
Time/Velocity	N/A
Relationship with other input/outputs	N/A



Format and configuration of screen	<ol style="list-style-type: none"> <li>1. 현재 생성된 플레이리스트 목록을 보여준다.</li> <li>2. 생성된 플레이리스트에 음악을 추가하거나 삭제할 수 있다.</li> <li>3. 플레이리스트 추가 버튼을 사용해 새 플레이리스트를 생성할 수 있다.</li> </ol> 
Format and configuration of window	N/A
Data type	Button, Text, Image
Instruction type	누르는 버튼에 따른 명령어
Exit message	N/A

[Table 6] Play Situation Interface

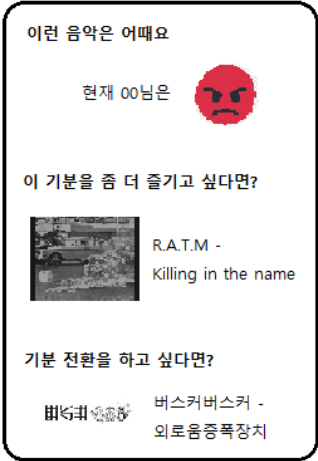
	상황 설정 인터페이스
Purpose/ Description	프로그램이 음악을 재생하는 상황을 설정하는 인터페이스

Input Source/ Output destination	사용자/앱을 실행하는 스마트폰 화면
Range/Accuracy/ Margin of error	N/A
Unit	Screen
Time/Velocity	N/A
Relationship with other input/outputs	N/A
Format and configuration of screen	<p>1. 현재 사용가능한 센서들을 표시한 화면과 현재 설정되어 있는 상황을 나타낸다.</p> <p>2. 추가/삭제 버튼을 누름으로써 음악을 재생하는 상황을 수정할 수 있다.</p> 
Format and configuration of window	N/A
Data type	Button, Text, Image

Instruction type	누르는 버튼에 따른 명령어
Exit message	N/A

[Table 7] Music Recommend Interface

	음악 추천 인터페이스
Purpose/Description	사용자의 기분에 맞는 음악을 추천하는 화면
Input Source/ Output destination	사용자/앱을 실행하는 스마트폰 화면
Range/Accuracy/ Margin of error	N/A
Unit	Screen
Time/Velocity	N/A
Relationship with other input/outputs	N/A

Format and configuration of screen	<ol style="list-style-type: none"> <li>수집한 데이터를 바탕으로 사용자의 기분을 나타낸다.</li> <li>사용자의 기분에 맞는 크게 두 가지 테마의 음악을 추천한다. <ul style="list-style-type: none"> <li>- 사용자의 기분을 증폭시키는 음악 테마</li> <li>- 사용자의 기분을 완화하는 음악 테마</li> </ul> </li> <li>사용자는 버튼을 눌러 바로 음악을 재생하거나 플레이리스트에 추가할 수 있다.</li> </ol> 
Format and configuration of window	N/A
Data type	Button, Text, Image
Instruction type	누르는 버튼에 따른 명령어
Exit message	N/A

### 3.1.2. Hardware Interfaces

이 소프트웨어는 Android 또는 iOS 기반의 스마트폰에서 작동 가능하다. 스마트폰에는 최소 1GB RAM과 1.0GHz 단일 프로세서가 있어야 한다. 또한 앱의 기능을 사용하기 위해서는 마이

크 접근 권한을 허용해야 하고 연결 가능한 스피커와 센서들이 있어야 한다.

### 3.1.3. Software Interfaces

이 앱을 사용하기 위해서는 멜론 또는 유튜브 뮤직과 같은 외부 음악 스트리밍 계정의 계정이 필요하다.

### 3.1.4. Communication Interfaces

사용자가 이 앱의 기능을 이용하기 위해서는 인터넷과 서버, 그리고 센서와 스피커에 연결되어야 한다. 사용자가 앱을 사용하는 동안 외부 음악 앱에 대한 로그인 세션이 유지되어야 한다.

## 3.2. Functional Requirements

### 3.2.1. Use Case

[Table 8] 회원가입의 Use Case

Use case name	회원가입
Actor	가입되지 않은 사용자
Description	가입이 되어 있지 않은 사용자의 가입을 진행한다.

Normal Course	<ol style="list-style-type: none"> <li>1. 애플리케이션을 작동시키면 홈 화면에 로그인과 회원가입 버튼이 있다.</li> <li>2. 사용자는 회원 가입 버튼을 누른다.</li> <li>3. 사용자는 약관 및 개인정보 수집 동의에 체크하고 사용자 정보를 입력한다. <ul style="list-style-type: none"> <li>- 사용자 이름</li> <li>- 이메일 주소</li> <li>- 비밀번호</li> <li>- 생년월일</li> </ul> </li> <li>4. 이메일 인증을 진행한다.</li> <li>5. 음원 스트리밍 서비스 계정을 연동시킨다.</li> <li>6. 가입이 완료된다.</li> </ol>
Precondition	<p>사용자는 시스템에 가입한 상태가 아니어야 한다.</p> <p>사용자는 올바른 정보를 기입해야 한다.</p> <p>중복되는 이메일 주소는 사용될 수 없다.</p> <p>올바르지 않은 입력이 들어왔을 때에는 이메일 인증을 통해 검증한다.</p> <p>음원 스트리밍 서비스의 계정이 필요하다.</p>
Post Condition	<p>모든 정보는 보안성을 지니는 데이터베이스에 저장되어야 한다.</p> <p>회원가입에 성공하면 홈 화면으로 이동한다.</p>

[Table 9] 로그인 Use Case

Use case name	로그인
Actor	가입된 사용자
Description	시스템에 가입된 사용자가 서비스를 사용하기 위해 시스템 접근을 시도하는 과정이다.
Normal Course	<ol style="list-style-type: none"> <li>1. 사용자가 이전에 로그인을 한 기록이 있는지 확인한다.</li> <li>2-1. 로그인을 한 기록이 있는 경우 <ul style="list-style-type: none"> <li>- 이전의 기록을 바탕으로 자동 로그인을 진행한다.</li> </ul> </li> <li>2-2. 로그인 한 기록이 없을 경우 <ul style="list-style-type: none"> <li>- 사용자에게 ID(혹은 가입 메일)와 비밀번호를 요구한다.</li> </ul> </li> <li>3. 입력 정보가 올바르다면 사용자가 시스템에 들어가 서비스를 이용할 수 있도록 한다.</li> </ol>
Precondition	사용자는 이미 가입된 상태여야 한다.
Post Condition	로그인을 완료하면 홈 화면으로 이동한다.

[Table 10] 기본 설정의 Use Case

Use case name	기본 설정
Actor	가입된 사용자
Description	성공적인 서비스 이용을 위해 스피커를 시스템과 연결하는 과정이다.

Normal Course	<ol style="list-style-type: none"> <li>1. 스피커 연결 버튼을 누른다.</li> <li>2. 디바이스에 연결된 기기 중 연결할 스피커를 누른다.</li> <li>3. 장소에 맞게 이름을 설정한다.</li> <li>4. 스피커 작동 여부를 확인한다.</li> <li>5. 스피커 연결이 완료된다.</li> </ol>
Precondition	<p>사용자는 시스템에 이미 가입된 상태여야 한다.</p> <p>사용자가 서비스를 사용할 공간에 스피커가 설치되어야 한다.</p> <p>스마트폰 등 애플리케이션이 설치된 디바이스와 스피커가 블루투스로 연결되어야 한다.</p>
Post Condition	-

[Table 11] 상황 설정의 Use Case

Use case name	상황 설정
Actor	가입된 사용자
Description	사용자의 취향에 맞게 서비스를 사용하기 위해 음악 재생 상황을 설정하는 과정이다.



Normal Course	<ol style="list-style-type: none"> <li>1. 홈화면의 음악 재생 상황 설정 버튼을 누른다.</li> <li>2. 추가 버튼을 누른 후 원하는 상황을 설정한다. <ul style="list-style-type: none"> <li>- 시간</li> <li>- 장소</li> <li>- 음성 인식 결과 (소리)</li> </ul> </li> <li>3. 원하는 플레이 리스트를 설정한다.</li> <li>4. 이름을 작성한 후 완료 버튼을 누른다.</li> <li>5. 상황 설정이 완료된다.</li> </ol>
Precondition	<p>사용자는 시스템에 로그인 된 상태여야 한다.</p> <p>스트리밍 계정 연결이 완료되어야 한다.</p> <p>사용자가 서비스를 사용할 공간에 스피커가 설치되어야 한다.</p> <p>스마트폰 등 애플리케이션이 설치된 디바이스와 스피커가 블루투스로 연결 되어야 한다.</p>
Post Condition	-

[Table 12] 음악 재생의 Use Case

Use case name	음악 재생
Actor	가입된 사용자
Description	음악 재생 상황에 대한 사용자의 설정에 따라 음악을 재생하는 과정이다.

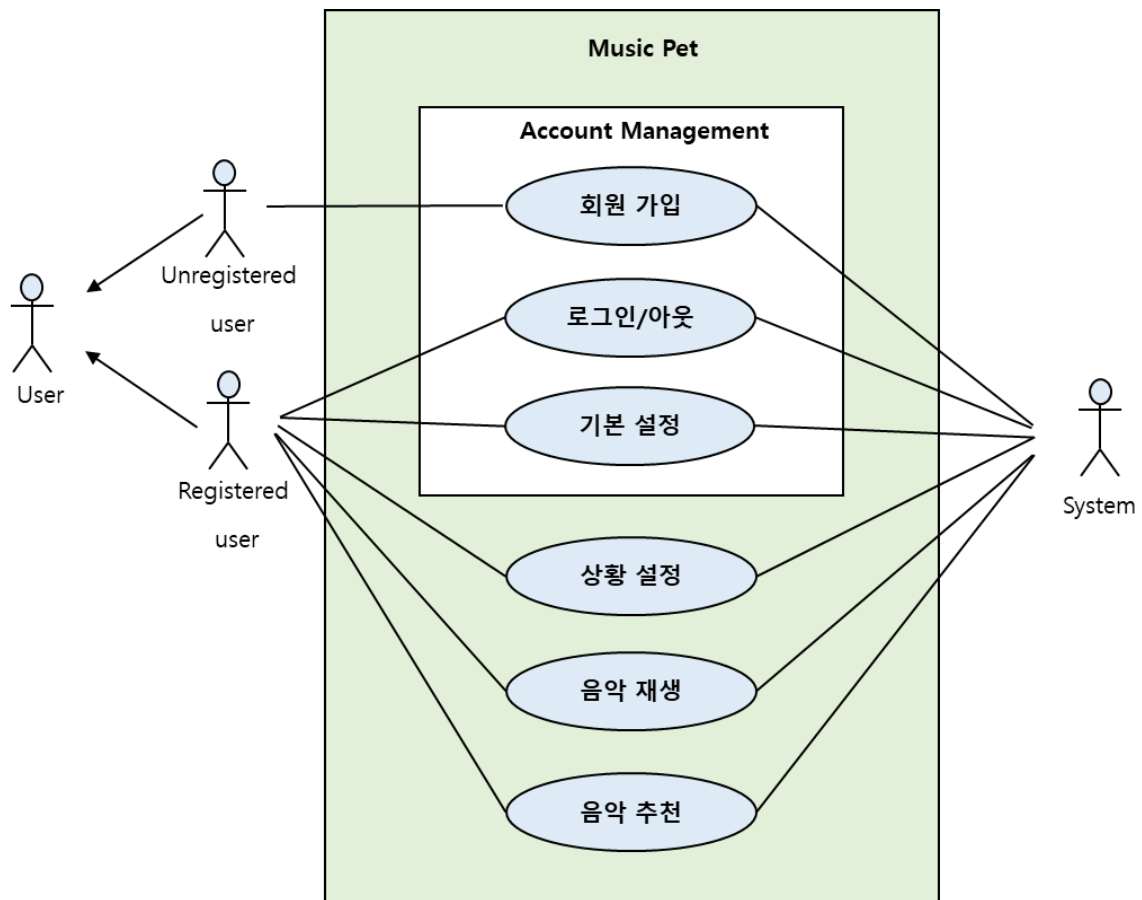
Normal Course	<ol style="list-style-type: none"> <li>1. 상황 설정의 조건을 만족하는지 확인한다.</li> <li>2. 설정에 맞는 플레이 리스트를 재생한다.</li> </ol>
Precondition	<p>사용자는 시스템에 로그인 된 상태여야 한다.</p> <p>시스템이 사용자의 음원 스트리밍 서비스 계정과 연결된 상태여야 한다.</p> <p>사용자가 서비스를 사용할 공간에 스피커가 설치되어야 한다.</p> <p>스마트폰 등 애플리케이션이 설치된 디바이스와 스피커가 블루투스로 연결되어야 한다.</p> <p>사용자는 위치 정보 제공에 동의해야 한다.</p>
Post Condition	현재 재생되는 노래 정보가 홈 화면으로 전달되어야 한다.

[Table 13] 음악 추천의 Use Case

Use case name	음악 추천
Actor	가입된 사용자
Description	현재 날씨나 테마에 맞는 음악 추천을 받고, 그 음악을 재생하거나 플레이 리스트에 추가하는 과정이다.
Normal Course	<ol style="list-style-type: none"> <li>1. 음악 추천 화면을 누른다.</li> <li>2. 테마별로 추천된 음악을 확인한다.</li> <li>3. 버튼을 눌러 노래를 재생하거나 플레이 리스트에 추가한다.</li> </ol>

Precondition	<p>사용자는 시스템에 로그인 된 상태여야 한다.</p> <p>시스템이 사용자의 음원 스트리밍 서비스 계정과 연결된 상태여야 한다.</p>
Post Condition	-

### 3.2.2. Use Case Diagram



[Figure 1] Use Case Diagram

### 3.3. Performance Requirements

이 문서의 성능 요구사항은 시스템 성능의 수치 값을 토대로 설명되어진다.

#### 3.3.1. Static numerical requirement

- iPhone, iPod – IOS 10.0 이상이 필요하다.

- MAC – macOS 11.0 이상 및 Apple M1 칩 탑재 Mac이 필요하다.
- Android – 안드로이드 버전 5.1 이상이 필요하다.

### 3.3.2. Dynamic Numerical Requirement

- 로그인 프로세스는 2초 이내에 완료되어야 한다.
- 하드웨어는 200ms 이내에 반응해야 한다.
- 음악 추천 시스템은 2초 이내에 추천을 완료해야 한다.
- 하드웨어는 60dB 소리 이상부터 반응한다.
- 하드웨어는 150W 빛 이상부터 반응한다.

## 3.4. Design Constraints

앱과 하드웨어 모두 미니멀한 디자인을 추구한다. 시스템이 사용자에게 편안함과 안정감 그리고 친근함을 선사하기 위한 목적성을 띄고 있기 때문에 자연의 테마를 활용한다. 앱의 메인 색은 초록색으로 하되, 나무 색감과 단풍 색감을 활용하여 완성한다.

## 3.5 Standards Compliance

시스템의 모든 프로그램은 안드로이드와 Swift를 통해 각각 Android와 IOS 지원하는 앱을 개발한다. 서버DB는 MySQL을 사용하며, Git을 통해 프로젝트 관리를 진행한다. 마지막으로 Docker 컨테이너를 활용하여 애플리케이션에 필요한 환경을 구축한다.

## 3.6 Software System Characteristics

### 3.6.1. Product Requirements

#### 3.6.1.1. Usability Requirements

보다 미니멀한 UI를 형성하여 사용자가 직관적으로 필요요소들을 찾을 수 있어야 한다. 사용자 인터페이스는 목적에 맞는 디자인적 요소와 버튼들의 효율적인 배치를 통해 별도의 설명서 없이도 모든 기능을 쉽게 사용할 수 있어야 한다.

#### 3.6.1.2. Efficiency Requirements

사용자가 질문했을 시, 하드웨어 ‘Music pet’이 200ms 이내에 반응하도록 한다. 응답시간이 200ms를 초과했을 시, 사용자가 대화에서 부자연스러움을 느끼게 된다. 전등을 켤 때, 문을 열었을 때, 모두 이 기준을 따른다. 음악을 추천해줄 때 소요되는 시간은 최대 2s로 한다.

### 3.6.1.3. Dependability Requirements

이 시스템의 음악 추천 시스템은 사용자의 신뢰를 필요로 한다. 사용자로부터 추출할 수 있는 두피열 데이터와 음성 데이터들과 정밀한 머신 러닝 기법을 통해 사용자에게 추천해주는 음악과 사용자가 실질적으로 원하는 음악이 일치하도록 한다.

### 3.6.1.4. Security Requirements

관리자는 철저한 본인 인증 후, 관리자 권한에 액세스할 수 있다. 권한이 없는 자가 시스템정보, 사용자의 개인정보에 접근하는 것을 방지한다. DB에 저장되어 있는 패스워드는 암호화되어 관리자측 또한 판별할 수 없으며, 사용자가 로그인할 때 사용자를 식별하는 용도로만 활용된다.

## 3.6.2 Organizational Requirements

### 3.6.2.1. Organizational Requirements

- 소프트웨어 진흥법을 준수한다.
- 신용정보의 이용 및 보호에 관한 법률을 준수한다.
- 개인정보보호법과 공정거래법을 준수한다.

### 3.6.2.2. Environmental Requirements

‘Music pet’의 목소리는 Naver CLOVA AI voice를 활용한다. 사용자가 AI voice를 직접 선택할 수 있다. 인공지능 스피커는 60dB 소리이상부터 반응하고, 조명은 150W 이상부터 반응한다.

## 3.6.3. External Requirements

### 3.6.3.1. Safety/Security Requirements

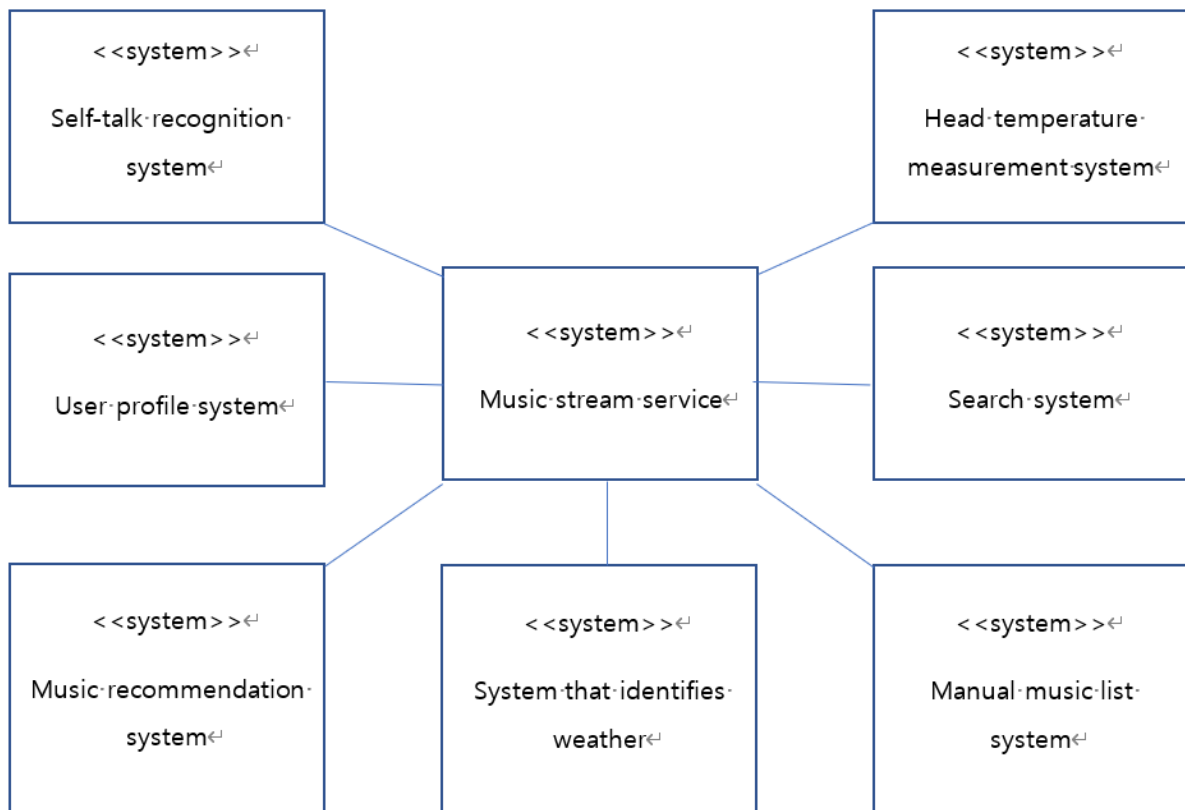
사용자의 데이터 보호를 최우선으로 한다. DB와 웹 서버 데이터를 주기적으로 백업한다. 외부 시스템, 권한이 없는 사용자의 접근 등 외부 침입으로부터 데이터를 안전하게 보호하고 데이터가 유출되는 상황을 막는다.

### 3.6.3.2. Regulatory Requirements

회원가입시, 하드웨어에서 얻어온 사용자의 두피열 데이터와 사용자의 음악 취향에 관한 메타 데이터를 활용하는 것에 대한 동의를 받는다. 이를 거절할 경우, 날씨에 따른 음악추천 시스템만 제공한다.

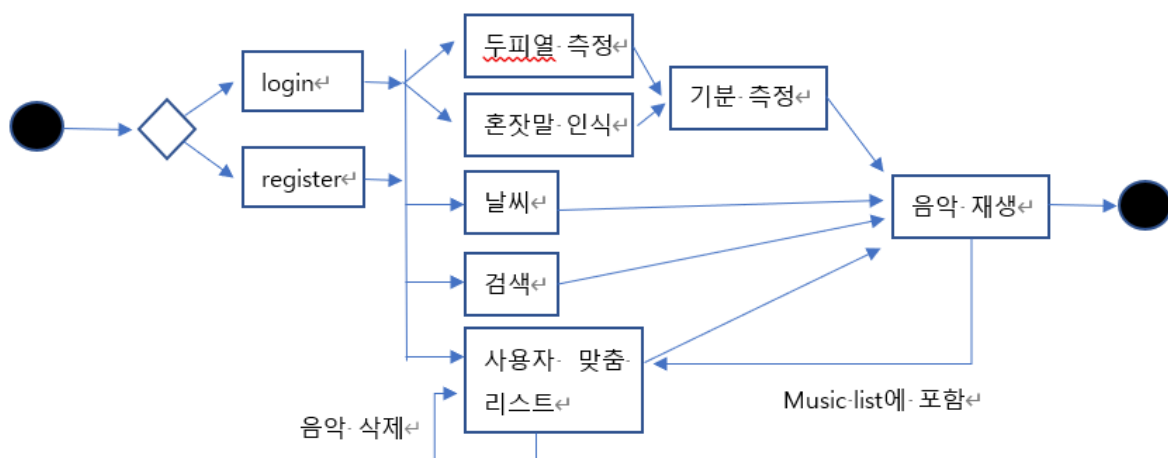
## 3.7. Organizing the specific requirement

### 3.7.1. context model



[Figure 2] Context Model

### 3.7.2. Process Model

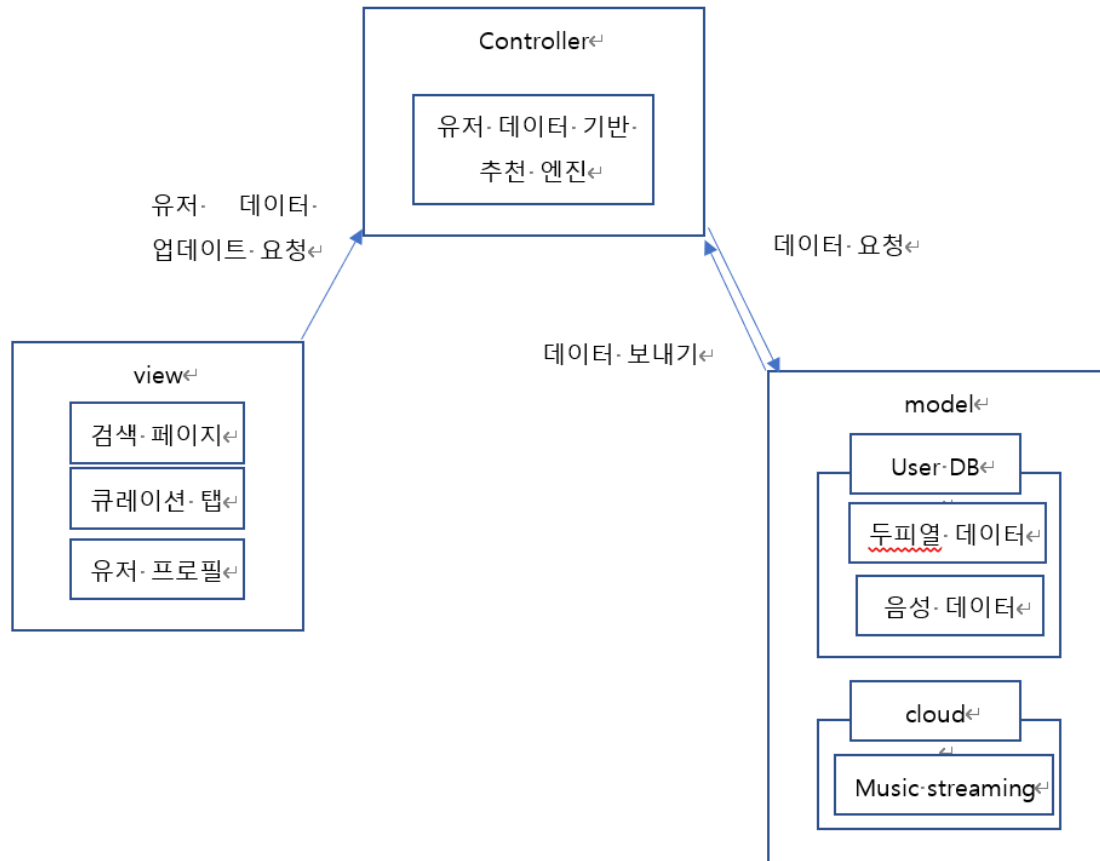


[Figure 3] Process Model

### 3.7.3. Interaction Model

See 3.2.2. Use Case Diagram

### 3.8. System Architecture



[Figure 4] System Architecture

### 3.9. System Evolution

#### 3.9.1. Assumption

‘music pet’은 코로나로 인한 장기화된 격리 상황에 지친 이들을 달래주기 위해 음악을 추천해 준다. 기존 음악 스트리밍 서비스가 제공하는 수동적인 검색 시스템의 불편함을 개선하고자 사용자 데이터 기반 추천 솔루션을 설계하였다.

#### 3.9.2. Anticipated Changes

기기 성능 발전 및 다양화되는 추세에 따라 사용자 데이터 측정 장비가 다양화될 것으로 예상된다. 그리고 그 다양한 장비와 기기들의 호환성을 고려하여 데이터 가공 및 정규화하는 과정이 변화할 수 있다.

사회문화적 현상에 의해 변화되는 사람들의 취향을 반영한 음악 추천 알고리즘이 달라질 것으로 예상된다.

## 4. supporting information

### 4.1. Software Requirement Specification

이 요구사항은 IEEE Recommendation(IEEE Recommended Practice for Software Requirements Specification, IEEE-Std-830)을 따라 작성하였다.

### 4.2. Document History

[Table 14] Document History

날짜	버전	설명	작성자
2022/4/22	0.1	문서 작업 시작	전체
2022/4/23	1.1	1.1, 1.2, 1.5	임형선
2022/4/23	1.2	2.1	장준
2022/4/23	1.3	2.2	김지현
2022/4/23	1.4	2.3, 2.4, 2.5	송홍빈
2022/4/28	1.5	3.1	장준
2022/4/28	1.6	3.2	김지현
2022/4/28	1.7	3.3~3.6, 1.3, 1.4	송홍빈
2022/4/28	1.8	3.7~3.9, 4	임형선



2022/4/29	2.0	1 개정	전체
2022/4/29	2.1	2 개정	전체
2022/4/29	2.2	3 개정	전체