

사진첩 정리 및 검색 프로그램 IC
- 3. Design -

[Revision history]

Revision date	Version #	Description	Author
	1.0.0	초안	
	1.0.1	클래스 다이어그램 내용 추가	
	1.0.2	다이어그램 수정	

[Contents]

1. Introduction	1
2. Class Diagram	2
3. Sequence Diagram	27
4. State machine diagram	36
5. Implementation requirements	37
6. Glossary	37
7. References	37

1. Introduction

본 문서는 Analysis 단계의 문서에 이어지는 Design 단계의 문서다.

1.1. Summary

최근 소셜 미디어의 트렌드는 사진을 공유하는 것이다. 이에 따라, 소셜 미디어를 활발히 이용하는 사용자들은 사진 촬영을 자주하고, 결과적으로 사진첩에 수백 장의 사진이 쌓이게 된다. 이렇게 쌓인 사진들을 효율적으로 관리하고 원하는 사진을 쉽게 찾기 위해 사용자들은 사진 정리에 상당한 시간을 할애하고 있다.

때문에 본 문서에서는 자동으로 사진첩을 정리해 주는 프로그램을 제안한다. 시스템이 자동으로 사진첩을 정리해 줌으로써 사용자는 훨씬 더 편리하게 사진을 관리할 수 있을 것으로 기대한다.

1.2 Important Points of Design

- 사용자 경험 기반의 설계
- 효과적인 정리 능력
- 빠른 검색 기능

2. Class diagram

본 프로젝트는 파이썬을 사용하여 구현한다. 따라서 클래스 다이어그램은 파이썬 문법에 따라 표현한다. 본 프로젝트에서 사용된 특정 디자인 패턴은 없으며, GUI를 의미하는 View 부분과 비즈니스 로직을 의미하는 Core 부분으로 분리하여 설계한다. 프로젝트의 외부 라이브러리 및 GUI 구현에 사용된 PyQt5의 상속은 간략하게 표현한다.

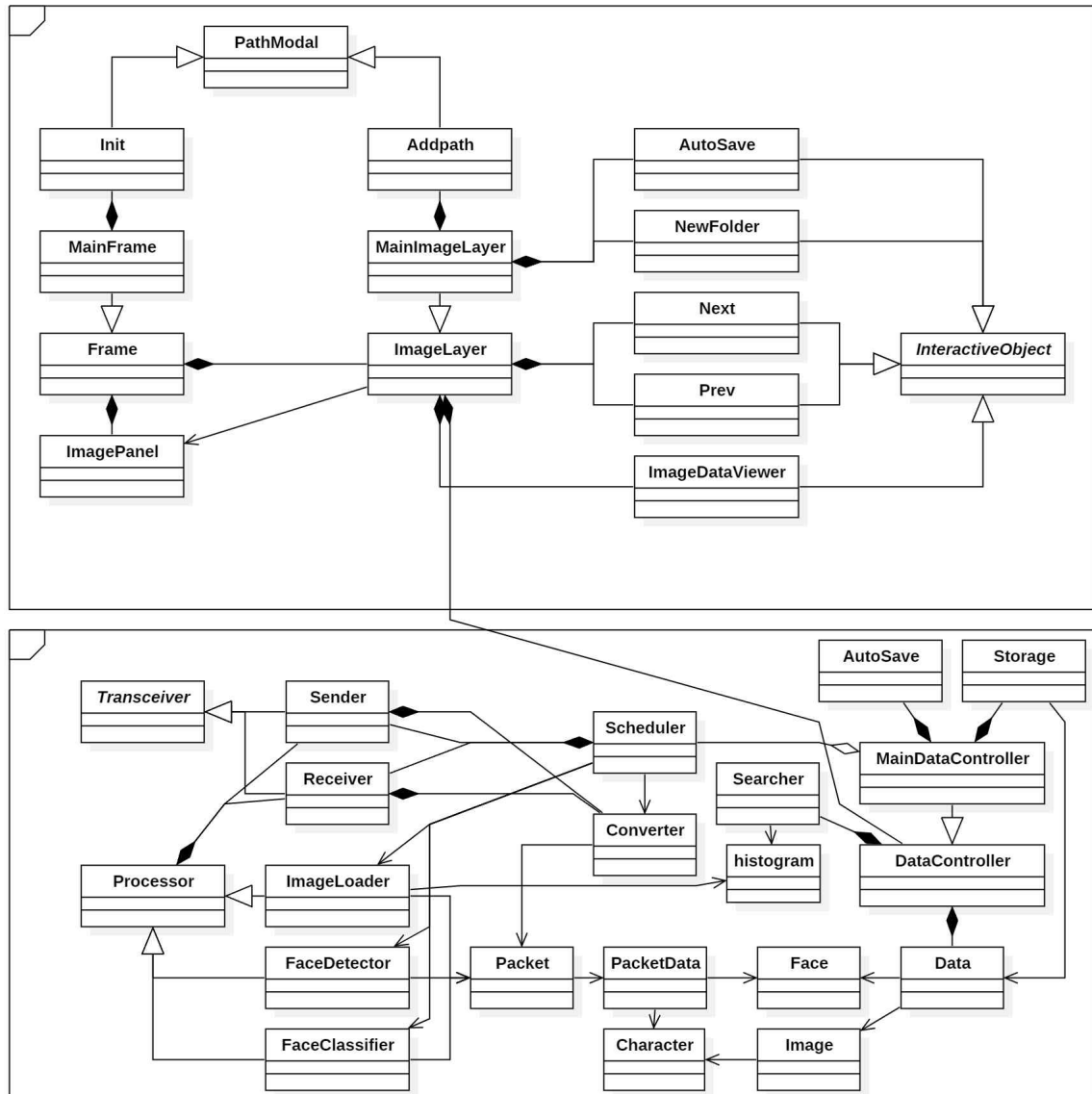


Fig. 1. Overall Simple Class Diagram

Fig. 1. 은 View 부분과 Core 부분을 분리하여 클래스 이름만을 나타내어 연관관계를 표현한 클래스 다이어그램이다(외부 라이브러리의 클래스 상속은 표현하지 않았다). View와 Core 부분의 분리를 최대한 유지하는 것을 원칙으로 하며, View에서 Core 부분의 상호작용은 DataController 클래스 또는 이를 상속받는 클래스를 통해 이루어진다.

2.1.1 View Class Diagram Description

Frame				
Class Description				
PyQt5의 QWidget을 상속받아 프로그램의 GUI의 기반이 되는 메인 프레임을 생성하는 클래스이다. 또한 검색 기능을 사용하면 새로운 윈도우가 해당 클래스를 통해서 생성된다.				
Attributes				
구분				
Name	Type	Scope	Visibility	
Description				
__TITLE	int	static	private	
프로그램 Window에 표시 될 타이틀 제목				
__x	int	instance	private	
프로그램 Window가 표시 될 때 Display에서의 위치 x값				
__y	int	instance	private	
프로그램 Window가 표시 될 때 Display에서의 위치 y값				
__width	int	instance	private	
프로그램 Window의 width 값				
__height	int	instance	private	
프로그램 Window의 height 값				
__image_panel	ImagePanel	instance	private	
현재 Frame에서 사용 할 ImagePanel 객체를 저장하는 변수				
__image_layer	ImageLayer	instance	private	
현재 Frame에서 사용 할 ImageLayer을 저장하는 변수				
Operation				
구분				
Name	Argument	Returns	Scope	Visibility
Description				
__init__	self, int, int, int, int, ImageLayer, DataController	None	instance	public
생성자 함수. 매개 변수를 통한 멤버 변수 초기화. 내부에서 init_ui 메소드를 호출하여 ui를 그림				
init_ui	self	None	instance	public
ui 초기화				
resizeEvent	self, QResizeEvent	None	instance	public
gui 크기 변경시 이벤트 처리				
show	self	None	instance	public
QWidget의 show 메소드 오버라이드 함. gui를 보여줄 때, 이미지 크기를 조정 함.				
closeEvent	self, QCloseEvent	None	instance	public
QWidget의 close 메소드를 오버라이드 하며, __image_layer의 closeEvent 메소드를 실행한다.				

MainFrame				
Class Description				
Frame을 상속받아 프로그램의 메인 Window를 구현하는 클래스				
Attributes				
구분				
Name		Type	Scope	Visibility
Description				
__X		int	static	private
프로그램 Window가 표시 될 때 Display에서의 위치 x값				
__Y		int	static	private
프로그램 Window가 표시 될 때 Display에서의 위치 y값				
__WIDTH		int	static	private
프로그램 Window의 width 값				
__HEIGHT		int	static	private
프로그램 Window의 height 값				
Operation				
구분				
Name	Argument	Returns	Scope	Visibility
Description				
__init__	self, pathlib.Path, int, int, int, int	None	instance	public
생성자 함수. 부모 생성자를 호출하여 클래스 초기화.				

ImagePanel

Class Description

pyqt5의 QGraphicsView를 상속받아 이미지를 보여주고 리사이즈, 이동 이벤트, 및 얼굴 바운더리 박스와 클릭 이벤트를 제어하는 클래스

Attributes

구분

Name	Type	Scope	Visibility
Description			
__DEFAULT_IMAGE	numpy.ndarray	static	private
이미지를 표시할 수 없을 때 표시하는 기본 이미지			
__image	numpy.ndarray	instance	private
현재 이미지 데이터			
__faces	dict<int Character>	instance	private
현재 얼굴 데이터 저장			
__face_rect	list<QGraphicsRectItem>	instance	private
얼굴 데이터를 바탕으로 바운더리 박스를 표시하는 QGraphicsRectItem를 저장			
__scale	float	instance	private
현재 확대 축소 scale 값 저장			
__image_scale	float	instance	private
현재 이미지의 scale 값 저장			
__scene	QGraphicsScene	instance	private
QGraphicsScene 인스턴스 저장			
__image_item	QGraphicsPixmapItem	instance	private
QGraphicsScene에 추가할 QGraphicsPixmapItem 인스턴스 저장			

Operation

구분

Name	Argument	Returns	Scope	Visibility
Description				
__init__	self, QWidget	None	instance	public
생성자 함수. 멤버 변수 초기화				
set_image	self, dict<int Character>, numpy.ndarray	None	instance	public
numpy의 ndarray를 받아 __image 변수에 저장				
set_face	self	None	instance	public
얼굴 데이터를 바탕으로 QGraphicsRectItem을 만들어 __face_rect에 저장 및 __scene에 등록				
draw_faces	self, float, float	None	instance	public
__faces에 담긴 얼굴 데이터를 바탕으로 __face_rect에 있는 QGraphicsRectItem의 위치 및 크기 조정				
remove_faces	self	None	instance	public
__scene에 있는 모든 얼굴 데이터 삭제				
draw_image	self	None	instance	public
__image에 저장된 이미지로 이미지를 그린다.				

resize_image	self, int, QPoint	None	instance	public
scale factor 값과 위치 정보를 담은 QPoint를 받아 scale factor 값 만큼 QPoint 위치를 기준으로 이미지 크기 변경				
resize_event	self	None	instance	public
Window의 크기에 맞게 이미지를 표시하는 전체 창의 크기 변경				
image_move	self, QPoint	None	instance	public
QPoint 위치 값 만큼 이미지 이동				
click_event	self, QPoint	None	instance	public
좌표 값을 받아 해당 좌표값에 위치하는 바운더리 박스 번호 반환				

ImageLayer				
Class Description				
PyQt5의 QWidget을 상속받아 이미지 위에 상호작용 가능한 버튼들을 표시하며 해당 이벤트를 제어하는 클래스				
Attributes				
구분				
Name	Type	Scope	Visibility	
Description				
__image_panel	ImagePanel	instance	private	
ImagePanel 저장 변수				
__image_data_viewer	ImageViewer	instance	private	
ImageViewer 저장 변수				
__prev	Prev	instance	private	
Prev 저장 변수				
__next	Next	instance	private	
Next 저장 변수				
__data_controller	DataController	instance	private	
DataController 저장 변수				
__ctrl_pressed	bool	instance	private	
ctrl 키 눌렀는지 여부 저장 변수				
__prev_pos	QPoint	instance	private	
휠 클릭을 눌렀을 때 마우스 이동시 마지막 이동 위치 저장 변수				
__wheel_pressed	bool	instance	private	
마우스 휠 눌렀는지 여부 저장 변수				
__move_event	QPoint	instance	private	
마우스 드래그 이벤트를 위한 좌표 값 저장				
__modal_is_exist	bool	instance	private	
검색 창이 띄어져 있는지 확인하는 변수				
__modal	QWidget	instance	private	
검색 창 저장하는 변수				
Operation				
구분				
Name	Argument	Returns	Scope	Visibility
Description				
__init__	self, QWidget, ImagePanel, DataController	None	instance	public
생성자 함수. 멤버 변수 초기화				
__next_event	self	None	instance	private
__next의 이벤트 제어 함수. 다음 이미지로의 변경 이벤트				
__prev_event	self	None	instance	private
__prev의 이벤트 제어 함수. 이전 이미지로의 변경 이벤트				
__file_move_event	self, QPoint	None	instance	private
파일 이동 이벤트 제어 함수. 마우스 변화량을 기반으로 특정 폴더로 파일 이동 이벤트				

__search_modal	self, DataController	None	instance	private
검색을 통해 받은 DataController로 Frame을 통해 새로운 창을 만들어서 보여 줌				
__button_rander	self, bool	None	instance	private
마우스 올라올 때 버튼들을 보여 줌				
closeEvent	self, QCloseEvent	None	instance	public
QWidget의 closeEvent를 오버라이드 함. __data_controller의 close 메소드를 실행 함				
resizeEvent	self, QResizeEvent	None	instance	public
Window의 크기가 변경 될 때, 하위 위젯 크기 변경 및 위치 조정				
keyPressEvent	self, QKeyEvent	None	instance	public
키가 클릭 되었을 때, 이벤트 처리				
keyRelease	self, QKeyEvent	None	instance	public
키가 클릭 해제 되었을 때, 이벤트 처리				
wheelEvent	self, QWheelEvent	None	instance	public
마우스 휠이 움직였을 때, 이벤트 처리				
mouseDoubleClickEvent	self, QMouseEvent	None	instance	protected
마우스 더블 클릭 할 때, 이벤트 처리				
mousePressEvent	self, QMouseEvent	None	instance	public
마우스 클릭 되었을 때 이벤트 처리				
mouseReleaseEvent	self, QMouseEvent	None	instance	public
마우스 클릭 해제 되었을 때 이벤트 처리				
mouseMoveEvent	self, QMouseEvent	None	instance	public
마우스가 움직일 때 이벤트 처리				

MainImageLayer				
Class Description				
ImageLayer을 상속받는 MainFrame에서 동작할 MainImageLayer				
Attributes				
구분				
Name	Type	Scope	Visibility	
Description				
__data_controller	MainDataController	instance	private	
MainDataController 저장 변수				
__new_folder	NewFolder	instance	private	
NewFolder 저장 변수				
__auto_save	AutoSave	instance	private	
AutoSave 저장 변수				
__add_path	AddPath	instance	private	
AddPath 저장 변수				
Operation				
구분				
Name	Argument	Returns	Scope	Visibility
Description				
__init__	self, QWidget, ImagePanel, MainDataController	None	instance	public
생성자 함수. 멤버 변수 초기화				
__new_folder_event	self	None	instance	private
__new_folder의 콜백 함수이다.. __new_folder에서 이벤트 발생 시 실행되어, AddPath의 show 메서드를 실행하여 모달 창을 보이게 함				
__add_folder_event	self, pathlib.Path	None	instance	private
AddPath의 콜백 함수이다. 프로그램에 경로를 추가하는 로직을 실행한다.				
__auto_organization	self	None	instance	private
__AutoSave의 콜백 함수이다. 사진을 정리하는 로직을 실행한다.				
resizeEvent	self, QResizeEvent	None	instance	public
Window의 리사이즈 할 때, 하위 위젯의 위치를 변경 한다.				

PathModal				
Class Description				
QDialog를 상속받아 경로를 입력받는 모달 창				
Attributes				
구분				
Name	Type	Scope	Visibility	
Description				
__WIDTH	int	static	private	
모달 창의 기본 width 값을 저장하는 상수				
__HEIGHT	int	static	private	
모달 창의 기본 height 값을 저장하는 상수				
__path_input	QLineEdit	instance	private	
경로를 입력 받는 QLineEdit을 가지고 있는 변수.				
__callback	callable	instance	private	
경로를 입력 받았을 때 실행할 콜백 메소드				
Operation				
구분				
Name	Argument	Returns	Scope	Visibility
Description				
__init__	self, callable, str, str	None	instance	public
생성자 함수. 멤버 변수 초기화				
__init_ui	self, str, str	None	instance	private
초기 ui 디자인				
__browse_event	self	None	instance	private
browse 버튼을 눌렀을 때 실행 하는 이벤트 함수. 탐색기를 동작시킨다.				
__confirm_event	self	None	instance	private
성공적으로 경로 입력을 완료 하고, 성공 버튼을 눌렀을 때 실행 되는 이벤트 함수. 생성자에서 매개 변수로 전달 받은 callback 함수를 동작 시킨다.				
show	self, QPoint	None	instance	private
디스플레이에 표시하는 함수. QPoint 정보를 받아 해당 위치를 중심으로 표시한다.				

Init				
Class Description				
PathModal을 상속받아 초기 경로 입력 모달 창 구현				
Attributes				
구분				
Name	Type	Scope	Visibility	
Description				
__SUCCESS_NAME	str	static	private	
PathModal의 성공 시 클릭하는 버튼 이름				
__CANCEL_NAME	str	static	private	
PathModal을 종료할 때 클릭하는 버튼 이름				
Operation				
구분				
Name	Argument	Returns	Scope	Visibility
Description				
__init__	self, callable	None	instance	public
생성자 함수. 부모 생성자 호출				

AddPath				
Class Description				
PathModal을 상속받아 New Folder 기능의 경로 입력 모달 창 구현				
Attributes				
구분				
Name	Type	Scope	Visibility	
Description				
__SUCCESS_NAME	str	static	private	
PathModal의 성공 시 클릭하는 버튼 이름				
__CANCEL_NAME	str	static	private	
PathModal을 종료할 때 클릭하는 버튼 이름				
Operation				
구분				
Name	Argument	Returns	Scope	Visibility
Description				
__init__	self, QWidget, callable	None	instance	public
생성자 함수. 부모 생성자 호출				

InteractiveObject				
Class Description				
상호작용하는 위젯들을 일반화 한 클래스. 위치 조정과 클릭 시 이벤트 설정을 일반화				
Attributes				
구분				
Name	Type	Scope	Visibility	
Description				
__width	int	instance	private	
위젯의 width 값 저장				
__height	int	instance	private	
위젯의 height 값 저장				
__x_ratio	float	instance	private	
위젯의 상대적 위치를 나타내는 x 값 저장				
__y_ratio	float	instance	private	
위젯의 상대적 위치를 나타내는 y 값 저장				
__axes	tuple<int>	instance	private	
위젯의 상대적 위치에서 절대적 위치 이동 값 x, y 저장				
Operation				
구분				
Name	Argument	Returns	Scope	Visibility
Description				
__init__	self, str, QWidget, callable, float, float, int, int, tuple<int>	None	instance	public
생성자 함수. 멤버 변수 초기화 및 callback 함수 등록				
resize_event	self, size:QSize	None	instance	public
QSize에 따라 자신의 위치를 조정				

AutoSave				
Class Description				
InteractiveObject를 상속받아 Auto Save 버튼을 구현한 클래스				
Attributes				
구분				
Name		Type	Scope	Visibility
Description				
__X_RATIO		float	static	private
위젯의 상대적 위치를 나타내는 x 값 저장				
__Y_RATIO		float	static	private
위젯의 상대적 위치를 나타내는 y 값 저장				
__WIDTH		int	static	private
위젯의 width 값 저장				
__HEIGHT		int	static	private
위젯의 height 값 저장				
Operation				
구분				
Name	Argument	Returns	Scope	Visibility
Description				
__init__	self, QWidget, callable, float, float, int, int, tuple	None	instance	public
생성자 함수. 부모 클래스 생성자 호출				

NewFolder				
Class Description				
InteractiveObject를 상속받아 New Folder 버튼을 구현한 클래스				
Attributes				
구분				
Name		Type	Scope	Visibility
Description				
__X_RATIO		float	static	private
위젯의 상대적 위치를 나타내는 x 값 저장				
__Y_RATIO		float	static	private
위젯의 상대적 위치를 나타내는 y 값 저장				
__WIDTH		int	static	private
위젯의 width 값 저장				
__HEIGHT		int	static	private
위젯의 height 값 저장				
Operation				
구분				
Name	Argument	Returns	Scope	Visibility
Description				
__init__	self, QWidget, callable, float, float, int, int	None	instance	public
생성자 함수. 부모 클래스 생성자 호출				

Next				
Class Description				
InteractiveObject를 상속받아 Next 버튼을 구현한 클래스				
Attributes				
구분				
Name	Type	Scope	Visibility	
Description				
__X_RATIO	float	static	private	
위젯의 상대적 위치를 나타내는 x 값 저장				
__Y_RATIO	float	static	private	
위젯의 상대적 위치를 나타내는 y 값 저장				
__WIDTH	int	static	private	
위젯의 width 값 저장				
__HEIGHT	int	static	private	
위젯의 height 값 저장				
Operation				
구분				
Name	Argument	Returns	Scope	Visibility
Description				
__init__	self, QWidget, callable, float, float, int, int	None	instance	public
생성자 함수. 부모 클래스 생성자 호출				

Prev				
Class Description				
InteractiveObject를 상속받아 Prev 버튼을 구현한 클래스				
Attributes				
구분				
Name	Type	Scope	Visibility	
Description				
__X_RATIO	float	static	private	
위젯의 상대적 위치를 나타내는 x 값 저장				
__Y_RATIO	float	static	private	
위젯의 상대적 위치를 나타내는 y 값 저장				
__WIDTH	int	static	private	
위젯의 width 값 저장				
__HEIGHT	int	static	private	
위젯의 height 값 저장				
Operation				
구분				
Name	Argument	Returns	Scope	Visibility
Description				
__init__	self, QWidget, callable, float, float, int, int	None	instance	public
생성자 함수. 부모 클래스 생성자 호출				

ImageViewer				
Class Description				
InteractiveObject를 상속받아 이미지 정보 뷰어를 구현한 클래스				
Attributes				
구분				
Name	Type	Scope	Visibility	
Description				
__X_RATIO	float	static	private	
위젯의 상대적 위치를 나타내는 x 값 저장				
__Y_RATIO	float	static	private	
위젯의 상대적 위치를 나타내는 y 값 저장				
__WIDTH	int	static	private	
위젯의 width 값 저장				
__HEIGHT	int	static	private	
위젯의 height 값 저장				
Operation				
구분				
Name	Argument	Returns	Scope	Visibility
Description				
__init__	self, QWidget, callable, float, float, int, int	None	instance	public
생성자 함수. 부모 클래스 생성자 호출				
set_data	self, str, datetime.datetime, pathlib.Path	None	instance	public
이미지 정보 변경				

2.2. Core Class Diagram

주요 기능을 구현한 Core 부분의 클래스 다이어그램에 대해 설명한다. Core는 View에서의 데이터 요청을 처리하는 프로그램의 주요 로직이다.

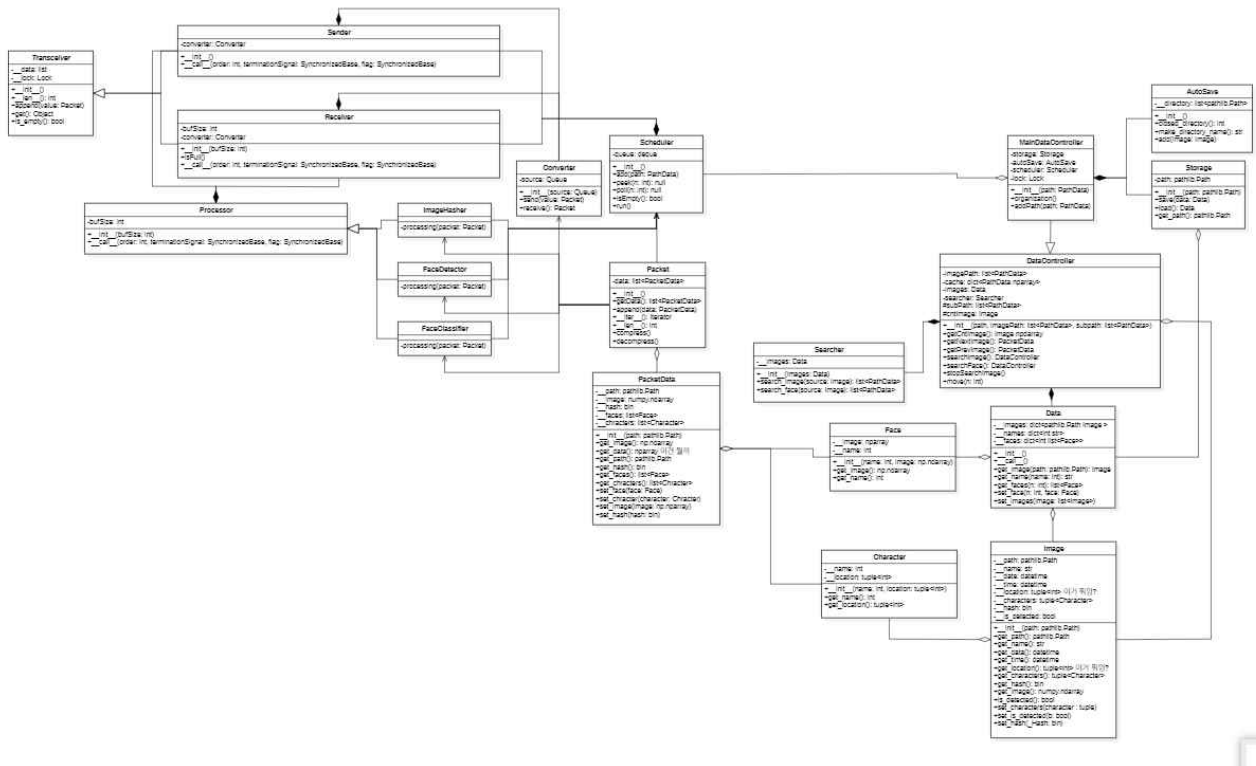


Fig. 3. Core Class Diagram

Fig. 3. 는 Core의 클래스 다이어그램이다. DataController 가 Core 부분의 동작을 제어하고, View에서 오는 이벤트를 처리한다. Scheduler는 멀티 프로세싱을 동작시키는 클래스로 ImageHasher, FaceDetector 그리고 FaceClassifier를 멀티 프로세싱하며 데이터를 빠르게 추출한다. 또한 추출된 데이터를 Data 클래스를 통해 저장하여 Searcher를 통해 얼굴 또는 유사 이미지를 검색할 수 있다.

2.2.1 Core Class Diagram Description

단순히 데이터를 저장하는 클래스인 Face, Character, Image, Data, PacketData 그리고 Packet 클래스에 대해서는 따로 설명하지 않았다.

DataController				
Class Description				
View에서 생성되는 이벤트를 처리하는 클래스				
Attributes				
구분				
Name	Type	Scope	Visibility	
Description				
__image_path	list<pathlib.Path>	instance	private	
탐색된 이미지 경로를 저장하는 리스트 변수				
__cache	dict<pathlib.Path numpy.ndarray>	instance	private	
불러온 이미지를 임시 저장하는 딕셔너리 변수				
__images	Data	instance	private	
Data 인스턴스를 가지는 변수				
__searcher	Searcher	instance	private	
Searcher 저장 변수				
_sub_path	dict<int, pathlib.Path>	instance	private	
new folder을 통해 생성된 경로를 저장하는 딕셔너리 변수				
_cnt_image_index	int	instance	private	
현재 이미지의 __image_path 에서의 인덱스 값을 저장하는 변수				
Operation				
구분				
Name	Argument	Returns	Scope	Visibility
Description				
__init__	self, list<pathlib.Path>, dict< int, pathlib.Path >	None	instance	public
이미지 경로 리스트와 folder 경로 딕셔너리를 받아 객체 생성				
get_cnt_image	self	Image, numpy.ndarray	instance	public
현재 이미지의 Image 객체와 이미지 데이터 반환				
get_next_image	self	Image, numpy.ndarray	instance	public
다음 이미지의 Image 객체와 이미지 데이터 반환				
get_prev_image	self	Image, numpy.ndarray	instance	public
이전 이미지의 Image 객체와 이미지 데이터 반환				
search_image	self	DataController	instance	public
현재 이미지와 유사한 이미지 검색 후 새로운 DataController 반환				

search_face	self, Character	DataController	instance	public
매개변수로 전달된 얼굴이 담겨있는 사진 검색 후 새로운 DataController 반환				
stop_search_image	self	None	instance	public
현재 DataController가 MainDataController가 아닐 시 자신을 제거				
move	self, int	None	instance	public
현재 이미지를 sub_path에서 전달된 정수 값을 키로 하는 경로에 이동				

MainDataController

Class Description

DataController를 상속 받으며 View의 이벤트를 처리하는 주요 컨트롤러 클래스

Attributes

구분

Name	Type	Scope	Visibility
Description			
__storage	Storage	instance	private
Storage의 인스턴스를 저장하는 변수			
__auto_save	AutoSave	instance	private
AutoSave의 인스턴스를 저장하는 변수			
__scheduler	Scheduler	instance	private
Scheduler 인스턴스를 저장하는 변수			
__lock	Lock	instance	private
쓰레드 사용으로 변수 접근 시 lock 사용을 위한 Lock 객체 저장 변수			

Operation

구분

Name	Argument	Returns	Scope	Visibility
Description				
__init__	self, pathlib.Path	None	instance	public
메인 경로를 전달 받아 인스턴스 생성				
organization	self	None	instance	public
__auto_save을 이용하여 현재 폴더를 자동 정리해주는 함수				
add_path	self, pathlib.Path	None	instance	public
현재의 sub_path에 새로운 경로 추가				

Searcher				
Class Description				
이미지를 검색해주는 클래스				
Attributes				
구분				
Name	Type	Scope	Visibility	
Description				
__images	Data	instance	private	
검색에 기반이 되는 데이터를 저장하는 변수				
Operation				
구분				
Name	Argument	Returns	Scope	Visibility
Description				
__init__	self, Data	None	instance	public
Data를 전달받아 멤버 변수 초기화 후 인스턴스 생성				
search_image	self, Image	list<pathlib.Path>	instance	public
__images의 데이터를 바탕으로 전달받은 Image와 유사한 이미지 검색 후 경로를 리스트로 만들어 반환				
search_face	self, Character	list<pathlib.Path>	instance	public
__iamges의 데이터를 바탕으로 전달받은 Character의 얼굴이 들어가 있는 이미지 검색 후 경로를 리스트로 만들어 반환				

Storage				
Class Description				
저장된 데이터를 불러오거나 데이터를 저장하는 클래스				
Attributes				
구분				
Name	Type	Scope	Visibility	
Description				
__path	pathlib.Path	instance	private	
저장되는 데이터의 경로를 저장하는 변수				
Operation				
구분				
Name	Argument	Returns	Scope	Visibility
Description				
__init__	self, pathlib.Path	None	instance	public
경로를 전달 받아 멤버 변수 초기화 및 인스턴스 생성				
save	self, Data	None	instance	public
전달 받은 Data 객체를 __path에 저장되어 있는 경로에 저장				
load	self	Data	instance	public
__path에 저장되어 있는 경로에 데이터를 로드 후 반환				
get_path	self	pathlib.Path	instance	public
path 반환				

AutoSave				
Class Description				
이미지를 속성에 따라 자동 분류하고 정리해주는 클래스				
Attributes				
구분				
Name	Type	Scope	Visibility	
Description				
__directory	list<pathlib.Path>	instance	private	
정리 할 디렉토리 후보 저장되는 리스트 변수				
Operation				
구분				
Name	Argument	Returns	Scope	Visibility
Description				
__init__	self	None	instance	public
멤버 변수 초기화 및 인스턴스 생성				
make_directory_name	self	list<str>	instance	public
디렉토리의 알맞은 이름 생성 후 반환				
add	self, Image	None	instance	public
이미지 객체 통해 이미지 분석 후 분류하여 __directory에 저장				

Scheduler				
Class Description				
백그라운드에서 이미지 정보를 추출하는 프로세스를 관리하는 스케줄러 클래스				
Attributes				
구분				
Name	Type	Scope	Visibility	
Description				
__queue	deque<pathlib.Path>	instance	private	
먼저 정보를 추출 해야 할 이미지 경로를 담은 큐를 저장하는 변수				
Operation				
구분				
Name	Argument	Returns	Scope	Visibility
Description				
__init__	self	None	instance	public
Scheduler 인스턴스 생성				
add	self, pathlib.Path	None	instance	public
__queue에 pathlib.Path를 추가				
peek	self	pathlib.Path	instance	public
__queue에 다음 pathlib.Path 반환. __queue에서 삭제하지 않음.				
poll	self	pathlib.Path	instance	public
__queue에 다음 pathlib.Path를 __queue에서 삭제하며 반환.				
isEmpty	self	bool	instance	public
__queue 가 비었는지 여부 반환				
run	self	None	instance	public
스케줄러 프로세스 시작하는 함수				

Transceiver				
Class Description				
멀티 프로세싱 간 프로세스 사이 송수신을 구현하는 추상 클래스				
Attributes				
구분				
Name	Type	Scope	Visibility	
Description				
__data	deque<Packet>	instance	private	
데이터를 저장하는 리스트 변수				
__lock	Lock	instance	private	
동시 접근을 막는 Lock 변수				
Operation				
구분				
Name	Argument	Returns	Scope	Visibility
Description				
__init__	self	None	instance	public
멤버 변수 초기화				
__len__	self	int	instance	public
__data 의 길이 반환				
append	self, Packet	None	instance	public
__data에 Packet 추가				
get	self	Packet	instance	public
__data의 Packet 반환				
is_empty	self	bool	instance	public
__data가 비어있으면 True, 비어있지 않으면 False 반환				
Sender				
Class Description				
Transceiver을 상속받아 송신기를 구현한 클래스				
Attributes				
구분				
Name	Type	Scope	Visibility	
Description				
__converter	Converter	instance	private	
Converter 인스턴스를 저장하고 있는 변수				
Operation				
구분				
Name	Argument	Returns	Scope	Visibility
Description				
__init__	self	None	instance	public
멤버 변수 초기화 및 인스턴스 생성				
__call__	self, int, SynchronizedBase, SynchronizedBase	int	instance	public
매개 변수의 정보를 통해 __data에 있는 데이터를 지속적으로 전달하는 함수				

Receiver				
Class Description				
Transceiver을 상속받아 수신기를 구현한 클래스				
Attributes				
구분				
Name	Type	Scope	Visibility	
Description				
__buf_size	int	instance	private	
buf size를 저장하고 있는 변수				
__converter	Converter	instance	private	
Converter 인스턴스를 저장하고 있는 변수				
Operation				
구분				
Name	Argument	Returns	Scope	Visibility
Description				
__init__	self, int	None	instance	public
버프 사이즈를 받아 초기화 후 인스턴스 생성				
isFull	self	bool	instance	public
__data 의 길이와 __buf_size 의 값을 비교하여 현재 __data가 꽉 찼는지 반환				
__call__	self, int, SynchronizedBase, SynchronizedBase	int	instance	public
매개 변수의 정보를 통해 데이터를 수신하여 __data에 지속적으로 저장하는 함수				

Processor				
Class Description				
멀티 프로세싱에서 프로세스의 기반을 구현한 클래스				
Attributes				
구분				
Name	Type	Scope	Visibility	
Description				
__buf_size	int	instance	private	
buf size를 저장하고 있는 변수				
Operation				
구분				
Name	Argument	Returns	Scope	Visibility
Description				
__init__	self, int	None	instance	public
버퍼 사이즈를 받아 초기화 후 인스턴스 생성				
_processing	self, Packet	Packet	instance	protected
추상 함수로 자식 클래스가 필요한 로직을 구현				
__call__	self, int, SynchronizedBase, SynchronizedBase	int	instance	public
매개 변수의 정보를 통해 지속적으로 데이터를 수신하여 processing 함수로 가공하고 sender로 전송하는 함수				
ImageHasher				
Class Description				
Processor을 상속 받는 클래스로 이미지를 읽고 분석하여 해시 값을 구하는 클래스				
Attributes				
None				
Operation				
구분				
Name	Argument	Returns	Scope	Visibility
Description				
_processing	self, Packet	Packet	instance	protected
부모 클래스의 _processing을 오버라이딩 한 함수로 이미지를 읽고 분석하여 해시 값을 구하는 클래스 이다. 읽은 이미지는 Packet에 담겨 전달 된다.				

FaceDetector				
Class Description				
Processor을 상속 받는 클래스로 이미지에서 얼굴을 탐색한다				
Attributes				
None				
Operation				
구분				
Name	Argument	Returns	Scope	Visibility
Description				
_processing	self, Packet	Packet	instance	protected
부모 클래스의 _processing을 오버라이딩 한 함수로 이미지에서 얼굴을 탐색하여 정보를 Packet에 담아 전달한다.				
FaceClassifier				
Class Description				
Processor을 상속 받는 클래스로 전달 받은 얼굴 정보를 바탕으로 얼굴을 분류한다.				
Attributes				
None				
Operation				
구분				
Name	Argument	Returns	Scope	Visibility
Description				
_processing	self, Packet	Packet	instance	protected
부모 클래스의 _processing을 오버라이딩 한 함수로 얼굴 데이터를 읽어 분류한다.				

3. Sequence diagram

본 챕터에서는 Use Case를 바탕으로 시간을 기준으로 시스템의 흐름을 나타내는 시퀀스 다이어그램에 대해 설명한다.

3.1 Init Path

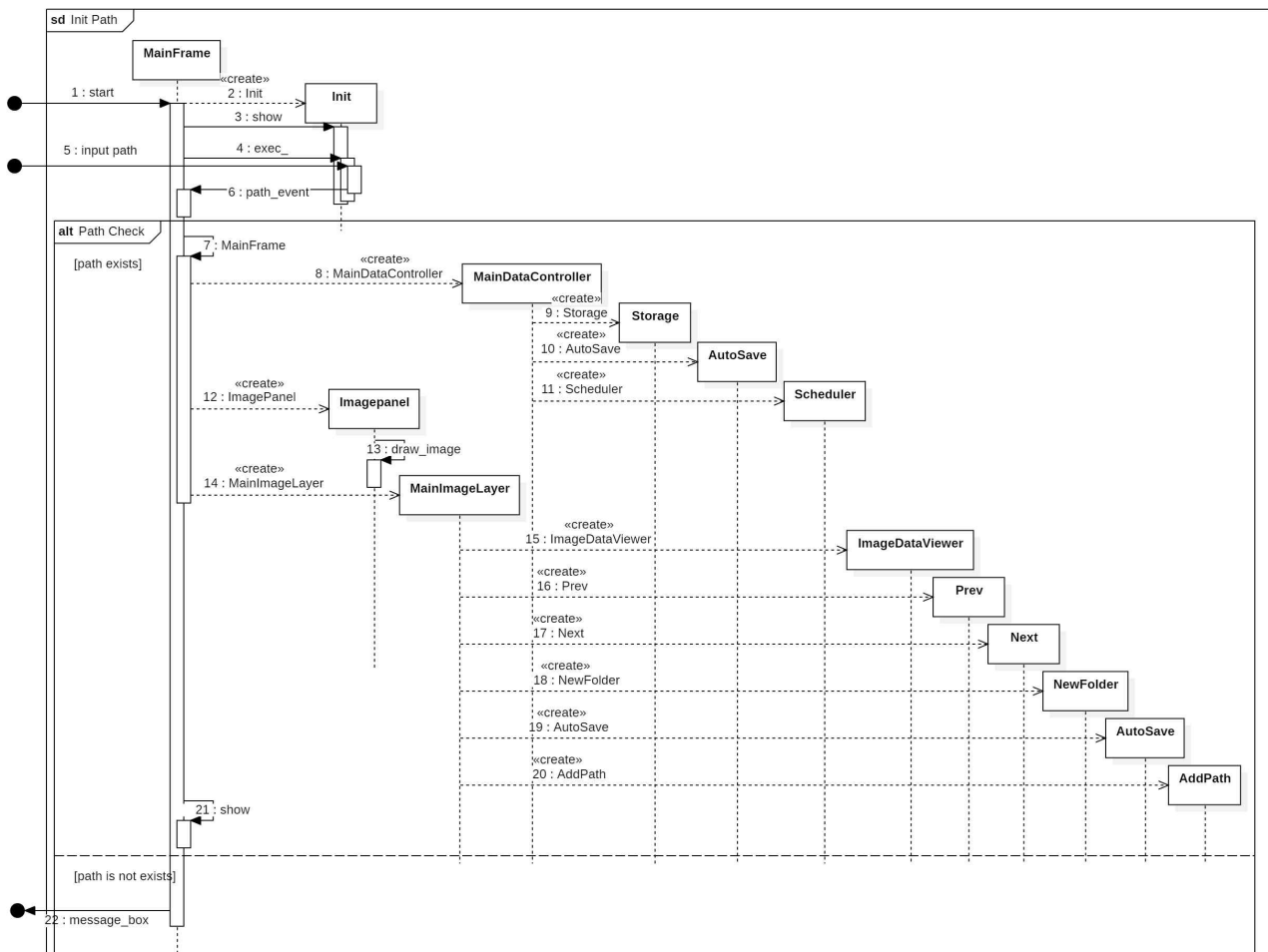


Fig. 4. Init Path Sequence Diagram

Fig. 4. 는 Init Path (Use Case #1)의 시퀀스 다이어그램이다. 사용자가 프로그램을 시작하면 MainFrame 클래스의 스텁 메소드가 실행된다(1. **start**). 스텁 메소드에서 Init 인스턴스를 생성(2. **Init**), 사용자에게 보여주며(3. **show**), 사용자의 입력을 기다린다(4. **exec**). 사용자가 경로를 입력하면(5. **input path**), Init 인스턴스는 콜백함수(6. **path_event**)를 실행하여 입력된 값을 MainFrame으로 전달한다. MainFrame은 전달받은 시스템 경로가 유효한지 확인하고 두 가지 동작을 수행한다.

먼저 첫 번째 동작은 전달받은 시스템 경로가 유효할 때 실행된다. MainFrame의 인스턴스가 만들어지며(7. **MainFrame**) 이를 위해 하위 요소의 인스턴스가 모두 생성된다. 먼저 MainDataController(8. **MainDataController**)가 생성되고, 그 하위 요소 인스턴스

인, Storage(9. **Storage**), AutoSave(10. **AutoSave**), Scheduler(11. **Scheduler**)가 생성된다. 다음으로 ImagePanel(12. **ImagePanel**)이 생성되고 이미지를 그린다(13. **draw_image**). 마지막으로 MainImageLayer 인스턴스가 생성되고(14. **MainImageLayer**) 그 하위요소인 ImageDataViewer(15. **ImageDataViewer**), Prev(16. **Prev**), Next(17. **Next**), NewFolder(18. **NewFolder**), AutoSave(19. **AutoSave**), AddPath(20. **AddPath**)가 생성된다. MainFrame의 인스턴스 생성 과정을 모두 마치면 MainFrame의 생성자가 끝나고 인스턴스가 만들어진다. 그리고 스스로를 표시하며 종료된다(21. **show**).

두 번째 동작은 전달받은 시스템 경로가 유효하지 않을 때 실행된다. 프로그램을 추가적인 동작을 하지 않고 경고 메시지를 출력한다.

3.2 Hover Cursor Over Image

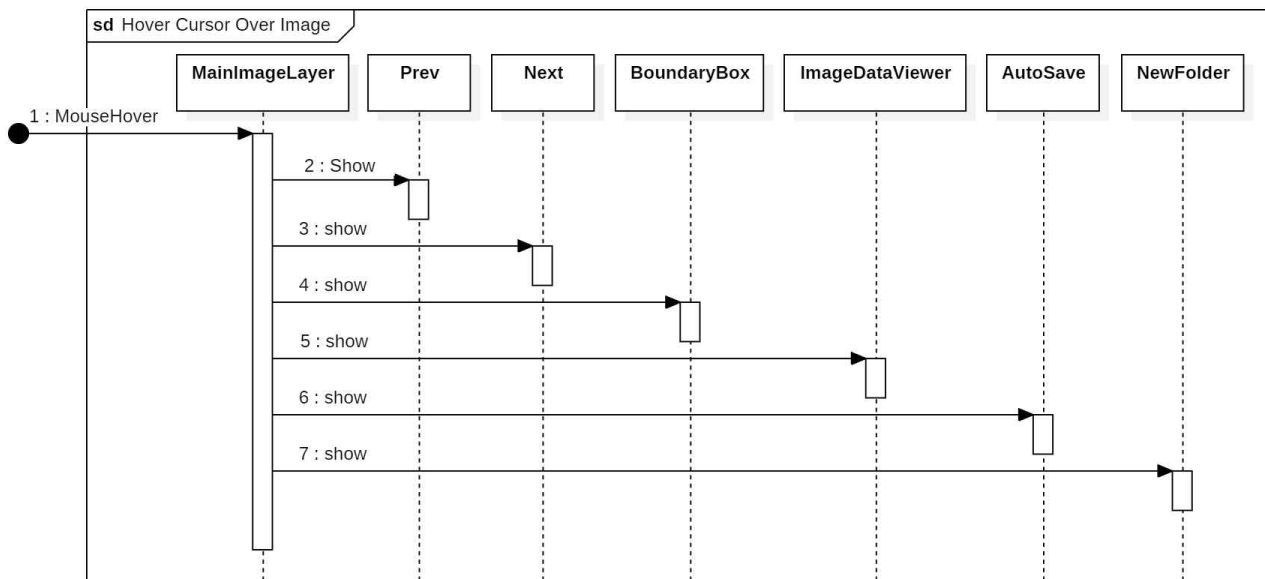


Fig. 5. Hover Cursor Over Image Sequence Diagram

Fig. 5. 는 Hover Cursor Over Image (Use Case #2)의 시퀀스 다이어그램이다. 사용자가 프로그램 위에 마우스를 올리면 시작된다(1. **MouseHover**). MainImageLayer에서 MouseHover 함수를 통해 감지되어 표시해야 할 모든 상호작용 오브젝트에 show((2, 3, 4, 5, 6, 7) **show**) 함수를 호출한다. 마우스가 프로그램 위를 떠난다면 위와 마찬가지로 동작하기에 따로 추가하지 않았다.

3.3 Add Path

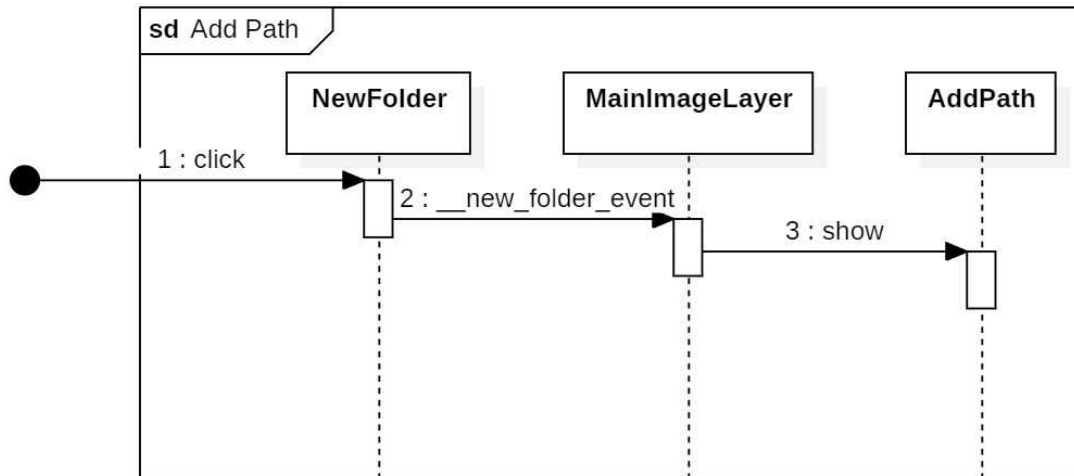


Fig. 6. Add Path Sequence Diagram

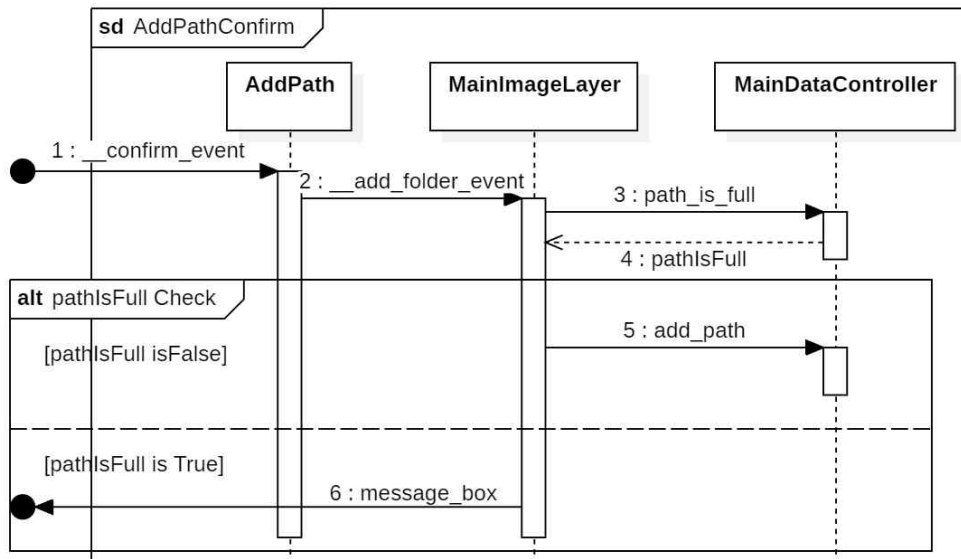


Fig. 7. Add Path Confirm Sequence Diagram

Fig. 6. 은 Add Path (Use Case #3)를 나타낸 시퀀스 다이어그램이다. New Folder를 클릭하면(1: click). NewFolder에서 MainImageLayer에 있는 callback 함수를 호출한다(2: __new_folder_event). 콜백함수는 AddPath 인스턴스에 show 메소드(3: show)를 호출하여 사용자에게 경로를 추가할 수 있는 모달 창을 보여준다.

Fig. 7. 은 경로를 추가하는 모달 창의 경로 입력을 나타낸 시퀀스 다이어그램이다. 사용자가 경로 입력란에 경로를 입력하거나 또는 Browse 버튼을 통해서 경로를 선택한 다음의 동작을 보여준다(Browse 버튼은 라이브러리로 구현되어 있어 따로 설명하지 않는다.). 경로를 입력하고 확인 버튼을 누르면 AddPath의 확인 버튼 이벤트(1: __confirm_event)가 실행된다. 이는 콜백 함수인 MainImageLayer의 메소드가 실행시킨다(2: __add_folder_event). 콜백 함수는 MainDataController에 시스템 서브 폴더가

꼭 차 있는지 `path_is_full(3: path_is_full)` 메소드로 확인한다. 이때 `pathIsFull(4: pathIsFull)` 값에 따라 두 가지 동작을 한다.

먼저 `pathIsFull`이 `False` 라면 `MainImageLayer` 인스턴스에서 다시 `MainDataController`의 `add_path` 메소드(5: `add_path`)를 호출하여 입력받은 경로를 추가한다.

두 번째로 `pathIsFull`이 `True` 라면 `MainImageLayer`에서 `message_box`를 호출하여 사용자에게 더 이상 폴더를 추가할 수 없음을 알린다.

3.4 Zoom In & Out Photo

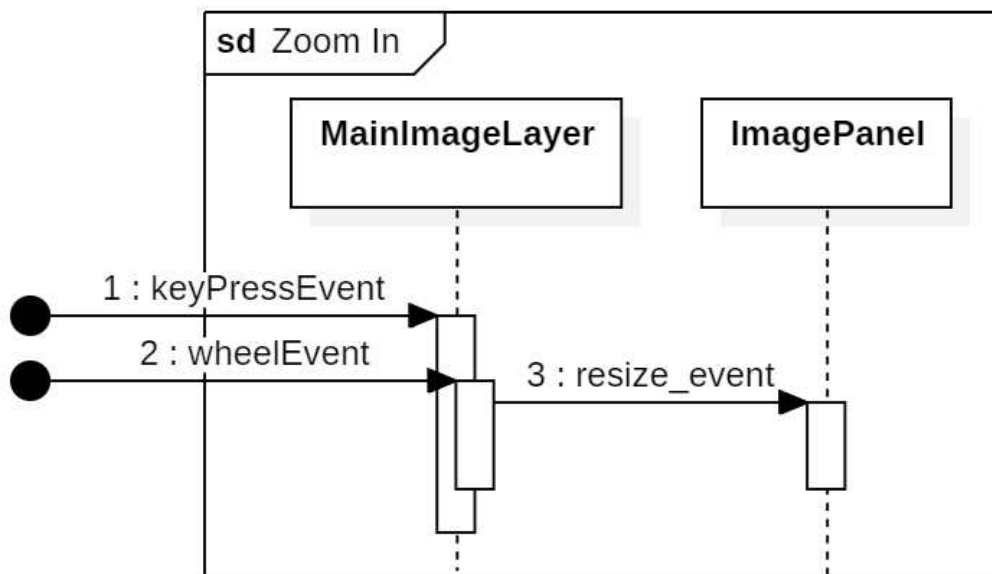


Fig. 8. Zoom In Photo Confirm Sequence Diagram

Fig. 8. 은 Zoom In (Use Case #4)를 나타낸 시퀀스 다이어그램이다(Zoom out은 Zoom In과 완전히 동일하여 따로 설명하지 않는다.). Zoom In을 하려면 사용자는 컨트롤 키와 휠 업을 해야 한다. 따라서 `MainImageLayer`에서 `keyPressEvent(1: keyPressedEvent)`와 `wheelEvent(2: wheelEvent)`를 이용하여 컨트롤 키가 눌린 시점에 휠 동작을 하는지 체크한다. 컨트롤 키가 눌린 상태에서 휠 동작이 감지 되면 `MainImageLayer`가 `ImagePanel`의 `resize_event` 메소드(3: `resize_event`)를 호출하여 이미지의 크기를 조정한다. Zoom In과 Zoom Out의 차이는 `wheelEvent` 메소드에서 `resize_event`로 전달되는 값의 차이이다.

3.5 Browse Photos

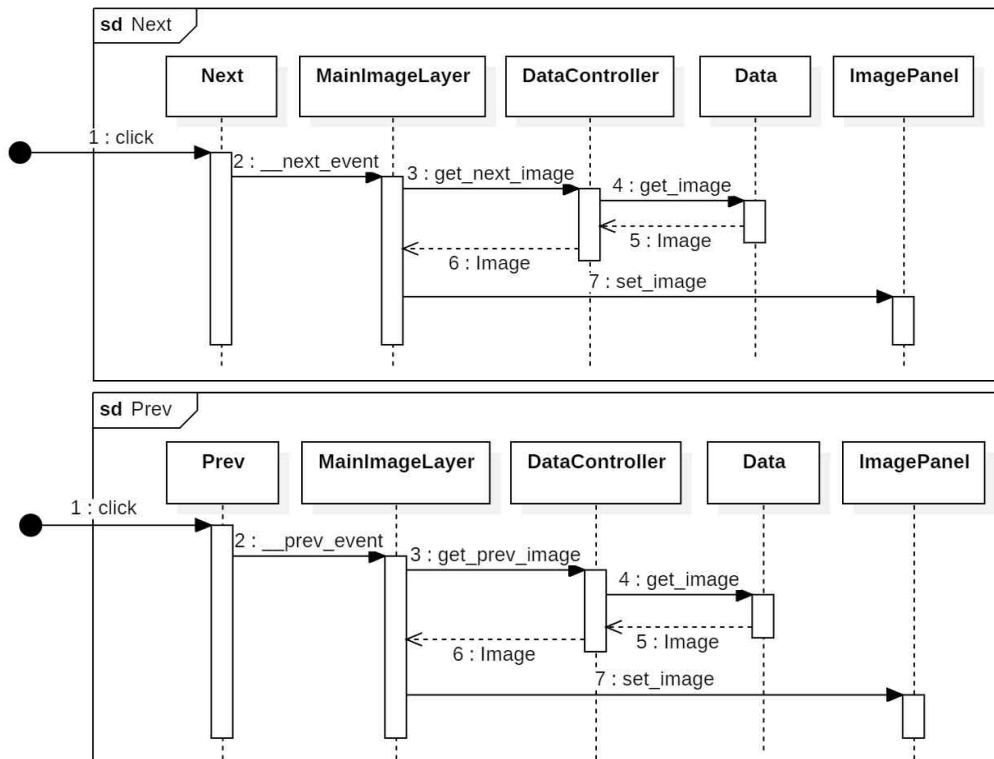


Fig. 9. Browse Photos Sequence Diagram

Fig. 9. 는 Browse Photos (Use Case #5)를 나타낸 시퀀스 다이어그램이다. Next 동작과 Prev 동작을 따로 나누어져 있지만, 전체적인 동작을 동일하므로 Next를 기준으로 설명한다. 먼저 사용자가 Next 인스턴스 즉 버튼을 클릭하면서 시작된다(1: click). 버튼을 클릭하면 Next 인스턴스에서 MainImageLayer에 있는 콜백함수를 실행한다(2: __next_event). 이후 DataController의 get_next_image(3: get_next_image)를 호출하여 다음 이미지를 가져온다. 다음 DataController의 get_image(4: get_image) 메소드를 호출하여 Data에서 Image 정보를 받은 다음 이 Image 정보를 DataController의 cache에 저장하고 MainImageLayer에 다시 반환한다. 다음 set_image(7: set_image)를 호출하여 ImagePanel의 이미지를 업데이트 한다.

3.6 Search Photos Sequence Diagram

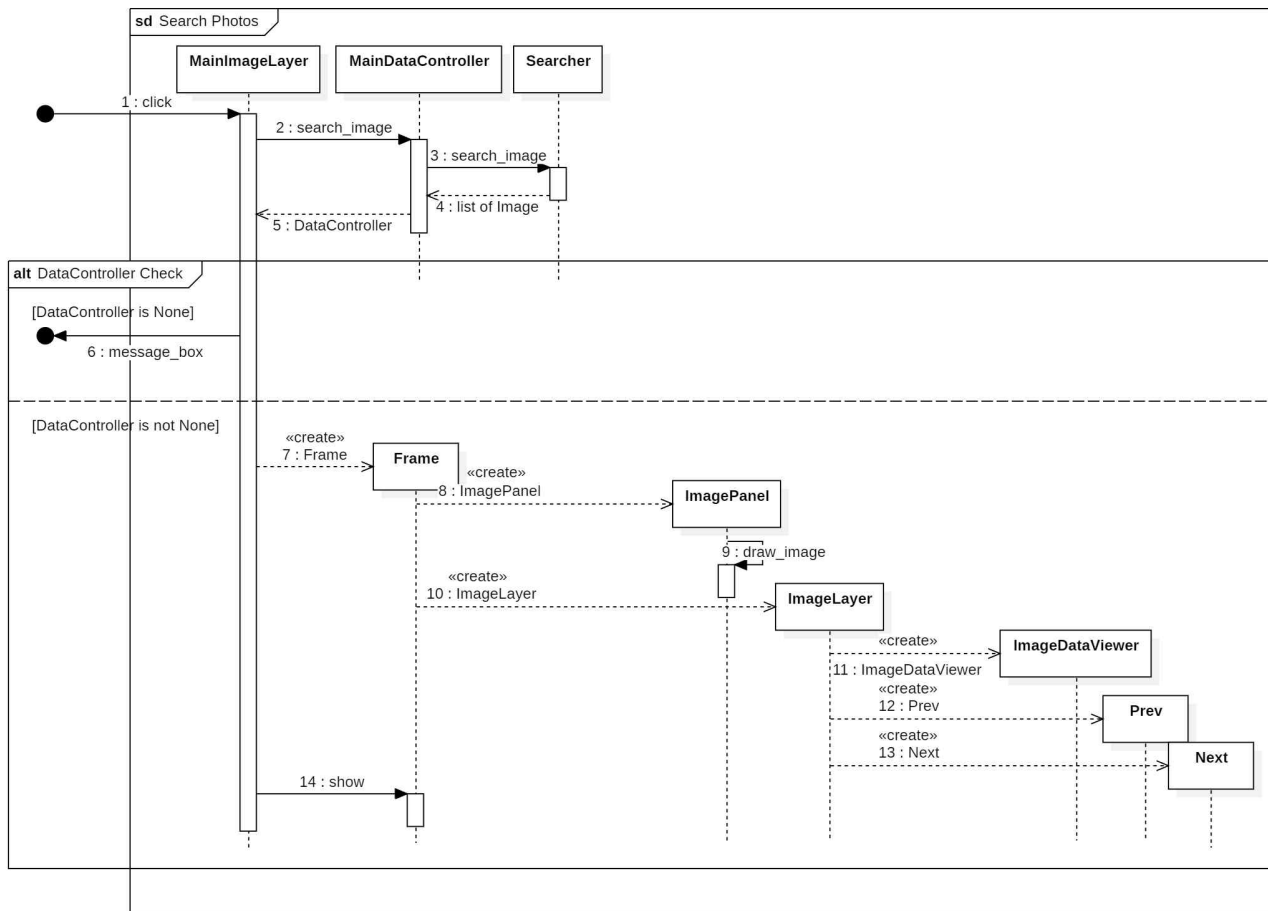


Fig. 10. Search Photos Sequence Diagram

Fig. 10.은 Search Photos (Use Case #7)를 나타내는 시퀀스 다이어그램이다. MainImageLayer의 click 메소드를 통해서 사용자가 사진을 클릭하는 입력을 받으면서 시작된다. MainImageLayer에서 MainDataController의 search_image(2: search_image) 메소드를 통해서 Searcher의 search_image(3: search_image)를 이용하여 이미지를 검색한다. 검색된 이미지를 리스트로 반환되며 MainDataController에서 이를 바탕으로 한 DataController를 반환한다. 만약 리스트가 비어있다면 None을 반환한다. 따라서 다음 동작은 두 가지이다.

먼저 첫 번째 동작은 반환된 DataController의 값이 None일 때다. 이때는 사용자에게 검색된 이미지가 없음을 뜻하는 경고 창을 띄어주고 끝난다(6: message_box).

두 번째는 DataController의 값이 None이 아닐 때다. 동작은 새로운 탐색 창을 만들어 준다. 먼저 Frame 인스턴스를 만든다. 만드는 과정은 앞서 설명했기에 생략한다. 인스턴스를 만들면 show(14: show)를 호출하여 사용자에게 보여준다.

3.7 Search Faces

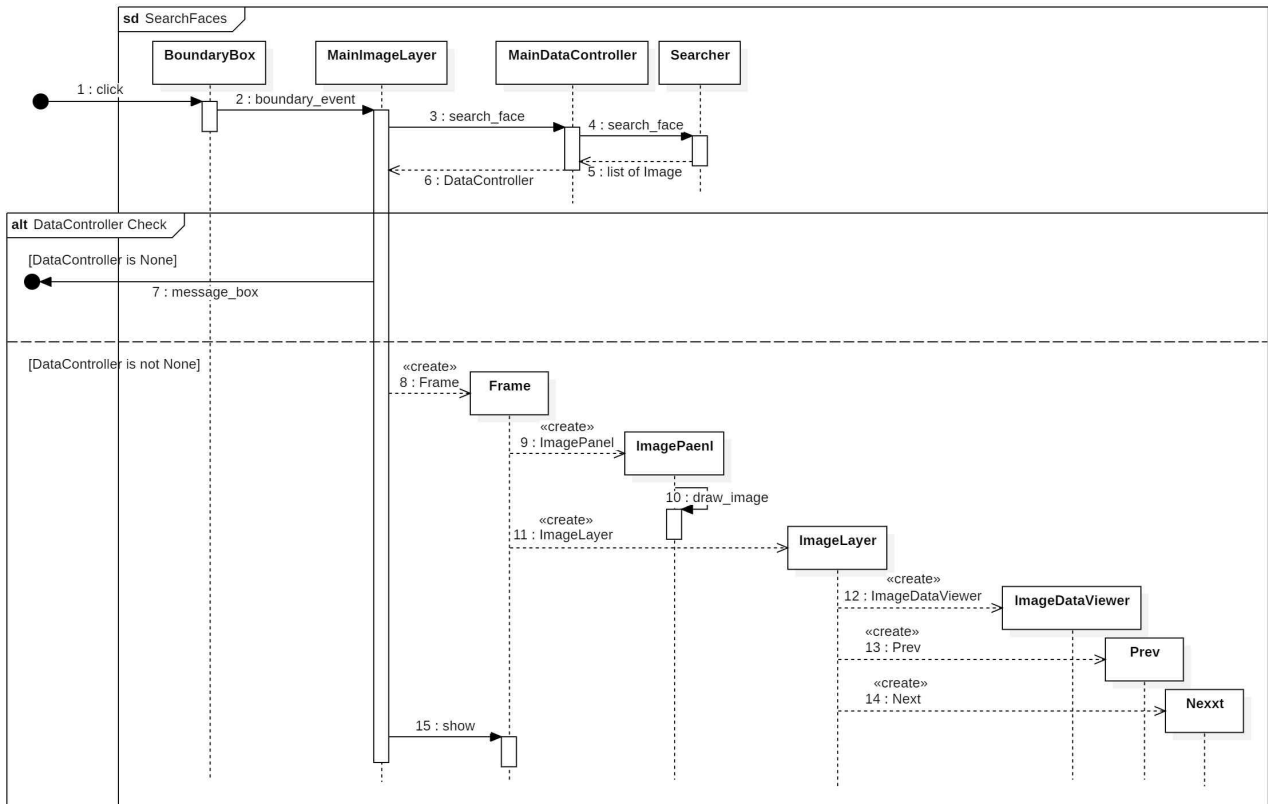


Fig. 11. Search Faces Sequence Diagram

Fig. 11.은 Search Faces (Use Case #8)을 나타낸 시퀀스 다이어그램이다. 사용자는 BoundaryBox 인스턴스가 나타내는 얼굴을 클릭하면서 시작된다(1: **click**). BoundaryBox 인스턴스는 MainImageLayer의 콜백 함수(2: **boundary_event**)를 실행하며 이는 MainDataController의 **search_face**(3: **search_face**)를 호출한다. 이 메소드는 Searcher 인스턴스의 **search_face**(4: **search_face**)를 호출하여 클릭한 얼굴과 동일한 얼굴이 포함되어 있는 이미지를 검색하여 리스트로 반환한다. MainDataController는 반환된 이미지 리스트를 바탕으로 DataController 인스턴스를 만들어 MainImageLayer에 전달한다. 만약 이미지 리스트가 비어있다면 None을 반환한다. 이후의 과정(8: **Frame**)은 3.6 Search Photos의 Frame(7: **Frame**) 이후와 동일하다.

3.8 Move Photo

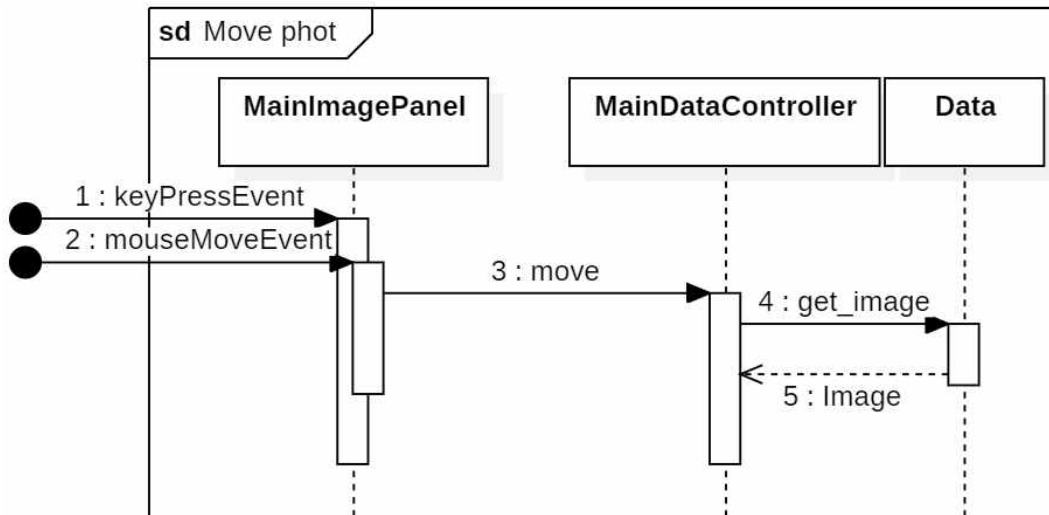


Fig. 12. Move Photo Sequence Diagram

Fig. 12. 는 Move Photo (Use Case #6)를 나타낸 시퀀스 다이어그램이다. 사용자는 마우스 좌클릭과 드래그를 통해서 현재 이미지를 특정 폴더로 옮긴다. 이를 위해 MainImageLayer의 keyPressEvent(1: **keyPressEvent**)와 mouseMoveEvent(2: **mouseMoveEvent**)를 사용한다. 각각 마우스 좌클릭과 마우스의 움직임을 인식한다. 마우스 좌클릭을 누른 상태로 마우스를 움직이면 MainImagePanel은 MainDataController의 move(3: **move**) 메소드를 호출한다. 이는 Data 클래스의 get_image(4: **get_image**)를 호출하여 현재 이미지의 정보를 가져와 위치를 이동시킨다.

3.9 Auto Organize Sequence Diagram

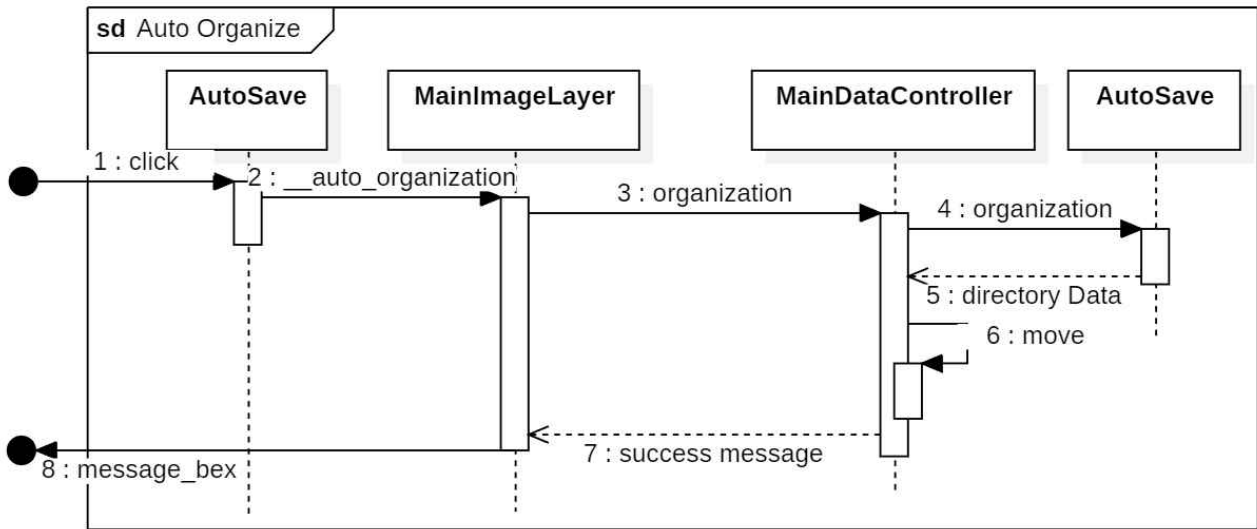


Fig. 13. Auto Organize Sequence Diagram

Fig. 13.은 Auto Organize (Use Case #9)의 시퀀스 다이어그램이다. 사용자가 AutoSave의 인스턴스를 클릭하며 실행된다(1: click). 이는 MainImageLayer의 콜백 함수(2: `__auto_organization`)를 호출한다. 다시 이는 MainDataController의 `organization`(3: `organization`)을 호출하고 이 함수는 또 다시 AutoSave의 `organization`(4: `organization`) 함수를 호출한다. AutoSave는 자동 정리를 마치고 `directory Data`를 반환한다. MainDataController의 `organization` 메소드는 반환된 데이터를 바탕으로 이미지를 이동시킨다(6: `move`). 이후 정리가 완료되면 성공 메시지를 MainImageLayer로 전달해 주고 마지막으로 사용자에게 `message_box` 함수를 사용하여 성공됐음을 전달한다.

5. Implementation requirements

H/W platform requirements	
Processor	Intel or AMD 8-core or higher CPU
Memory	4GB RAM
Storage	10GB Usable Space
S/W platform requirements	
OS	Windows 7 or higher

6. Glossary

7. References