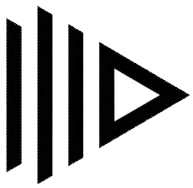
3. Design

Open Media Player



Open Media Player

Student No.	22313530
Name	BAE WON IL
E-Mail	baewonil@yu.ac.kr

[Revision history]

Revision date	Version #	Description	Author
05/16/2025	1.00	First Draft	BAE WON IL
06/05/2025	1.01	다이어그램 수정	BAE WON IL

= Contents =

1. Introduction .....

2. Class diagram .....

3. Sequence diagram .....

4. State machine diagram .....

5. Implementation requirements .....

6. Glossary .....

7. References .....

1. Introduction

1. Summary

현대의 멀티미디어는 다양한 형태로 제공되고 있고
이러한 멀티미디어를 사용함에 있어
디지털 취약계층과 청각장애우의 제한된 미디어 접근성을 개선하고자 한다.

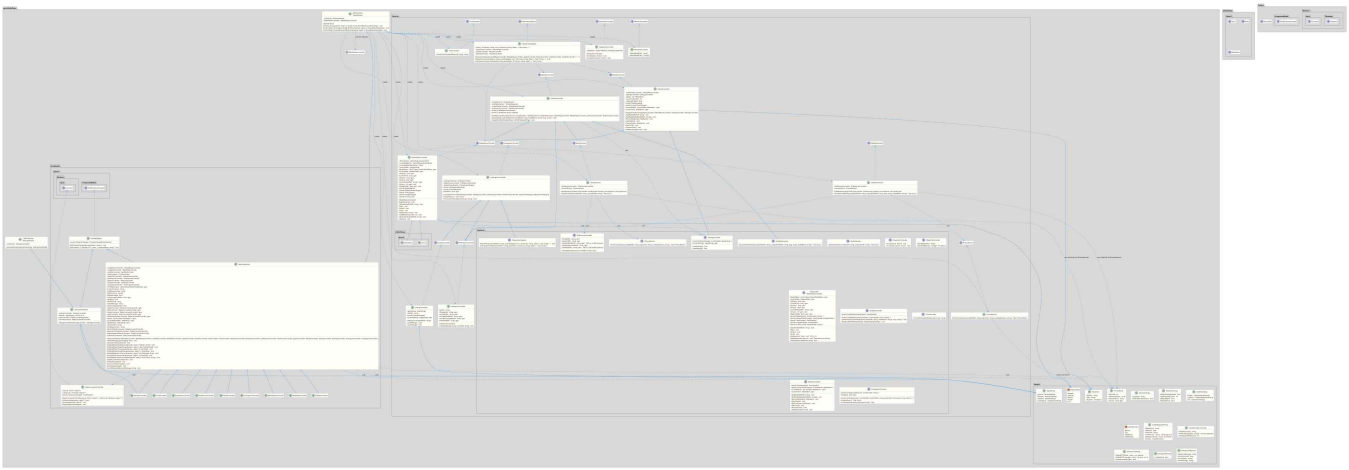
2. Introduce

이번에 제작하게 된 Project, Open Media Player는
단순히 재생만 하는것이 아닌, STT를 이용한 영상의 자막생성과
LM을 사용한 Helper Chat-bot과 동작 자동화를 수행함으로써
청각장애우의 제한된 사용경험을 보완하고
디지털 취약계층의 접근성을 개선을 할 수 있을것으로 기대된다.

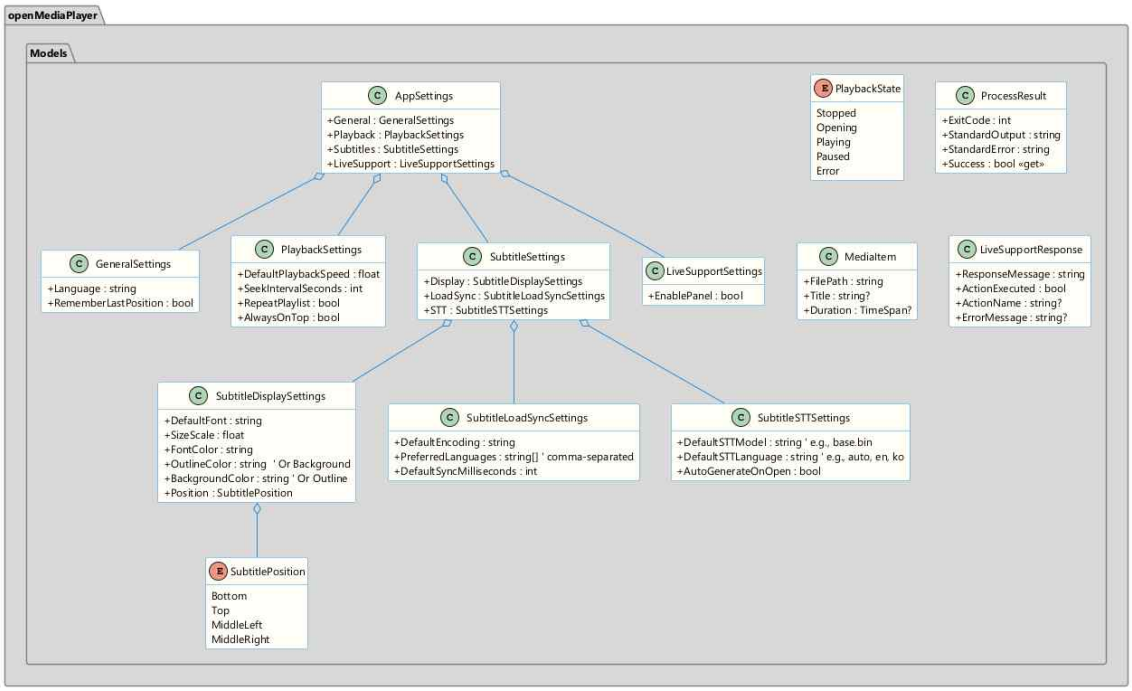
3. Goal

본 Analysis 문서에서 사용자가 어떤식으로 상호작용 하는지 분석하고
각 기능을 분석하여 명확하게 정의할 것이며.
이 작업을 수행함으로써, Open Media Player 프로젝트가 어떤식으로 구성되고
동작하게 되는지 상세하게 알 수 있을 것으로 기대된다.

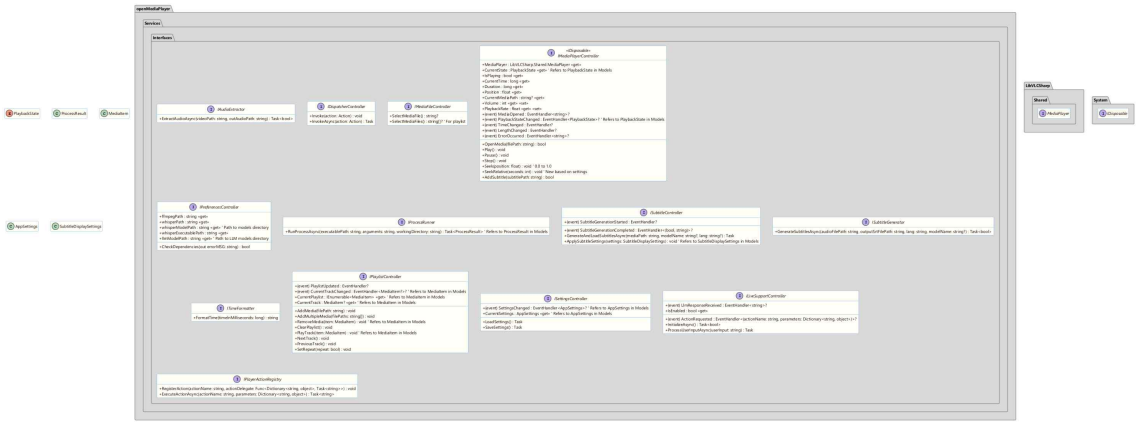
2. Class diagram



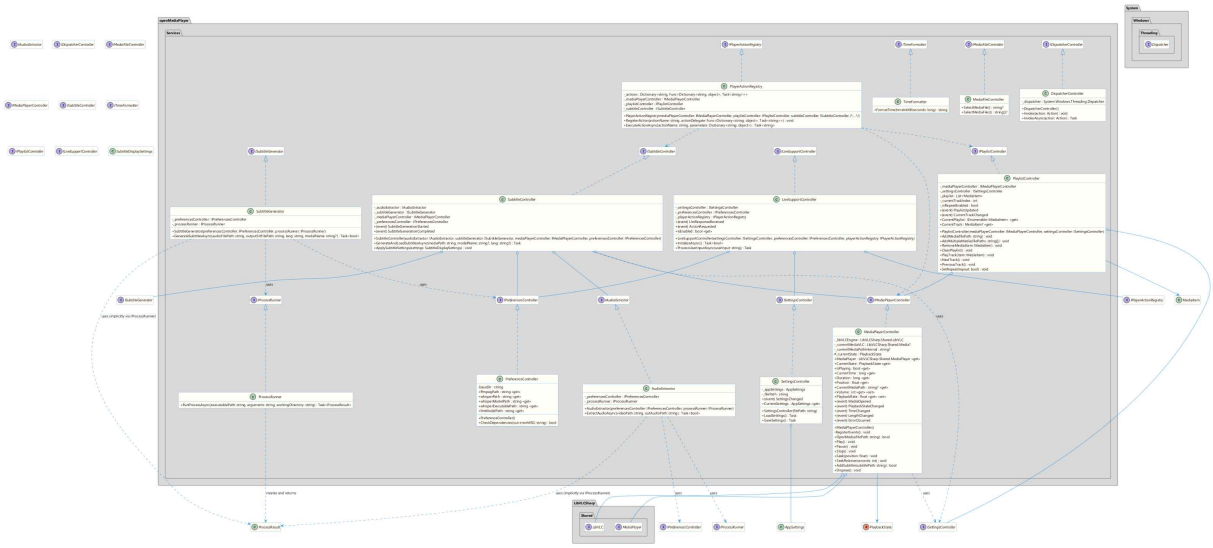
StarUML로 그렸을때 가독성이 좋지 못하여
PlantUML로 동일하게 구성하였다.
문서 내에 이미지 첨부로는 가독성이 좋지 못하여 별도로 전체 이미지를 첨부 후 git에도 동일한 class Diagram이미지를 업로드 하였다.


openMediaPlayer
Models
AppSettings
+General : GeneralSettings
+Playback : PlaybackSettings
+Subtitles : SubtitleSettings
+LiveSupport : LiveSupportSettings
PlaybackState
Stopped
Opening
Playing
Paused
Error
ProcessResult
+ExitCode : int
+StandardOutput : string
+StandardError : string
+Success : bool «get»
GeneralSettings
+Language : string
+RememberLastPosition : bool
PlaybackSettings
+DefaultPlaybackSpeed : float
+SeekIntervalSeconds : int
+RepeatPlaylist : bool
+AlwaysOnTop : bool
SubtitleSettings
+Display : SubtitleDisplaySettings
+LoadSync : SubtitleLoadSyncSettings
+STT : SubtitleSTTSettings
LiveSupportSettings
+EnablePanel : bool
MediaItem
+FilePath : string
+Title : string?
+Duration : TimeSpan?
LiveSupportResponse
+ResponseMessage : string
+ActionExecuted : bool
+ActionName : string?
+ErrorMessage : string?
SubtitleDisplaySettings
+DefaultFont : string
+SizeScale : float
+FontColor : string
+OutlineColor : string ' Or Background
+BackgroundColor : string ' Or Outline
+Position : SubtitlePosition
SubtitleLoadSyncSettings
+DefaultEncoding : string
+PreferredLanguages : string[] ' comma-separated
+DefaultSyncMilliseconds : int
SubtitleSTTSettings
+DefaultSTTModel : string ' e.g., base.bin
+DefaultSTTLanguage : string ' e.g., auto, en, ko
+AutoGenerateOnOpen : bool
SubtitlePosition
Bottom
Top
MiddleLeft
MiddleRight

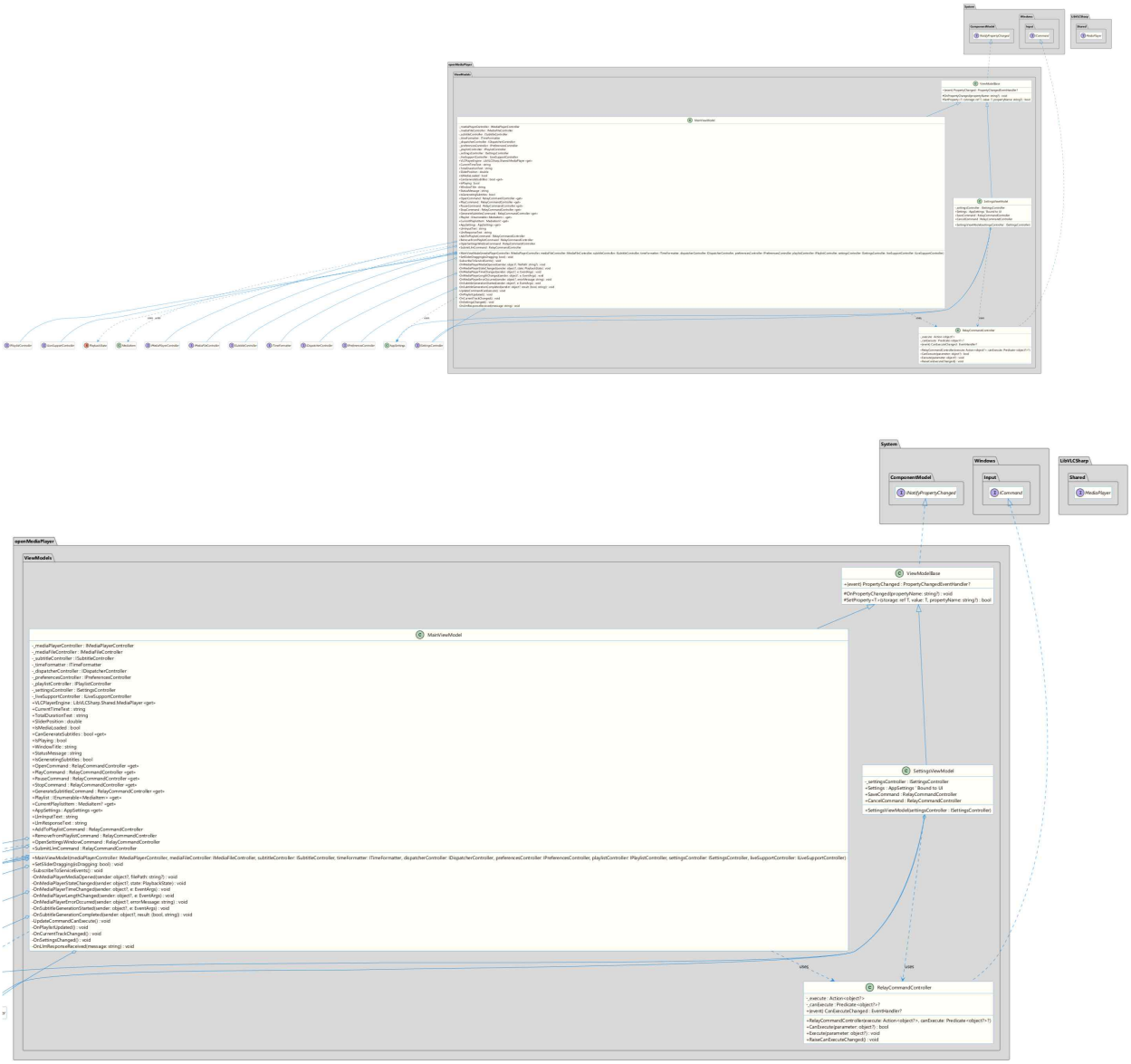
Models Part



Service Interface Part

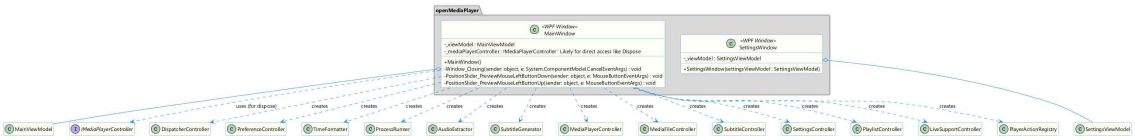


Services Part



(Package부분 확대)

View Model Part



Main App Part

패키지: openMediaPlayer.Models
애플리케이션 전체에서 사용되는 데이터 구조 포함.

- PlaybackState (enum): 미디어 재생 상태.
  - Stopped: 재생 중지.
  - Opening: 미디어 불러오는 중.
  - Playing: 미디어 재생 중.
  - Paused: 미디어 재생 일시 중지.
  - Error: 재생 중 오류 발생.

- ProcessResult (class): 외부 프로세스 실행 결과를 저장.
  - ExitCode (int): 프로세스가 반환한 종료 코드.
  - StandardOutput (string): 프로세스의 표준 출력 스트림에서 나온 텍스트 출력.
  - StandardError (string): 프로세스의 표준 오류 스트림에서 나온 텍스트 출력.
  - Success (bool, get-only): 프로세스가 성공적으로 실행되었는지 여부를 나타낸다.

    (Typically ExitCode is 0).

- MediaItem (class): 재생 목록의 단일 항목을 나타낸다.
  - FilePath (string): 미디어 파일의 전체 경로.
  - Title (string?): 현재 로드된 미디어의 제목.
  - Duration (TimeSpan?): 미디어의 길이.

- AppSettings (class): 설정 범주.
  - General (GeneralSettings): 일반 설정.
  - Playback (PlaybackSettings): 미디어 재생 설정.
  - Subtitles (SubtitleSettings): 자막 설정.
  - LiveSupport (LiveSupportSettings): LLM 기능 관련 설정.

- GeneralSettings (class): 일반 설정 저장.
  - Language (string): UI에 사용될 선택된 언어.
  - RememberLastPosition (bool): 미디어 파일의 마지막 재생 위치를 기억할지 여부.

- PlaybackSettings (class): 재생 설정.

- DefaultPlaybackSpeed (float): 재생 기본 속도.
- SeekIntervalSeconds (int): 탐색 시 앞뒤로 이동할 시간(초).
- RepeatPlaylist (bool): 마지막 항목 재생 후 재생 목록을 반복할지 여부.
- AlwaysOnTop (bool): 미디어 플레이어 창을 항상 다른 창 위에 표시할지 여부.

- SubtitleSettings (class): 자막 설정.
  - Display (SubtitleDisplaySettings): 자막 표시 속성.
  - LoadSync (SubtitleLoadSyncSettings): 자막 로딩 및 동기화 설정.
  - STT (SubtitleSTTSettings): 음성-텍스트 변환(자막 생성) 설정.

- SubtitlePosition (enum): 자막위치.
  - Bottom: 화면 하단.
  - Top: 화면 상단.
  - MiddleLeft: 수직 중앙, 왼쪽.
  - MiddleRight: 수직 중앙, 오른쪽.

- SubtitleDisplaySettings (class): 자막 표시 설정
  - DefaultFont (string): 자막의 기본 글꼴 모음.
  - SizeScale (float): 자막 크기의 배율.
  - FontColor (string): 자막 텍스트의 색상.
  - OutlineColor (string): 자막 윤곽선 색상.
  - BackgroundColor (string): 자막 배경색.
  - Position (SubtitlePosition): 자막의 기본 화면 위치.

- SubtitleLoadSyncSettings (class): 자막 파일 로드 동기화 관리
  - DefaultEncoding (string): 자막 파일의 기본 텍스트 인코딩.
  - PreferredLanguages (string[]): 선호하는 자막 언어 코드 배열.
  - DefaultSyncMilliseconds (int): 자막 타이밍을 조정하기 위한 기본 오프셋.

- SubtitleSTTSettings (class): STT 설정.
  - DefaultSTTModel (string): 사용할 기본 STT 모델 파일.
  - DefaultSTTLanguage (string): STT를 위한 기본 언어 또는 자동 감지를 위한 "auto".
  - AutoGenerateOnOpen (bool): 미디어 파일이 열릴 때 자동으로 자막 생성을 시작할지 여부.

- LiveSupportSettings (class): 실시간 지원 (LLM 연동) 기능설정.
  - EnablePanel (bool): 실시간 지원 패널/기능이 활성화되었는지 여부.

- LiveSupportResponse (class): LLM 응답 관련.
  - ResponseMessage (string): 사용자에게 표시할 메시지.

- ActionExecuted (bool): 입력을 기반으로 작업이 성공적으로 실행되었는지 여부.
- ActionName (string?): 실행된 작업의 이름.
- ErrorMessage (string?): 작업이 실패했거나 문제가 발생한 경우의 오류 메시지.

패키지: openMediaPlayer.Services.Interfaces
애플리케이션 인터페이스 정의.

- IAudioExtractor (interface): 비디오 파일에서 오디오를 추출.
  - ExtractAudioAsync(videoPath: string, outAudioPath: string): 비동기적으로 videoPath에서 오디오를 추출하여 outAudioPath에 저장. 성공 시 true를 반환.

- IDispatcherController (interface): UI 스레드에서 작업을 실행.
  - Invoke(action: Action):UI 스레드에서 동기적으로 실행.
  - InvokeAsync(action: Action): UI 스레드에서 비동기적으로 실행.

- IMediaFileController (interface): 미디어 파일 선택.
  - SelectMediaFile(): 단일 미디어 파일을 선택할 수 있는 대화 상자를 연다. 파일 경로 또는 null을 반환.
  - SelectMediaFiles(): 여러 미디어 파일을 선택할 수 있는 대화 상자를 연다. 파일 경로 배열 또는 null을 반환.

- IMediaPlayerController (interface): 미디어 재생 기능, IDisposable 구현.
  - MediaPlayer (LibVLCSharp.Shared.MediaPlayer, get-only): LibVLC 인스턴스에 접근 제공.
  - CurrentState (PlaybackState, get-only): 현재 재생 상태.
  - IsPlaying (bool, get-only): 미디어가 현재 재생 중이면 true.
  - CurrentTime (long, get-only): 현재 재생 시간.
  - Duration (long, get-only): 현재 미디어의 총 길이.
  - Position (float, get-only): 현재 재생 위치 표시.
  - CurrentMediaPath (string?, get-only): 현재 로드된 미디어 파일의 경로.
  - Volume (int, get/set): 재생 볼륨.
  - PlaybackRate (float, get/set): 재생 속도.
  - MediaOpened (event): 새 미디어 파일이 성공적으로 열림. 파일 경로 전달.
  - PlaybackStateChanged (event): PlaybackState가 변경될 때 발생, 새 상태를 전달.
  - TimeChanged (event): 재생 시간이 변경될 때 주기적으로 발생.
  - LengthChanged (event): 미디어의 길이 변경될 때 발생.
  - ErrorOccurred (event): 재생 오류 발생, 오류 메시지 전달.

- OpenMedia(filePath: string): 미디어 파일을 열고 준비, 성공 시 true.
- Play(): 재생을 시작하거나 다시 시작.
- Pause(): 재생을 일시 중지.
- Stop(): 재생을 중지.
- Seek(position: float): 미디어의 특정 위치로 탐색.
- SeekRelative(seconds: int): 지정된 시간만큼 앞이나 뒤로 탐색.
- AddSubtitle(subtitlePath: string): 외부 자막 파일을 추가. 성공 시 true 반환.

- IPreferencesController (interface): 외부 종속성에 대한 경로 접근 및 사용 가능 여부 확인.
  - ffmpegPath (string, get-only): FFmpeg 실행 파일 경로.
  - whisperPath (string, get-only): Whisper (STT) 관련 디렉토리/실행 파일 경로.
  - whisperModelPath (string, get-only): Whisper STT 모델을 포함하는 디렉토리 경로.
  - whisperExecutablePath (string, get-only): 메인 Whisper 실행 파일 경로.
  - llmModelPath (string, get-only): LLM 모델을 포함하는 디렉토리 경로.
  - CheckDependencies(out errorMSG: string):
    필요 종속성 확인, 정상 true, 아니면 false를 반환하고 errorMSG에 오류 메시지를 전달.

- IProcessRunner (interface): 외부 쉘 프로세스 실행.
  - RunProcessAsync(executablePath: string, arguments: string, workingDirectory: string):
    비동기적으로 주어진 인수와 디렉토리로 실행. ProcessResult 반환.

- ISubtitleController (interface): 자막 생성 및 표시 설정 관리.
  - SubtitleGenerationStarted (event): 자막 생성이 시작될 때 발생.
  - SubtitleGenerationCompleted (event):
    자막 생성이 완료될 때 발생.
    튜플 (bool success, string messageOrPath) 전달.
  - GenerateAndLoadSubtitlesAsync(mediaPath: string, modelName: string?, lang: string?):
    비동기적 자막을 생성후 로드 시도.
  - ApplySubtitleSettings(settings: SubtitleDisplaySettings): 표시 설정을 현재 자막에 적용.

- ISubtitleGenerator (interface): 오디오로부터 자막 파일 생성.
  - GenerateSubtitlesAsync(audioFilePath: string, outputSrtFilePath: string, lang: string,
    modelName: string?):
    오디오 파일에서 비동기적으로 SRT 자막 파일을 생성. 성공 시 true를 반환.

- ITimeFormatter (interface): 시간 값문자열로 형식화.

▶ FormatTime(timeInMilliseconds: long): 밀리초 단위의 시간 값을 HH:MM:SS로 변환.

■ IPlaylistController (interface): 미디어 항목 재생 목록을 관리.

- ▶ PlaylistUpdated (event): 재생 목록이 변경될 때(항목 추가, 제거, 순서 변경) 발생.
- ▶ CurrentTrackChanged (event): 재생 목록에서 현재 재생 중인 트랙이 변경될 때 발생.

- ▶ CurrentPlaylist (IEnumerable <MediaItem>, get-only): 현재 미디어 항목 목록.
- ▶ CurrentTrack (MediaItem?, get-only): 재생 목록에서 현재 활성화된 미디어 항목.
- ▶ AddMedia(filePath: string): 단일 미디어 파일을 재생 목록에 추가.
- ▶ AddMultipleMedia(filePaths: string[]): 여러 미디어 파일을 재생 목록에 추가.
- ▶ RemoveMedia(item: MediaItem): 특정 미디어 항목을 재생 목록에서 제거.
- ▶ ClearPlaylist(): 재생 목록에서 모든 항목을 제거.
- ▶ PlayTrack(item: MediaItem): 재생 목록에서 특정 미디어 항목 재생을 시작.
- ▶ NextTrack(): 재생 목록의 다음 트랙을 재생.
- ▶ PreviousTrack(): 재생 목록의 이전 트랙을 재생.
- ▶ SetRepeat(repeat: bool): 재생 목록 반복을 활성화하거나 비활성화.

■ ISettingsController (interface): ⚙ 애플리케이션 설정을 로드하고 저장하는 서비스를 정의.

- ▶ SettingsChanged (event): 애플리케이션 설정이 수정되었을 때 발생. 새 AppSettings를 전달.
- ▶ CurrentSettings (AppSettings, get-only): 현재 로드된 애플리케이션 설정.
- ▶ LoadSettings(): 영구 저장소에서 비동기적으로 설정을 로드.
- ▶ SaveSettings(): 현재 설정을 영구 저장소에 비동기적으로 저장.

■ ILiveSupportController (interface): SLM 상호작용 서비스.

- ▶ LlmResponseReceived (event): LM으로부터 응답을 받았을 때 발생.
  response 전달.
- ▶ ActionRequested (event): LM이 수행할 작업을 제안할 때 발생.
  작업 이름과 매개변수를 전달.
- ▶ IsEnabled (bool, get-only): LM 지원이 현재 활성화되어 있고 초기화되었는지 여부 확인.
- ▶ InitializeAsync(): LM 서비스를 비동기 초기화. 성공 시 true.
- ▶ ProcessUserInputAsync(userInput: string): 사용자 입력을 LM으로 보내 처리를 요청.

■ IPlayerActionRegistry (interface):
LM이나 애플리케이션의 다른 부분에서 트리거될 수 있는 작업을 등록, 실행하는 서비스.

- ▶ RegisterAction(actionName: string, actionDelegate: Func<Dictionary<string, object>, Task<string>>):
  매개변수를 받고 결과 메시지를 반환하는 델리게이트와 함께 명명된 작업을 등록.
- ▶ ExecuteActionAsync(actionName: string, parameters: Dictionary<string, object>):
  이름으로 등록된 작업을 비동기적으로 실행하고 필요한 매개변수를 전달. 결과 메시지를 반환.

패키지: openMediaPlayer.Services
openMediaPlayer.Services.Interfaces에 정의된 인터페이스 구현 포함.

- AudioExtractor (class): IAudioExtractor를 구현. FFmpeg를 사용하여 오디오를 추출.
  - ▶ IPreferencesController(FFmpeg 경로 가져오기) 및 IProcessRunner(FFmpeg 실행)에 의존.
    IProcessRunner를 통해 암묵적으로 ProcessResult를 사용.
- DispatcherController (class): IDispatcherController를 구현.
  System.Windows.Threading.Dispatcher를 사용하여 UI 스레드 접근을 관리.
- MediaFileController (class): IMediaFileController를 구현.
  시스템 대화 상자를 사용하여 사용자가 미디어 파일을 선택.
- MediaPlayerController (class): IMediaPlayerController를 구현.
  LibVLCSharp.Shared를 사용, 핵심 미디어 플레이어 로직.
- PreferenceController (class): IPreferencesController를 구현.
  외부 도구(FFmpeg, Whisper) 및 모델 디렉토리에 대한 경로를 관리.
- ProcessRunner (class): IProcessRunner를 구현.
  외부 프로세스를 비동기적으로 실행하고 출력을 캡처.
- SubtitleGenerator (class): ISubtitleGenerator를 구현.
  외부 STT 도구(Whisper)를 사용하여 오디오에서 자막 생성.
- TimeFormatter (class): ITimeFormatter를 구현.
  시간 값을 문자열로 Formatting (12:34:56).
- PlaylistController (class): IPlaylistController를 구현.
  MediaItem 객체 목록을 관리, 재생 순서, 반복 기능 및 재생 목록 지속성 제어.
- SettingsController (class): ISettingsController를 구현.
  애플리케이션 설정을 로드하고 저장하는 작업 처리.
- LiveSupportController (class): ILiveSupportController를 구현.
- PlayerActionRegistry (class): IPlayerActionRegistry를 구현.
  매개변수와 함께 이름으로 호출할 수 있는 사용 가능한 작업의 사전을 유지 관리.

패키지: openMediaPlayer.ViewModels
MVVM(Model-View-ViewModel) 패턴의 일부인 ViewModel을 포함.
ViewModel은 View를 위한 데이터를 준비하고 UI 로직을 처리.

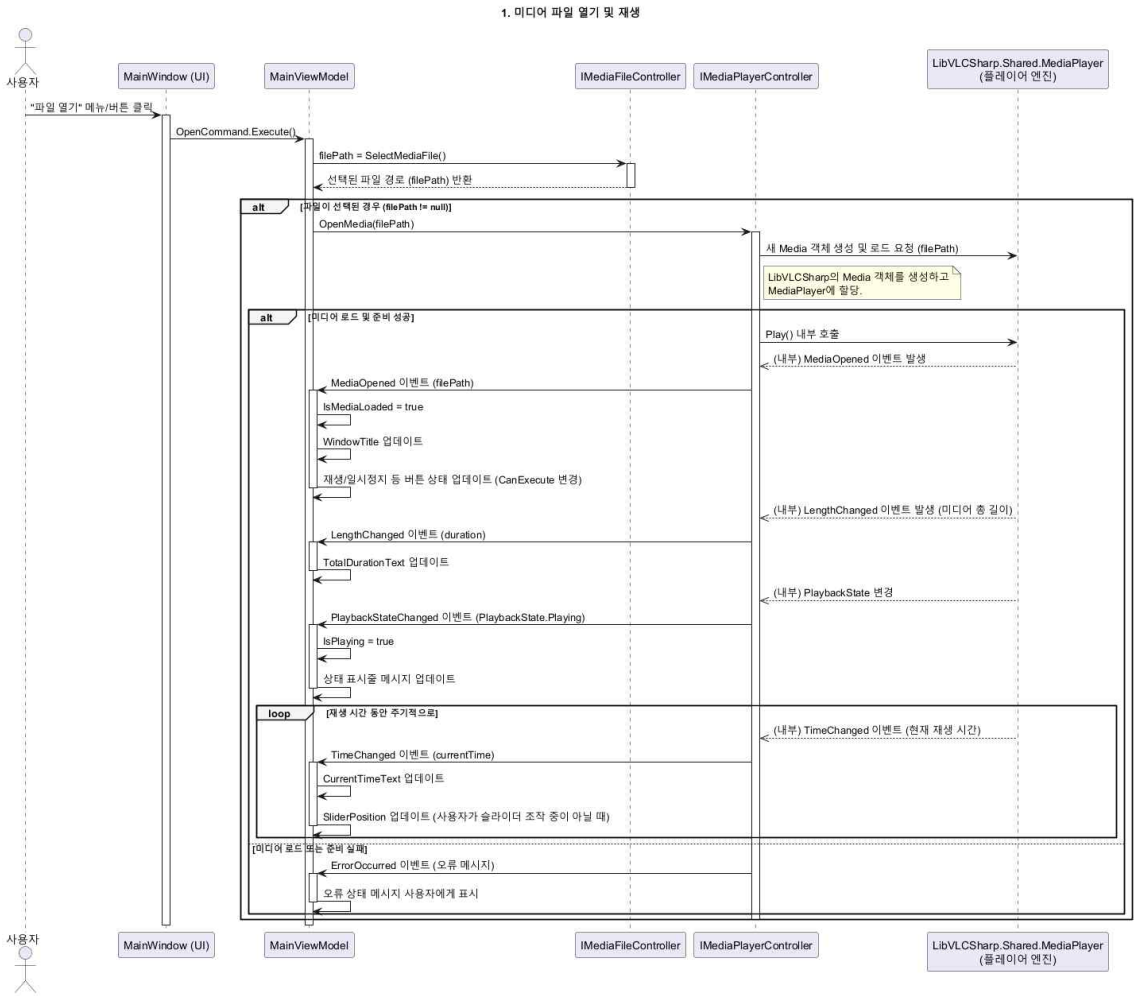
- ViewModelBase (class): System.ComponentModel.INotifyPropertyChanged를 구현.
  모든 ViewModel의 기본 클래스로, WPF와 같은 UI 프레임워크에서 데이터 바인딩에 필수적인 속성 변경 알림을 제공.
  - PropertyChanged (event): INotifyPropertyChanged의 표준 이벤트.
  - OnPropertyChanged(propertyName: string?):
    PropertyChanged 이벤트를 발생시키는 헬퍼 메소드.
  - SetProperty<T>(storage: ref T, value: T, propertyName: string?):
    속성의 지원 필드 설정 값이 변경된 경우에만 PropertyChanged를 발생시키는 헬퍼 메소드.
- RelayCommandController (class): System.Windows.Input.ICommand를 구현.
  UI 작업 ViewModel의 메소드에 바인딩 MVVM 클래스.
  - _execute (Action<object?>): 명령이 호출될 때 실행할 델리게이트.
  - _canExecute (Predicate<object?>?):
    현재 명령을 실행할 수 있는지 여부를 결정하는 델리게이트.
  - RaiseCanExecuteChanged(): CanExecute의 재평가를 수동으로 트리거하는 메소드.
- MainViewModel (class): ViewModelBase를 확장.
  메인 애플리케이션 창의 ViewModel, View(UI)와 다양한 서비스 간의 상호 작용을 조율.
  - SetSliderDragging(isDragging: bool): 사용자가 위치 슬라이더를 드래그할 때 원치 않는 업데이트를 방지하기 위해 상태를 관리.
- SettingsViewModel (class): ViewModelBase를 확장. 설정 창의 ViewModel.

패키지: openMediaPlayer

애플리케이션의 메인 진입점과 UI 창 정의(MVVM View)를 포함.

- MainWindow (class): 메인 애플리케이션 창(WPF Window)을 나타냄.
- SettingsWindow (class): 설정 창(WPF Window)을 나타냄.

3. Sequence diagram



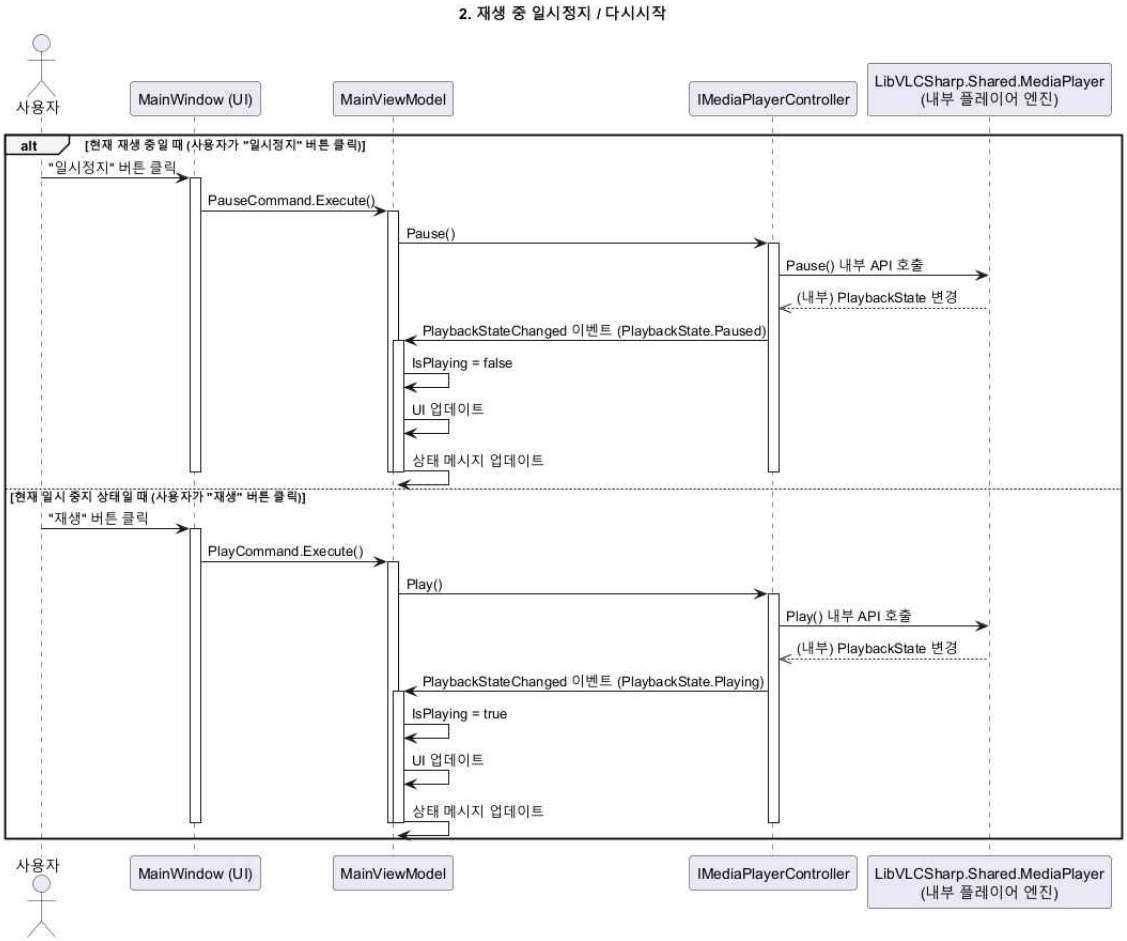
사용자가 파일 열기를 선택하면 MainViewModel의 OpenCommand로 실행 지령이 전달되고, MainViewModel은 IMediaFileController를 통해서 사용자로부터 재생 대상 파일 경로를 얻고, 이 경로를 사용해 IMediaPlayerController에게 파일을 열 것을 지시한다.

IMediaPlayerController는 LibVLCSharp 엔진을 사용해 파일을 로드->재생준비->재생 미디어가 성공적으로 열리면 MediaOpened, LengthChanged, PlaybackStateChanged 이벤트가 발생하고 MainViewModel로 전달된다.

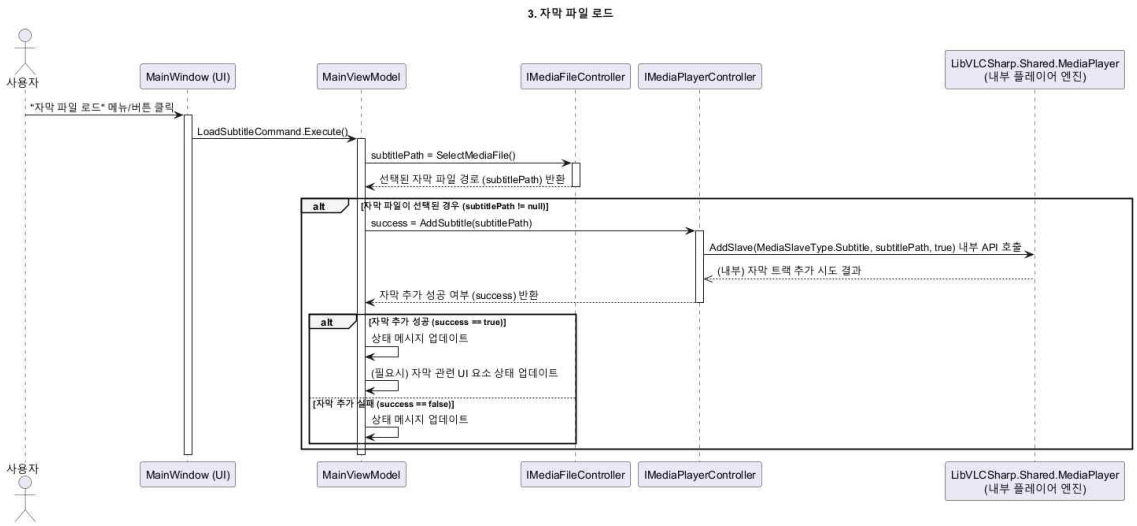
재생 시간은 TimeChanged이벤트로 주기적으로 MainViewModel에 통지한다.

MainViewModel은 통지된 이벤트로부터 UI에 해당사항을 갱신하여 보여준다.

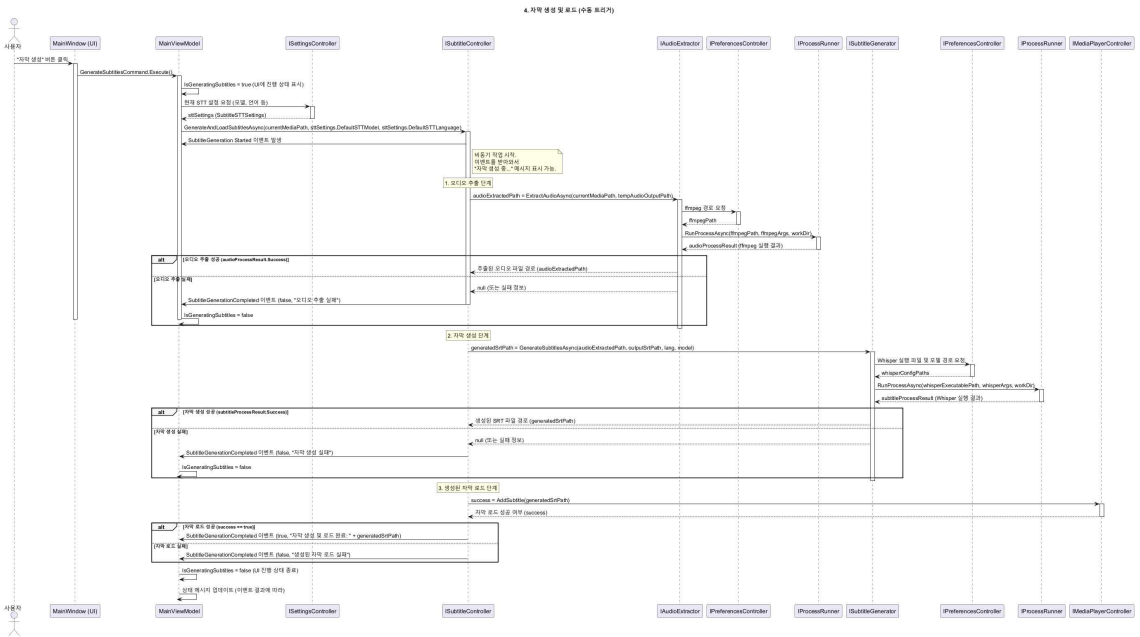
문제가 발생하면 ErrorOccured 이벤트가 발생해 오류상황을 알린다.



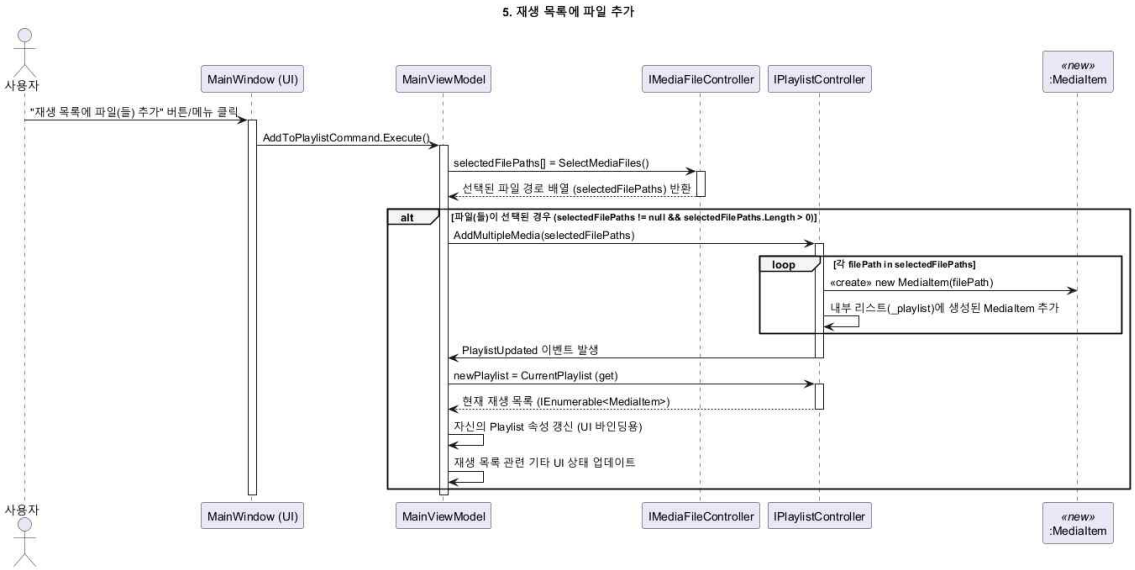
사용자가 재생이나 일시정지 버튼을 클릭하면,
MainWindow는 이 입력을 확인하여 MainViewModel에 정의된 PauseCommand 또는 PlayCommand를 실행시킨다.
MainViewModel은 미디어의 재생 상태에 따라 IMediaPlayerController에게 Pause() 또는 Play() 메소드를 호출하여 실제 동작을 요청.
IMediaPlayerController는 요청을 받아 LibVLCSharp 엔진을 제어하여 미디어 재생을 제어한다.
플레이어의 내부 재생 상태가 변경되면 IMediaPlayerController는 PlaybackStateChanged 이벤트를 발생시켜 MainViewModel에게 통지한다.
MainViewModel은 이벤트를 통해 현재 재생 상태를 갱신하고, UI를 갱신하여 사용자에게 현재 상태를 표시한다.



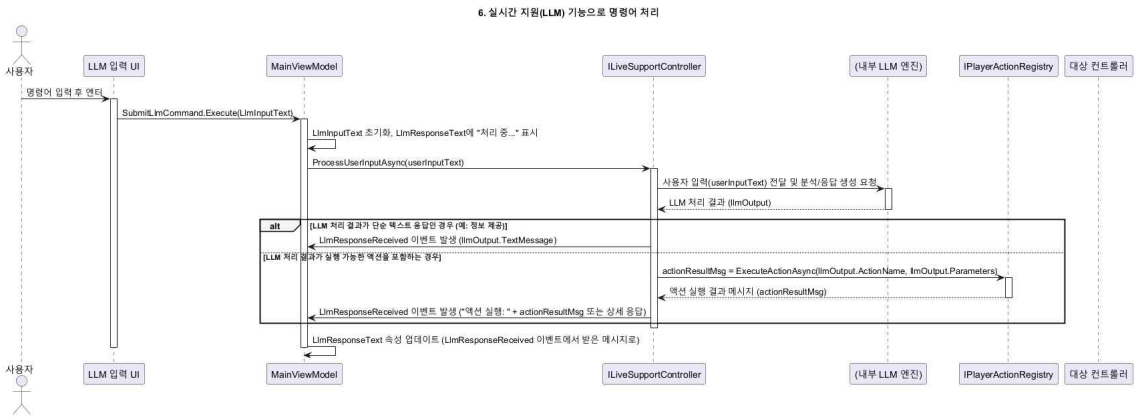
사용자가 자막 파일 로드 기능을 실행하면,
MainWindow -> MainViewModel -> LoadSubtitleCommand로 이어진다.
MainViewModel은 IMediaFileController를 호출해 로드할 자막 파일의 경로를 얻어오고 얻어온 경로로 MainViewModel은 IMediaPlayerController의 AddSubtitle 메소드를 호출 현재 재생 중인 미디어에 해당 자막을 추가하도록 요청한다.
IMediaPlayerController는 내부 LibVLCSharp 엔진의 기능을 이용, 자막 파일을 로드하고 미디어에 연결.
자막 추가 성공 여부는 AddSubtitle 메소드의 반환값(bool)을 통해 MainViewModel에 전달.
MainViewModel은 사용자에게 상태 메시지를 UI에 갱신.



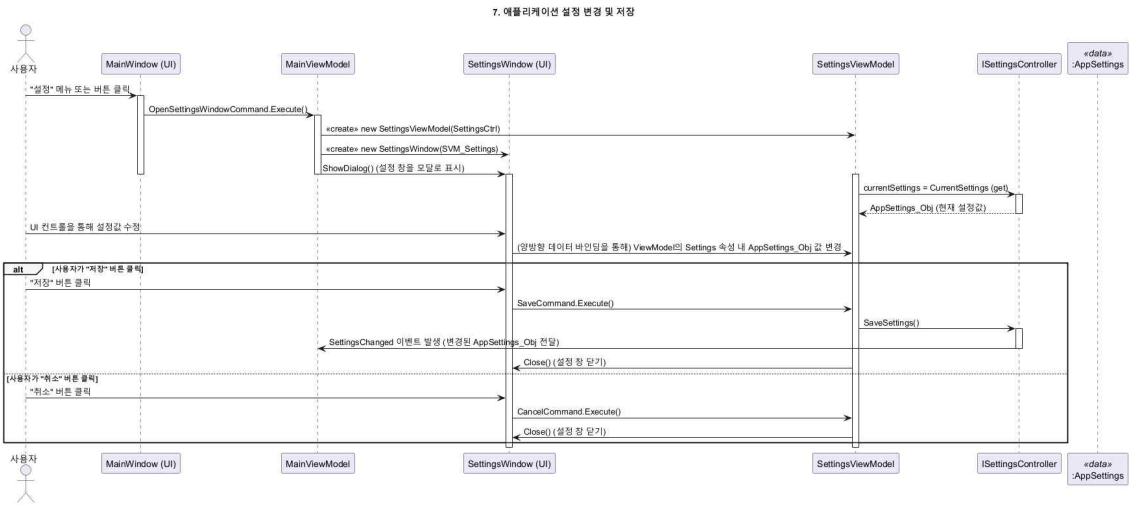
사용자가 자막 생성 기능을 실행,
MainViewModel은 ISettingsController 통해 STT(Speech-to-Text) 관련 설정 확인 후, ISubtitleController -> GenerateAndLoadSubtitlesAsync 메소드 호출 자막 생성 및 로드 프로세스 진행.
ISubtitleController는 SubtitleGenerationStarted 이벤트를 발생, IAudioExtractor를 호출 현재 미디어 파일에서 오디오를 추출.
IAudioExtractor는 IPreferencesController에서 외부 도구의 경로 정보를 얻고, IProcessRunner를 통해 오디오 파일을 생성.
오디오 추출이 완료되면, ISubtitleController는 ISubtitleGenerator에 추출된 오디오 파일과 STT 설정 전달해, 자막 파일 생성을 요청.
ISubtitleGenerator도 IPreferencesController에서 IProcessRunner를 통해 STT 엔진 실행 자막 파일 생성 완료하면, ISubtitleController는 IMediaPlayerController의 AddSubtitle 메소드를 호출하여 자막 파일을 현재 미디어에 로드.
성공 실패 여부, 최종 결과 메시지나 파일 경로는 SubtitleGenerationCompleted 이벤트를 통해 MainViewModel에 통지, UI가 업데이트되고 사용자에게 통지.
각 주요 단계 실패가 발생하면, 해당 시점에서 프로세스가 중단되고 실패 상태가 MainViewModel로 전달.



사용자가 재생 목록에 파일 추가 기능을 실행,
MainWindow는 MainViewModel의 AddToPlaylistCommand로 전달.
MainViewModel은 IMediaFileController를 사용, 사용자로부터 재생 목록에 추가할 미디어 파일 경로를 얻어옴.
선택된 파일 경로들을 가지고 MainViewModel은 IPlaylistController의 AddMultipleMedia 메소드를 호출.
PlaylistController는 전달받은 각 파일 경로에 대해 신규 MediaItem 객체를 생성 객체들을 내부적으로 관리하는 재생 목록에 추가.
모든 MediaItem이 추가된 후, IPlaylistController는 PlaylistUpdated 이벤트를 발생 재생 목록에 변경 통지.
MainViewModel은 PlaylistUpdated 이벤트를 수신하면, IPlaylistController의 CurrentPlaylist 속성을 통해 재생 목록 정보를 가져와 UI에 바인딩된 Playlist 속성을 갱신.

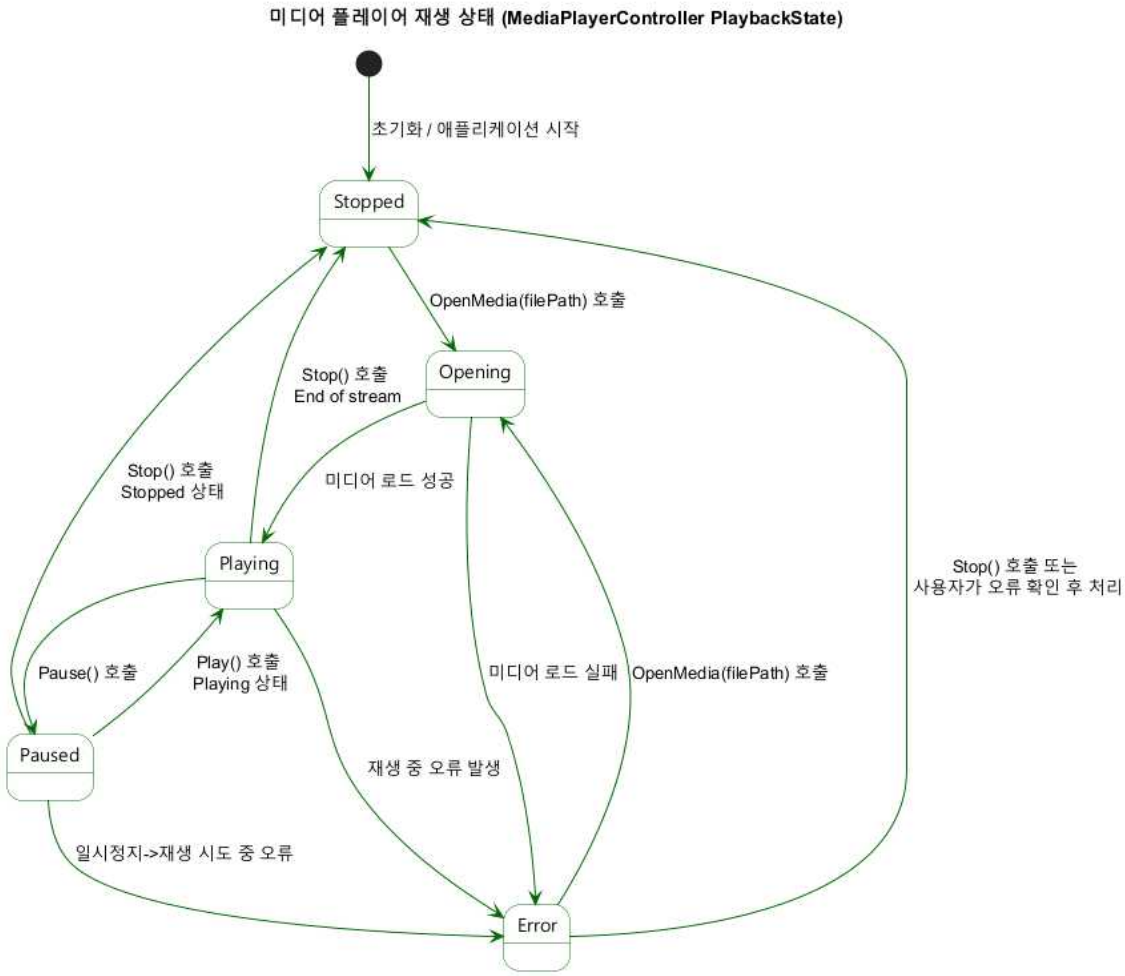


사용자가 LLM 입력 UI에 자연어로 된 명령을 입력하고 전송하면,
MainViewModel으로 전달.
MainViewModel은 ILiveSupportController의 ProcessUserInputAsync 메소드 호출
사용자 입력을 처리 요청.
ILiveSupportController는 LLM 엔진에 분석 요청,
LLM 엔진은 입력을 해석하여 단순 텍스트 응답을 생성하거나,
미리 정의된 특정 액션(actionName)과 그에 필요한 파라미터(parameters)를 식별하여 반환.
ILiveSupportController가 결과를 받으면, 단순 텍스트 응답이면
LlmResponseReceived 이벤트를 통해 MainViewModel에 통지해 갱신.
실행 가능한 액션을 포함, ILiveSupportController는 IPlayerActionRegistry의
ExecuteActionAsync 메소드를 호출하여 해당 액션의 실행을 요청.
IPlayerActionRegistry는 요청된 액션 이름에 해당하는 미리 등록된 함수(델리게이트)를 찾아 실행,
대상 컨트롤러의 메서드를 호출하여 실제 기능을 수행.
액션 실행 후, 결과 메시지는 다시 IPlayerActionRegistry를 거쳐
ILiveSupportController에게 전달, LlmResponseReceived 이벤트를 통해
MainViewModel에게 통지해 UI에 갱신.



사용자가 설정 기능을 실행,
MainViewModel은 SettingsViewModel과 SettingsWindow의 새 인스턴스를 생성
설정 창을 사용자에게 보여준다.
ISettingsController의 인스턴스가 SettingsViewModel에 주입되어 설정 데이터 처리를 담당하게 된다.
SettingsViewModel은 ISettingsController의 CurrentSettings 속성을 통해 현재 애플리케이션 설정을 가져오고, SettingsWindow의 UI 컨트롤들은 설정값들에 바인딩 되어 화면에 현재 상태를 표시.
설정값들을 변경하면, 양방향 데이터 바인딩을 통해 SettingsViewModel이 관리하는 AppSettings 객체에 즉시 반영.
저장 버튼을 클릭하면, SettingsViewModel의 SaveCommand가 실행되어 ISettingsController의 SaveSettings() 메서드가 호출된다.
ISettingsController는 SettingsViewModel으로부터 전달받은 AppSettings 객체의 현재 상태를 저장.
저장 완료되면, ISettingsController는 SettingsChanged 이벤트를 발생시켜 애플리케이션의 다른 부분들 설정이 변경되었음을 통지.
취소 버튼을 클릭하면, 변경된 내용은 저장되지 않고 설정 창만 닫힘.

4. State machine diagram



처음 애플리케이션이 시작되면 Stopped 상태로 시작한다.
미디어 파일을 열려고 하면 플레이어는 Opening 상태로 진입한다.
파일이 정상적으로 열리면 미디어를 로드하고, 실패하면 Error 상태로 전환하여 오류 발생을 통지한다.
Stopped 상태에서 미디어가 로드되어 있으면 Play 지령 시 바로 Opening을 거쳐 재생할 수 있다.

Opening상태에서 미디어가 로드되면
플레이어는 Playing상태로 전환되어 미디어를 실제로 재생한다.
Opening상태에서 문제가 생기면 Error상태로 진입한다.
또는 Opening도중에 Stop 지령을 내리면 Stopped상태로 돌아간다.

Playing 상태에서 Pause 지령을 내리면 Paused상태로 전환된다.
Stop지령을 내리면 Stopped상태로 전환되고, 미디어 끝에 도달해도 Stopped상태로 전환된다.

재생 중 오류가 발생하면 Error상태로 전환된다.
재생 중 새 미디어를 여는 경우, 현재 재생중인 미디어는 Stopped를 거쳐 정지 후 새 미디어를 불러와 Opening상태로 전환된다.

Paused 상태에서는 다시 Play 지령을 내리면 Playing 상태로 돌아가
이전에 멈춘 지점부터 재생을 이어나간다.
이 상태에서 Stop 지령을 받으면 Stopped 상태로 전환된다.
오류가 발생하는 경우 Error 상태로 전환되며
새 미디어를 여는 경우에는 Stopped를 거쳐 정지 후 새 미디어를 불러와 Opening 상태로 전환된다.

Error 상태에서는 대부분 기능이 제한될 수 있다.
다시 미디어를 불러와 정상적으로 열리면 Opening 상태로 전환 되며
실패하면 Error 상태로 돌아온다.
Stop 지령으로 오류 상태 해제 후 Stopped 상태로 돌아갈 수 있다.

5. Implementation requirements
    A. H/W Platform Requirements - Minimal
        1. CPU : 4 Core and 3Ghz Higher
        2. RAM : 4GB
        3. Storage : 10GB Free Space
    B. H/W Platform Requirements – Recommended
        1. CPU : 6 Core and 3Ghz Higher
        2. RAM : 8GB
        3. Storage : 10GB Free Space
        4. GPU : RX480 or GTX 1060 (6GB or More VRAM)
        5. Notes : GPU is Optional but recommended for best performance
    C. S/W Platform Requirements
        1. 64bit Windows 10
        2. Implementation Language : C#

6. Glossary

API	응용 프로그램이 운영체제 기능이나 다른 서비스의 기능을 사용할 수 있도록 미리 정의된 명령어, 함수, 프로토콜 등의 인터페이스 규격.
Chatbot (챗봇)	사용자와의 대화(텍스트 또는 음성)를 통해 정보를 제공하거나 특정 작업을 자동화하는 프로그램.
Playlist	여러 미디어 파일을 원하는 순서대로 모아놓은 목록
Codec	멀티미디어 데이터(오디오, 비디오)를 압축(Encoding)하거나 압축 해제(Decoding)하는 소프트웨어 또는 하드웨어 알고리즘입니다. (예: H.264, AAC, VP9).
Decoder	코덱을 사용하여 압축된 오디오 또는 비디오 데이터를 플레이어가 처리할 수 있는 원시(Raw) 데이터 형태로 변환하는 구성 요소입니다.
FFMPEG[1)]	오디오 및 비디오 파일의 기록, 변환, 스트리밍을 위한 매우 광범위한 라이브러리와 도구를 제공하는 오픈 소스 프로젝트.
LM / SLM[2)3)]	인간의 언어(자연어)를 통계적으로 모델링하여 이해하거나 생성할 수 있는 인공지능 모델. SLM은 비교적 적은 파라미터 수를 가진 경량화된 언어 모델을 의미.
Renderer	디코딩된 오디오 샘플이나 비디오 프레임 데이터를 실제 출력 장치(스피커, 화면)를 통해 사용자가 인지할 수 있도록 출력하는 구성 요소.
STT[4)]	사람의 음성 언어를 컴퓨터가 읽을 수 있는 텍스트 데이터로 변환하는 기술입니다.
Whisper[4]	OpenAI에서 개발하여 공개한 오픈 소스 음성 인식(STT) 모델입니다. 높은 정확도와 다국어 지원 능력을 특징으로 합니다.

MVVM	Model-View-ViewModel의 약어로 UI개발을 위해 비즈니스 로직(Back-End)과 프레젠테이션 로직(Front-End)을 분리하는 디자인 패턴
WPF[5]	Windows Presentation Foundation의 약어로 Windows based UI Framework로써 해상도 독립적 벡터 기반 렌더링을 지원한다.
C#	MS에서 개발한 객체지향 프로그래밍 언어 Java에서 JVM과 같이 .NET Runtime위에 실행된다. 크로스 플랫폼을 지원하며, 오픈소스로 개발되고 있다.
.NET[6]	MS에서 개발한 무료, 오픈소스, 크로스플랫폼 개발 Framework.

7. References

7-1. Open-AI Whisper(paper) : https://arxiv.org/abs/2212.04356

7-2. Open-AI Whisper(git) : https://github.com/openai/whisper

7-3. FFMPEG : https://www.ffmpeg.org/documentation.html

7-4. MS DirectX API: https://learn.microsoft.com/en-us/windows/win32/directx

7-5. MS Win32 API : https://learn.microsoft.com/en-us/windows/win32/apiindex/api-index-portal

7-6. MS .NET API : https://learn.microsoft.com/en-us/dotnet/api/

7-7. WPF .NET: https://learn.microsoft.com/en-us/dotnet/desktop/wpf/overview/?view=netdesktop-9.0

7-8. LLM-Powered GUI Agents in Phone Automation: Surveying Progress and Prospects https://arxiv.org/pdf/2504.19838

7-9. B-MOCA:BENCHMARKING MOBILE DEVICE CONTROL AGENTS ACROSS DIVERSE CONFIGURATIONS https://arxiv.org/pdf/2404.16660

---

1) FFMPEG : https://www.ffmpeg.org/documentation.html

2) LLM-Powered GUI Agents in Phone Automation: Surveying Progress and Prospects https://arxiv.org/pdf/2504.19838

3) B-MOCA:BENCHMARKING MOBILE DEVICE CONTROL AGENTS ACROSS DIVERSE CONFIGURATIONS https://arxiv.org/pdf/2404.16660

4) Open-AI Whisper(paper) : https://arxiv.org/abs/2212.04356

5) WPF .NET: https://learn.microsoft.com/en-us/dotnet/desktop/wpf/overview/?view=netdesktop-9.0

6) MS .NET API : https://learn.microsoft.com/en-us/dotnet/api/