

---

# **<VTUNES> TEST PLAN**

---

Version <1.0>

<2023/12/01>

김영현, 김지훈, 박종윤, 이용우, 최성훈

github : <https://github.com/Money-Juicer/VTunes>

설치파일 링크 : <https://github.com/Money-Juicer/VTunes/releases/tag/0.1.0>

---

# TABLE OF CONTENTS

## **1. Introduction**

1.1 Purpose of The Test Plan Document.....	3
--	---

## **2. Unit Test**

2.1 Test Risks / Issues .....	3
2.2 Test Approach(s) .....	3
2.3 Test Pass / Fail Criteria .....	4
2.4 Test Environmental / Staffing / Training Needs .....	4
2.5 Test Result.....	4

## **3. Integration Test**

3.1 Test Risks / Issues .....	5
3.2 Test Approach(s) .....	5
3.3 Test Pass / Fail Criteria .....	5
3.4 Test Environmental / Staffing / Training Needs .....	6
3.5 Test Result.....	6
3.6 Test Coverage.....	7

## **4. End To End Test**

4.1 Test Risks / Issues .....	9
4.2 Test Approach(s) .....	9
4.3 Test Pass / Fail Criteria .....	9
4.4 Test Environmental .....	9
4.5 Test Result.....	9

## **5. Performance Test**

5.1 Test Risks / Issues .....	13
5.2 Test Approach(s) .....	13
5.3 Test Pass / Fail Criteria .....	14
5.4 Test Environmental / Staffing / Training Needs .....	15
5.5 Test Result.....	14

## **6. Usability Test**

6.1 Test method .....	17
6.2 Test Risks / Issues .....	17
6.3 Test Approach(s) .....	17
6.4 Test Pass / Fail Criteria .....	17
6.5 Test Environmental / Staffing / Training Needs .....	17
6.6 Test Result.....	17

# 1 INTRODUCTION

## 1.1 PURPOSE OF THE TEST PLAN DOCUMENT

이 Test Plan Document 의 목적은 당사의 뮤직 플레이어 Electron 앱인 VTunes 의 품질, 신뢰성 및 기능을 평가하고 보장하기 위한 종합적인 전략을 설명하는 것이다. 이 문서는 체계적인 테스트를 통해 VTunes 내의 모든 구성 요소, 기능 및 상호 작용이 의도한 대로 작동하는지 확인하여 원활하고 즐거운 사용자 경험을 보장하는 것을 목표로한다. 테스트 프로세스를 안내하는 청사진으로 다양한 시나리오, 사용 사례 및 에지 사례를 포함하여 잠재적인 문제를 식별하고 수정한다. 또한 이 계획은 다양한 플랫폼 및 환경에서 VTunes 의 성능, 응답성 및 안정성을 평가하기 위한 명확한 기준과 측정 기준을 수립하는 것을 목표로한다. 궁극적으로 VTunes 를 위한 테스트 플랜은 당사의 뮤직 플레이어가 사용자의 기대와 요구를 충족시켜 우수한 청취 경험을 제공할 수 있도록 높은 수준의 우수성을 유지하기 위해 노력한다.

# 2 UNIT TESTING

## 2.1 TEST RISKS / ISSUES

본 프로젝트는 react 를 사용하는 어플리케이션 개발 프로젝트다. 이러한 특성 때문에 소스코드 대부분은 컴포넌트를 구성하고, 이들이 DOM 트리를 따라 유기적으로 연결되어 작동한다. 따라서 하나의 컴포넌트만 독립적으로 실행하는 것이 어려워 Unit test 에는 한계가 있다.

본 테스트 단계에서는 독립적인 환경에서 테스트할 수 있는 musicController 모듈의 state 를 redux 로 제대로 관리가 되는지 테스트한다.

## 2.2 TEST APPROACH(S)

본 어플리케이션의 작동 상태는 musicController 모듈의 state 로 정의 되어있다. 이 musicController 의 state 는 redux 로 관리된다. 본 테스트에서는 각 reducer 를 독립적인 환경에서 실행하고, 의 e 도된 대로 작동하여 예상된 결과를 내는지 테스트한다.

## 2.3 TEST PASS / FAIL CRITERIA

musicController 의 state 에 대한 reducer 를 실행했을 때 의도된 대로 작동하여 예상한 결과를 낸다면 test 를 통과했다고 본다. 반대로 예상하지 못한 결과를 낸다면 test 를 통과하지 못했다고 본다.

## 2.4 TEST ENVIRONMENTAL / STAFFING / TRAINING NEEDS

Environmental: jest 를 사용하여, test.js 파일로 작성된 테스트 코드를 실행시킨다.  
테스트 실행 명령어는 "react-scripts test --coverage --silent"이다.

## 2.5 TEST RESULT

Test Case	Description (Expected Output)	Result
LOAD_ALL_SUCCESS	musicController 가 load 를 성공하여 listOfPlaylist 에 mocking 한 플레이리스트가 있다.	Passed
LOAD_ALL_FAILURE	load 실패 시 listOfPlaylist 가 비어 있다.	Passed
ADD_PLAYLIST_SUCCESS	listOfPlaylist 에 mocking 한 새로운 플레이리스트가 들어가 있다.	Passed
ADD_PLAYLIST_FAILURE	listOfPlaylist 는 여전히 비어 있다.	Passed
DELETE_PLAYLIST_SUCCESS	Mocking 한 listOfPlaylist 에서 삭제한 Playlist2 가 존재하지 않는다..	Passed
DELETE_PLAYLIST_FAILURE	ListOfPlaylist 에 어떠한 수정도 일어나지 않는다.	Passed
CHANGE_SELECTED_PLAYLIST_SUCCESS	현재 선택된 플레이리스트가 Playlist1 에서 Playlist2 로 변경된다.	Passed
CHANGE_SELECTED_PLAYLIST_FAILURE	현재 선택된 플레이리스트가 Playlist1 에서 변경되지 않는다.	Passed
PREVIOUS_MUSIC (when current music is not the first in the playlist)	선택된 음악이 'Song2'에서 'Song1'로 변경된다.	Passed
PREVIOUS_MUSIC (when current music is the first in the playlist)	선택된 음악이 'Song1'에서 'Song3'로 변경된다.	Passed
NEXT_MUSIC (when current music is not the last in the playlist)	선택된 음악이 'Song2'에서 'Song3'로 변경된다.	Passed
NEXT_MUSIC (when current music is the last in the playlist)	선택된 음악이 'Song3'에서 'Song1'로 변경된다.	Passed
ADD_MUSIC_SUCCESS	플레이리스트에 'NewSong'이 추가된다.	Passed

ADD_MUSIC_FAILURE	플레이리스트에 어떠한 수정도 일어나지 않는다.	Passed
DELETE_MUSIC_SUCCESS	삭제된 음악이 playlist 에 존재하지 않는다.	Passed
DELETE_MUSIC_FAILURE	playlist 에 어떠한 수정도 일어나지 않는다.	Passed
MOD_SHOW_TIMER_BOX	IsShowTimerBox 가 false 에서 true 로, 다시 true 에서 false 로 변경	Passed
SET_IS_CURRENT_PLAYLIST_VIEWED	isCurrentPlaylistViewed 가 false 에서 true 로, 다시 true 에서 false 로 변경	Passed
SET_MUSIC_PLAYER_REF	musicPlayerRef 가 설정한 newRef 로 설정됨	Passed
SET_REST_TIME	restTime 이 60 에서 1 감소해서 59 로됨.	Passed
SET_IS_START_REDUCE_TIME	isStartReduceTime 이 false 에서 true 로 변경이 됨.	Passed

**PASS** src/tests/musicController.test.js

### 3 INTEGRATION TESTING

#### 3.1 TEST RISKS / ISSUES

음악 재생과 관련된 기능은 이미 검증된 라이브러리를 사용하여 개발하였다. 따라서 음악 재생 관련 기능을 테스트 하는 것은, 이미 검증된 라이브러리를 다시 테스트하는 것이기 때문에 불필요하다고 판단한다.

#### 3.2 TEST APPROACH(S)

Jest 를 사용하여 테스트할 컴포넌트를 렌더링하고, 잘 렌더링이 되는지와 이벤트 호출이 일어나는지 검사한다.

#### 3.3 TEST PASS / FAIL CRITERIA

컴포넌트가 제대로 렌더링 되고, 버튼 역할이 있다면 클릭 시 이벤트를 잘 호출할 수 있어야 테스트가 통과된다.

#### 3.4 TEST ENVIRONMENTAL / STAFFING / TRAINING NEEDS

Environmental: jest 를 사용하여, test.js 파일로 작성된 테스트 코드를 작성하였다.

React 요소를 테스트하기 위해 react 테스트 도구 (@testing-library/react)를 사용하였다.

테스트 실행 명령어는 "react-scripts test --coverage --silent"이다.

### 3.5 TEST RESULT

Test Case	DESCRIPTION (EXPECTED OUTPUT)	RESULT
Adder	Adder 컴포넌트가 정확하게 렌더링된다. Adder 버튼이 눌렀을 때, selectMusicFile 이벤트를 호출한다. Remover 버튼이 눌렀을 때, onIsDeleteClick 이벤트를 호출한다.	PASSED
Album	앨범 영역을 눌렀을 때 가사가 출력된다.	PASSED
ControlBar	ControlBar 를 정확하게 렌더링한다.	PASSED
durationHelper	초단위를 분단위로 정확하게 변환한다.	PASSED
index	Redux 와 HelmetProvider(title 을 바꾸는 라이브러리)와 함께 App 컴포넌트가 정확하게 렌더링된다.	PASSED
LyricsLine	LyricsLine 컴포넌트가 정확하게 렌더링된다.	PASSED
MusicItem	MusicItem 컴포넌트가 정확하게 렌더링된다.	PASSED
PlaylistItem	PlaylistItem 컴포넌트가 정확하게 렌더링된다.	PASSED
PlaylistSet	PlaylistSet 컴포넌트가 정확하게 렌더링된다. New Playlist 버튼을 눌렀을 때, Confirm 버튼과 Cancel 버튼을 정확하게 렌더링한다. Confirm 버튼을 눌렀을 때, 플레이리스트 이름을 전달하며 onAddPlaylist 이벤트를 호출한다. 만약 플레이리스트 이름이 invalid 하다면, 'Invalid Playlist Name'라는 경고 메시지를 띄운다.	PASSED

PlatlistSetContainer	PlaylistSetContainer 컴포넌트가 정확하게 렌더링된다.	PASSED
ScrollList	ScrollList 컴포넌트가 정확하게 렌더링된다.	PASSED
SearchEngine	SearchEngine 컴포넌트가 정확하게 렌더링된다.	PASSED
SideContainerContents	SideContainerContents 컴포넌트가 정확하게 렌더링된다.	PASSED
SideContainerFooter	SideContainerFooter 컴포넌트가 정확하게 렌더링된다.	PASSED
TimerBox	TimerBox 컴포넌트가 정확하게 렌더링된다.	PASSED
TimerBoxContainer	TimerBoxContainer 컴포넌트가 정확하게 렌더링된다.	PASSED
TimerSetting	TimerSetting 컴포넌트가 정확하게 렌더링된다. Timer 버튼을 눌렀을 시, onShowTimerBox 이벤트가 호출된다.	PASSED
VTunes	VTunes 컴포넌트가 정확하게 렌더링된다.	PASSED

```

PASS src/tests/TimerBox.test.js (5.892 s)
PASS src/tests/Adder.test.js (6.004 s)
PASS src/tests/PlaylistSet.test.js (6.502 s)
PASS src/tests/TimerSetting.test.js
PASS src/tests/ControlBar.test.js
PASS src/tests/PlaylistItem.test.js
PASS src/tests/index.test.js
PASS src/tests/PlaylistSetContainer.test.js
PASS src/tests/LyricsLine.test.js
PASS src/tests/SideContainerContents.test.js
PASS src/tests/VTunes.test.js
PASS src/tests/SearchEngine.test.js
PASS src/tests/TimerBoxContainer.test.js
PASS src/tests/MusicItem.test.js
PASS src/tests/ScrollList.test.js
PASS src/tests/durationHelper.test.js

```

### 3.6 TEST COVERAGE

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
<b>All files</b>	<b>61.2</b>	<b>57.2</b>	<b>51.87</b>	<b>62.35</b>	
src	100	57.14	100	100	
App.js	100	100	100	100	
TimerBox.jsx	100	100	100	100	
VTunes.jsx	100	57.14	100	100	16-28
src/component	55	50	52.63	56.36	
ControlBar.jsx	55	50	52.63	56.36	...03-104,121-153
src/component/Container	55.55	0	66.66	57.14	
LyricsLine.jsx	55.55	0	66.66	57.14	7-9
src/component/Container/SideController	76.13	80.95	52.77	81.01	
PlaylistItem.jsx	72.72	80	33.33	80	31-32,51-52
PlaylistSet.jsx	80.55	86.66	64.28	84.37	28,49,55-56,84
SearchEngine.jsx	73.33	76.47	53.84	77.77	34,64-65,81-85
src/component/Container/SideController/SideContainer	32.87	34.69	35	34.78	
MusicItem.jsx	68	56.52	62.5	70.83	19-24,34,56-57
SideContainerContents.jsx	14.58	15.38	16.66	15.55	...92,102,112-138
...t/Container/SideController/SideContainer/SideContainerFooter	46.34	53.33	32.25	48	
Adder.jsx	48.88	45.45	46.15	50	...,92-96,107-108
TimerSetting.jsx	43.24	60.86	22.22	45.45	35-36,40-42,58-89
src/component/common/ScrollList	100	100	100	100	
ScrollList.jsx	100	100	100	100	
src/containers	61.11	100	41.66	56.25	
PlaylistSetContainer.jsx	60	100	42.85	55.55	16-19
TimerBoxContainer.jsx	62.5	100	40	57.14	12-14
src/modules	67.69	67.24	58.49	68.44	
musicController.js	67.55	67.24	58.49	68.27	...44,461,469-485
rootReducer.js	100	100	100	100	
src/tests	100	100	100	100	
ReduxProviderWrapper.js	100	100	100	100	
Test Suites: 17 passed, 17 total					
Tests: 47 passed, 47 total					
Snapshots: 0 total					
Time: 22.271 s					

Jset 에 coverage 옵션을 주어 coverage 를 측정했다.

전체 소스 파일에 대해서 Branch 는 57.2%를 커버했고, Lines 는 62.35%를 커버했다.

## 4 END TO END TESTING

### 4.1 TEST RISKS / ISSUES

본 테스트는 Cypress 라는 웹 애플리케이션용 프론트엔드 테스트 도구를 사용한 테스트이다. Cypress 는 격리된 시뮬레이션 환경에서 테스트를 진행하기 때문에, 테스트 시에 본 어플리케이션이 파일을 저장하거나, 로드하기 위해 File System 에 접근할 수 없다. 따라서 파일을 저장하거나, 로드하여 진행하는 테스트는 한계가 있다.

### 4.2 TEST APPROACH(S)

본 테스트는 Cypress 라는 웹 애플리케이션용 프론트엔드 테스트 도구를 사용한 테스트이다. Cypress 를 이용하여 실제 유저가 인터페이스를 조작하는 경우를 시뮬레이션 하여 의도와 같이 UI 가 반응을 하는지 테스트할 것이다.



### 4.3 TEST PASS / FAIL CRITERIA

Cypress 를 이용하여 시뮬레이션 했을 때 예상된 UI 의 변화와 부합해야 통과한다.

### 4.4 TEST ENVIRONMENTAL

Environmental : 테스트를 실행하게 될 경우 컴파일시에 포함 파일만 적용 후 격리된 시뮬레이션 환경에서 테스트를 진행하기 때문에 빌드 이후 로딩 하지 않고 컴파일시 자동으로 포함되는 가상의 파일을 생성하여 환경을 만들어야 한다.

첨부된 파일(vtunes\_cypress)을 통해 test 를 실행하거나, git 에서 clone 을 받아 test branch 로 이동 후, 진행하면 된다. 테스트 명령어는 'npm start'를 통해서 VTunes 를 먼저 실행하고, 'npx cypress open'을 이용하여 실행한다.

Git 주소: <https://github.com/Money-Juicer/Vtunes.git>

### 4.5 TEST RESULT

#### **Renders the default screen elements**

Test	Expectation	Output	Result
Main controls button	Rendered	Rendered	Passed
Volume controls	Rendered	Rendered	Passed
Volume button	Rendered	Rendered	Passed
Additional controls	Rendered	Rendered	Passed
Progress section	Rendered	Rendered	Passed
Adder button	Rendered	Rendered	Passed
Remover button	Rendered	Rendered	Passed
Playlist menu	Rendered	Rendered	Passed
Timer button	Rendered	Rendered	Passed
Shuffle button	Rendered	Rendered	Passed
Repeat button	Rendered	Rendered	Passed
Search bar	Rendered	Rendered	Passed
Current_pl_btn(inactive)	Rendered	Rendered	Passed
Show_lyrics button	Rendered	Rendered	Passed
Show_album button	Not Rendered	Not Rendered	Passed

#### **Creates playlist (UI Test)**

Test	Expectation	Output	Result
------	-------------	--------	--------

Open Playlist Menu	Playlist menu opened	Playlist menu opened	Passed
Click New Playlist Button	Playlist name input space provided	Playlist name input space provided	Passed
Input name and cancel	Playlist Header closed	Playlist Header closed	Passed
Click New Playlist Button	Playlist name input space provided	Playlist name input space provided	Passed
Input name and confirm	Playlist Header closed	Playlist Header closed	Passed

Limitations: Does not actually create the meta data.

### **Prints ‘invalid playlist’ upon empty input**

Test	Expectation	Output	Result
Open Playlist menu	Playlist menu opened	Playlist menu opened	Passed
Create New Playlist	Playlist name input space provided	Playlist name input space provided	Passed
Confirm w/ no input	‘Invalid Playlist’ text on playlist header	‘Invalid Playlist’ text on playlist header	Passed

### **Rejects white-space only as playlist name**

Test	Expectation	Output	Result
Open Playlist menu	Playlist menu opened	Playlist menu opened	Passed
Create New Playlist	Playlist name input space provided	Playlist name input space provided	Passed
Confirm w/ white space only	‘Invalid Playlist’ text on playlist header	‘Invalid Playlist’ text on playlist header	Passed

### **Set timer (ui test)**

Test	Expectation	Output	Result
Click Timer button	Opens timer menu	Opens timer menu	Passed
Select 30 minutes	Time left is set to 30 minutes	Time left is set to 30 minutes	Passed

### **Reset timer (ui test)**

Test	Expectation	Output	Result
Hovers of Time-Left	Reset Timer button shows up	Reset Timer button shows up	Passed

Click on Reset Timer Button	Time left is reset to 00:00	Time left is reset to 00:00	Passed
-----------------------------	-----------------------------	-----------------------------	--------

#### **Modify shuffle state (click test)**

Test	Expectation	Output	Result
Confirm shuffle state button click-able	Click method returns no error	No error output	Passed

#### **Modify repeat state (click test)**

Test	Expectation	Output	Result
Confirm repeat state button click-able	Click method returns no error	No error output	Passed

#### **Modify mute state (click test)**

Test	Expectation	Output	Result
Click Volume button to mute (1)	Song is muted	Song is muted	Passed
Click Volume button to mute (2)	Song is unmuted	Song is unmuted	Passed

#### **Main music controls (click test)**

Test	Expectation	Output	Result
Click play/pause button (1)	Song output stops	Song output stopped	Passed
Click play/pause button (2)	Song output resumes	Song out resumed	Passed
Confirm skip button (Left) click-able	Click method returns no error	No error output	Passed
Confirm skip button (Right) click-able	Click method returns no error	No error output	Passed

#### **Lyrics on album click**

Test	Expectation	Output	Result
Click on any music	Music Selected	Music Selected	Passed
Click on Album image	Lyrics displayed	Lyrics displayed	Passed
Click on Lyrics	Album image displayed	Album image displayed	Passed

### Search music in playlist

Test	Expectation	Output	Result
Input text in search bar	No results found-returned	No results found-returned	Passed

### Remove music from current playlist

Test	Expectation	Output	Result
Open Playlist menu	Playlist menu opened	Playlist menu opened	Passed
Select a playlist and enter	Returned to base screen	Returned to base screen	Passed
Click on remover button	Delete music button is displayed	Delete music button is displayed	Passed
Confirm delete music button click-able	Click method returns no error	No error output	Passed

### Toggle current playlist button

Test	Expectation	Output	Result
Open Playlist menu	Playlist menu opened	Playlist menu opened	Passed
Select a playlist and enter	Returned to base screen	Returned to base screen	Passed
Confirm current playlist button – inactive and clickable	Click method returns no error	No error output	Passed

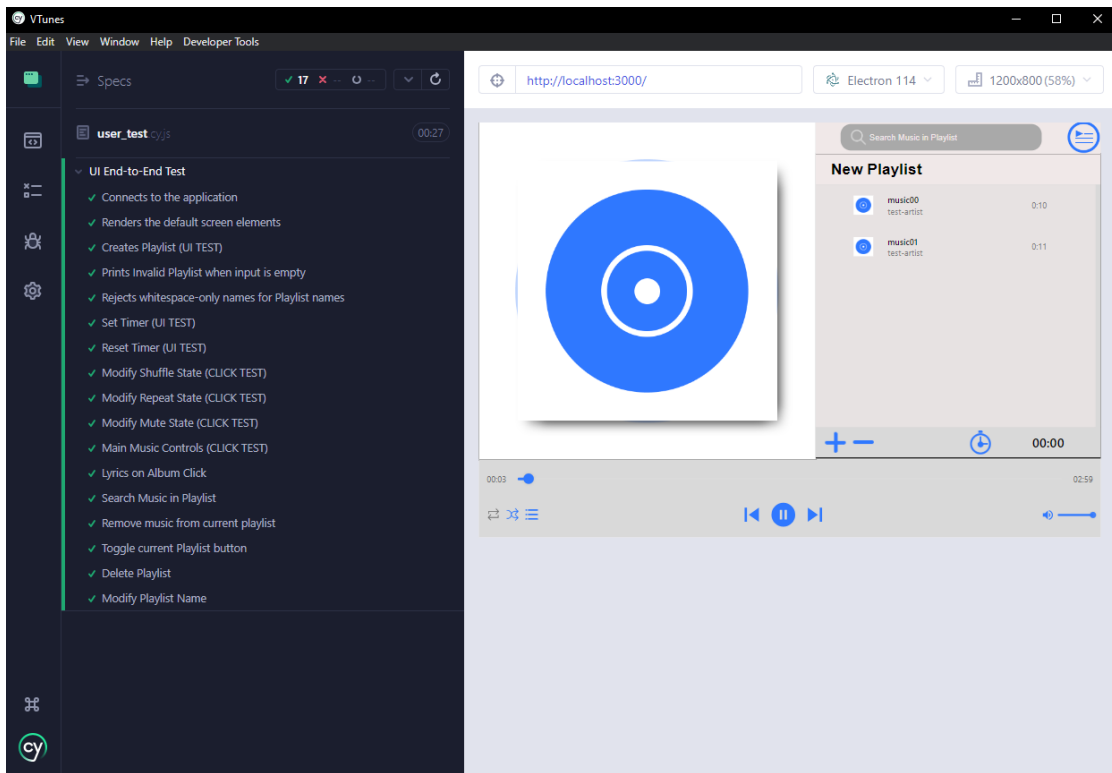
### Delete playlist

Test	Expectation	Output	Result
Open Playlist menu	Playlist menu opened	Playlist menu opened	Passed
Click Delete Playlist button	Remover button displayed	Remover button displayed	Passed
Confirm remover button click-able	Click method returns no error	No error output	Passed

### Modify playlist name

Test	Expectation	Output	Result
Open Playlist menu	Playlist menu opened	Playlist menu opened	Passed
Select a playlist and enter	Returned to base screen	Returned to base screen	Passed

Attempt text input on Playlist Name	Text box exist + input possible	Text box exist + input possible	Passed
-------------------------------------	---------------------------------	---------------------------------	--------



Cypress 결과화면을 통해서 test 가 성공적으로 이뤄졌다는 것을 확인할 수 있다.

## 5 PERFORMANCE TESTING

### 5.1 TEST RISKS / ISSUES

본 테스트는 VTunes Application 의 성능을 측정하는 테스트이다. 이는 프로그램이 실행되는 컴퓨터의 사양이나 실행 중인 프로세스 등의 실행 환경에 따라 다른 결과를 낼 수 있다는 한계가 있다. 따라서 하나의 컴퓨터에서만 성능 측정을 수행하며, VTunes 이외에 프로세스가 실행되는 것을 최소화하여 작업한다.

### 5.2 TEST APPROACH(S)

본 테스트는 기능이 수행되고 UI 가 렌더링되는 시점까지의 시간을 측정하고 메모리 사용량을 측정하는 테스트이다. VTunes Application 은 Electron Framework 로 동작하기 때문에, 앱을 실행하게 되면 내장된 웹 브라우저가 같이 실행된다. 기능의 수행 시간을 측정하기 위해 브라우저에 탑재되어 있는 개발자 도구 중 Performance Tab 의 Record 기능을 사용한다. Record 기능은 앱을 사용하는 화면을 녹화하면서

스크립트가 동작하고 UI 화면이 그려지는 시간을 구분하여 frame 마다 런타임의 성능을 측정한다. 메모리의 사용량은 작업관리자를 통해서 확인한다.

### 5.3 TEST PASS / FAIL CRITERIA

다음과 같은 조건을 만족해야 Test 를 Pass 했다고 판단한다.

1. 메모리 사용을 300mb 이내로 한다.
2. 모든 동작은 동작 실행 시작 후 1 초내로 작동해야 한다.

### 5.4 TEST ENVIRONMENTAL / STAFFING / TRAINING NEEDS

- Environmental:

CPU: 13<sup>th</sup> Gen Intel(R) Core(TM) i5-1340P 1.90 GHz

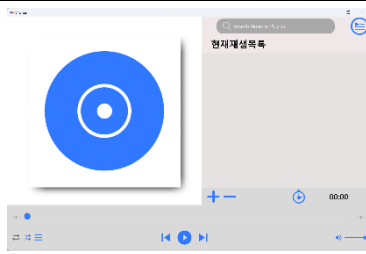
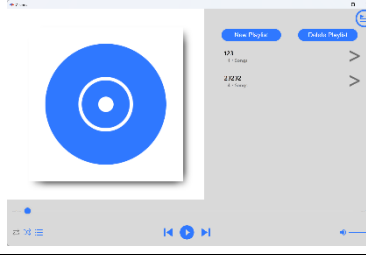
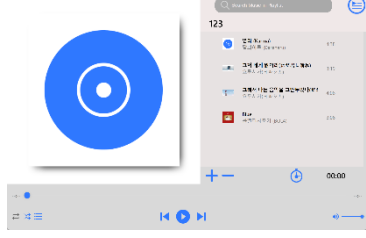
RAM: 16GB LPDDR5-5200 SDRAM

OS: Windows 11 Education 22H2

STORAGE: 256 GB PCIe 4.0 M.2 SSD

### 5.5 TEST RESULT

#### 수행 시간 측정

Test_case	action	Performance	actual	result
로딩		2 ms Loading 79 ms Scripting 4 ms Rendering 2 ms Painting 46 ms System	132ms	pass
플레이리스트 목록 조회		44 ms Scripting 17 ms Rendering 9 ms Painting 57 ms System	130ms	pass
플레이리스트 음원 조회		3 ms Loading 74 ms Scripting 79 ms Rendering 9 ms Painting 54 ms System	219ms	pass

최초 음원 재생		7 ms Loading 111 ms Scripting 28 ms Rendering 13 ms Painting 86 ms System	245ms	pass
이전 음원 재생		2 ms Loading 66 ms Scripting 46 ms Rendering 32 ms Painting 92 ms System	238ms	pass
다음 음원 재생		4 ms Loading 57 ms Scripting 31 ms Rendering 18 ms Painting 52 ms System	161ms	pass
음원 셔플		12 ms Loading 64 ms Scripting 18 ms Rendering 9 ms Painting 62 ms System	165ms	pass
음원 검색		89 ms Scripting 35 ms Rendering 22 ms Painting 99 ms System	246ms	pass
가사 출력		28 ms Scripting 49 ms Rendering 7 ms Painting 50 ms System	134ms	pass

## 메모리 사용량 확인

이름	상태	17% CPU	39% 메모리	1% 디스크	0% 네트워크
▼ VTunes(5)		1.3%	203.5MB	0.1MB/s	0Mbps
VTunes		0%	86.3MB	0MB/s	0Mbps
VTunes		0%	60.1MB	0MB/s	0Mbps
VTunes		0.8%	46.6MB	0.1MB/s	0Mbps
VTunes		0%	5.6MB	0MB/s	0Mbps
VTunes		0.4%	5.0MB	0MB/s	0Mbps



## **6 USABILITY TESTING**

### **6.1 TEST METHOD**

- 1) Usability Testing 를 위한 테스터를 모집한다.
- 2) 해당 테스터에게 VTunes 에 대한 간략한 설명과 함께 화면 녹화에 대해 동의를 받는다.
- 3) 자유롭게 VTunes 를 사용할 수 있는 환경에서 VTunes 가 설치되어 있는 노트북을 통해 VTunes 를 사용할 수 있도록 한다.
- 4) 테스터의 테스트가 끝난 이후, 화면녹화를 돌려 보며 테스터가 어떻게 VTunes 를 사용했는지 확인한다.

### **6.2 TEST RISKS / ISSUES**

음원 파일의 형식이 mp3, wav, ogg 형식이 아닌 파일을 열고자 하면 해당 파일들은 VTunes 로 실행이 불가능하다.

### **6.3 TEST APPROACH(S)**

최소한의 정보만이 주어진 상황에서 미션을 주었을 때, 해당 목적을 달성하기 위한 방법에 대해서는 테스터가 자율적으로 찾을 수 있도록 시간을 부여한다. 다만 해당 목적을 달성하기에 어려움을 테스터가 느꼈을 경우에 한해 사용자 매뉴얼을 제공한다.

### **6.4 TEST PASS / FAIL CRITERIA**

Test Result 에 있는 체크리스트를 바탕으로 실행여부를 검사했다.

### **6.5 TEST ENVIRONMENTAL / STAFFING / TRAINING NEEDS**

테스터가 자유롭게 VTunes 를 사용해볼 수 있는 환경을 제공한다. 따라서 방에 VTunes 이 설치파일이 제공되어 있는 노트북만 있고 테스터는 해당 방에 들어가서 노트북을 통해 VTunes 를 사용한다.

해당 과정을 개발자가 확인하기 위하여 테스터의 개인정보 수집 동의 과정을 거친 후 화면 녹화를 실시한다.

### **6.6 TEST RESULT**

**블랙박스 테스트 목록**

**제공용:**

목표	설명
<b>Music Manipulation</b>	음원 추가하거나 삭제하기
	음원의 순서를 수정하기
<b>Playlist Manipulation</b>	플레이리스트를 추가하거나 삭제하기
	플레이리스트 제목 수정하기
<b>Timer Manipulation</b>	타이머 실행하거나 종료하기
<b>Lyrics Manipulation</b>	가사를 표시하거나 표시 끄기

## 개발자들 체크리스트 용:

### Music Manipulation

Play / Pause	음원을 재생하고 일시정지 한다.	✓
Mute Music	음원 재생 시에 음소거를 실행한다.	✓
Volume Control	음원 재생 중에 음량을 조절한다.	✓
Repeat	음원 반복 재생 방식을 바꿔본다.	✓
Shuffle	음원 재생 순서를 바꿔본다.	✓
Show Current Playlist	현재재생목록을 확인한다.	✓
Progress Bar	재생 중인 음원의 진행도를 바꿔본다.	✓
Lyrics Jump	표시된 가사에 원하는 위치를 클릭하여 음원의 재생 위치를 변경한다.	✓

### Playlist Manipulation

Enter Playlist Menu	플레이리스트 메뉴창에 진입한다.	✓
Create Playlist	플레이리스트를 생성한다.	✓
Delete Playlist	플레이리스트를 삭제한다.	✓
Rename Playlist	플레이리스트의 제목을 바꿔본다.	✓

Switch Playlist	재생중인 플레이리스트를 바꿔본다.	✓
Search Music	음원을 검색한다.	✓

### Timer Manipulation

Set Timer	타이머를 실행한다.	✓
Reset Timer	실행한 타이머를 종료한다.	✓

### Lyrics Manipulation

Show Lyrics	가사를 표시될 수 있도록 앨범 창을 클릭한다.	✓
Close Lyrics	표시된 가사를 없애기 위해 앨범창을 클릭한다.	✓