
No-code & Low-code

개발 툴

설계제안서

제출일	2022.05.03	소속학부	소프트웨어학부
과목명	캡스톤디자인 종합프로젝트 2	조명	ASDF
담당교수	이상준	조원	김동현
지도교수	이상준		장혜진

- 목 차 -

1. 프로젝트 소개	3
2. 기존 시스템 분석	4
3. 특징 및 기능	6
4. 인터페이스	8
5. 소프트웨어 요구사항	11
6. 시스템 대블록 설계	15
7. 액티비티 기반 소블록 인터페이스	16
8. 구성도	17
9. 유즈케이스 다이어그램	20
10. 클래스 다이어그램	21
11. 시퀀스 다이어그램	22
12. 알고리즘 구성	28
13. 데이터 및 파일 시스템 구조	30
14. 시나리오	33
15. 역할 분담 및 개발 일정	35
16. 관련 기술 및 개념	36
17. 기대효과	37
18. 개발도구 및 개발환경	38
19. 참고 문헌	40

1. 프로젝트 소개

본 프로젝트는 프로그래밍을 학습하는 입문자나 관련 지식이 없어 익숙하지 않은 개발환경에 진입장벽을 느끼는 비전공자들에게 PC 환경에서 자신이 원하는 프로그램을 쉽고 간단하게 제작할 수 있는 프로그래밍 툴을 제공한다.

프로그램이 제공하는 위젯을 그대로 사용하여 관련 지식 없이도 GUI 프로그램을 제작할 수 있고, 사용자가 직접 위젯을 수정하거나 확장하여 자신이 원하는 특수한 기능을 가진 GUI 프로그램을 제작할 수 있도록 한다.

최종적으로, 프로그래밍을 잘 모르거나 입문 단계에 있는 사람들에게 쉽고, 간단하고, 효율적으로 GUI 프로그래밍을 할 수 있는 PC 환경의 No-code, Low-code 프로그래밍 툴을 지원하여 프로그래밍이 어렵고 복잡하지 않다는 인식을 심어주고, 사용자들에게 프로그램 개발에 대한 진입장벽을 낮추고 접근성을 높이면서 동기를 부여하는 것과 학습에 대한 의욕을 가지도록 함을 목표로 한다.

2. 기존 시스템 분석

기존의 No-code & Low-code 개발 플랫폼의 개발 대상은 모바일 어플리케이션, 웹 어플리케이션, 웹 사이트 등의 PC 환경 일반 프로그램이 아니거나, 비즈니스 솔루션, 워크플로 자동화 등의 기능을 지원하는 프로그램 등이 대다수이다. 이는 보통 기업의 생산성을 향상하는 용도로 사용되거나 기업 홍보 등에 이용되며, 프로그래밍 입문자나 비전공자 등의 일반적인 사용자들을 위한 것이 아니었고 비즈니스, 워크플로와 관련된 배경지식이 요구된다. 이러한 특성으로 인해, 일반적으로 PC 환경에서의 프로그래밍을 학습하는 입문자나 관련 지식이 없는 비전공자는 익숙하지 않은 개발환경에 진입장벽을 느낄 수밖에 없었고 자신이 원하는 PC 프로그램을 제작하기에 어려움을 가지게 된다.

2.1 기존의 No-code & Low-code 플랫폼

Draftbit	https://draftbit.com/
Low-code 모바일 앱 제작, react native 소스코드 산출 가능	
Amazon Web Services- honeycode	https://www.honeycode.aws/
No-code 웹, 모바일 앱 구축, 데이터베이스, 비즈니스 로직 , UI 제작	

Outsystems	https://www.outsystems.com/
No-code & Low-code 로 비즈니스 용도의 웹 앱, PC 프로그램 제작	
DevOn NCD	https://www.lgcns.com/Solution/DevOn-NCD
flow chart 기반, No-code 로 비즈니스 로직 개발	

AppGyver	https://www.appgyver.com/
Low-code 로 모바일 앱, 웹 앱 개발	
Microsoft Powerapp	비즈니스 앱 Microsoft Power Apps
No-code & Low-code 비즈니스 앱 개발 도구	

Softpower Smartmaker	스마트메이커-소개 (smartmaker.com)
No-code 안드로이드 앱, 아두이노 개발 도구	
Glide	Glide – Create apps & websites without code. (glideapps.com)
스프레드 시트 함수를 모바일 앱으로 변환	

Adalo	https://www.adalo.com/
템플릿을 통해 제작한 앱 페이지를 연결하여 하나의 모바일 앱 제작	
FlutterFlow	https://flutterflow.io/
No-code 모바일 앱 제작, 코드 다운로드 가능	

3. 특징 및 기능

- 특징

기존의 No-code & Low-code 개발 플랫폼과 달리 PC 환경에서 실행되는 일반 프로그램을 제작한다. 개발 툴은 메모장, 그림판 등 서로 다른 기능을 수행하는 서브 프로그램인 위젯을 제공하며, 사용자는 위젯을 빈 화면인 프로그래밍 인터페이스에 추가하고, 그 기능을 서로 연결하는 방식으로 코딩 작업 없이 자신이 원하는 GUI 프로그램을 제작한다. 사용자가 기본으로 제공하는 것 이상의 기능을 가지는 위젯이 필요한 경우, Python 프로그래밍을 통하여 자신이 원하는 기능을 가지는 사용자 위젯을 제작할 수 있으며, 기본 위젯과 같은 방식으로 사용 가능하다. 최종적으로 사용자는 약간에 코딩 작업 으로, 또는 코딩 작업 없이 자신이 원하는 GUI 프로그램을 간단히 제작할 수 있다.

- 기능

1. 사용 가능한 위젯을 표시하는 '위젯 리스트 인터페이스'와 위젯이 추가되는 '프로그래밍 인터페이스' 제공

현재 개발 툴이 제공하는 위젯의 종류를 확인하고 표시하는 '위젯 리스트 인터페이스'를 제공한다. 또한 사용자가 위젯을 추가하여 GUI 프로그램 인터페이스를 제작할 수 있는 '프로그래밍 인터페이스'를 제공한다.

2. 새로운 '위젯 파일' 생성

개발 툴은 고유한 기능을 가지는 서브 프로그램인 '위젯'을 제공하며, 위젯의 Python 코드를 가진 '위젯 파일' 또한 제공한다. 사용자가 자신이 원하는 특수한 기능을 가지는 위젯을 사용하고 싶은 경우, Python 프로그래밍을 통해 새로운 '위젯 파일'을 생성하여 위젯을 제작할 수 있다.

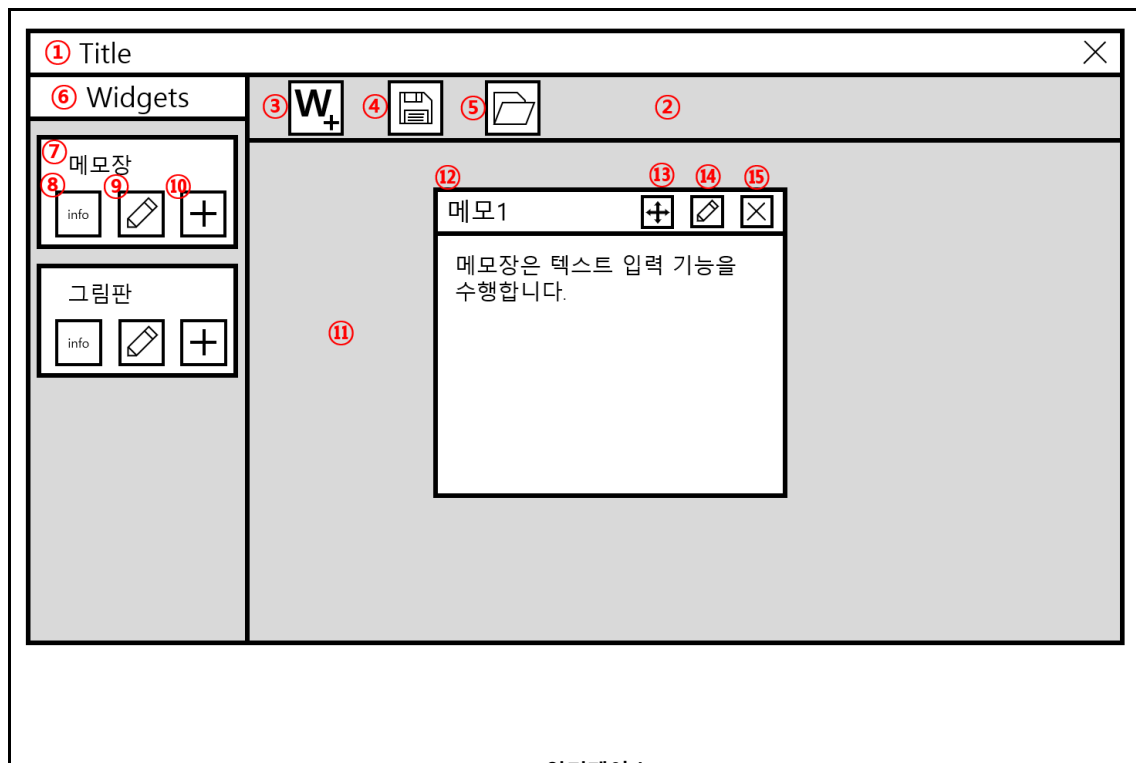
3. '위젯'을 생성하여 프로그래밍 인터페이스에 추가

개발 툴에서 사용 가능한 위젯을 위젯 리스트 인터페이스에서 확인한 후, 사용하고자 하는 위젯을 생성하여 프로그래밍 인터페이스에 추가한다. 생성된 위젯은 고유의 기능을 수행한다.

4. '연결 위젯'을 통해 서로 다른 기능을 가지는 위젯을 연결하여 '프로그램' 제작

서로 다른 위젯들의 독립된 기능을 하나의 '연결 위젯'에 연결한다. '연결 위젯'은 연결된 위젯들의 기능을 사용하여 자신의 기능을 확장한다. '연결 위젯'을 통해 서로의 기능이 유기적으로 연결된 위젯들은 하나의 온전한 기능을 수행하는 '프로그램'이 된다.

4. 인터페이스 (UI)



4. 인터페이스

- ① 타이틀 : 프로그램의 타이틀을 표시한다.

- ② 툴바 : 개발 툴의 '위젯 파일 생성' 기능과 '프로그램 저장', '프로그램 불러오기' 등의 기능을 사용할 수 있는 버튼을 제공한다.
- ③ 위젯 파일 생성 버튼 : 새로운 '위젯 파일'을 생성한다. 파일은 위젯이 필수적으로 가져야 하는 라이브러리, 변수, 함수 등이 미리 작성되어 있는 Python 파일이다.
- ④ 프로그램 저장 버튼 : 현재 프로그래밍 인터페이스에서 사용중인 위젯들, 즉 프로그램의 정보를 저장한다.
- ⑤ 프로그램 불러오기 버튼 : 저장되어 있는 프로그램 파일을 불러온 후, 프로그래밍 인터페이스에 프로그램을 생성한다.
- ⑥ 위젯 리스트 인터페이스 : 현재 개발 툴에서 사용 가능한 '위젯 파일'의 목록을 표시한다.
- ⑦ 사용 가능한 위젯 파일 : 사용 가능한 각 '위젯 파일'의 이름과 '정보' 버튼, '코드 수정' 버튼, '추가' 버튼을 제공한다.
- ⑧ 정보 버튼 : 해당 '위젯 파일'에 대한 정보와 위젯이 수행하는 기능을 간략한 텍스트로 표시한다.
- ⑨ 코드 수정 버튼 : 해당 '위젯 파일'의 Python 코드를 수정할 수 있도록 위젯의 Python 파일을 연다.

- ⑩ 추가 버튼 : 해당 '위젯 파일'의 위젯을 생성하여 프로그래밍 인터페이스에 추가한다. 추가 시 위젯이 필수적으로 가져야 하는 정보(이름 등)을 입력 받는다.
- ⑪ 프로그래밍 인터페이스 : 위젯 리스트 인터페이스에서 추가 버튼을 통해 만들어진 위젯이 추가되어 기능을 수행한다.
- ⑫ 사용중인 위젯 : 현재 프로그램에서 사용중인 '위젯'이다. 예시로 사용된 것은 메모장 위젯으로, 이름은 "메모 1"이며 "메모장은 텍스트 입력 기능을 수행합니다."라는 텍스트를 표시하는 기능을 수행하고 있다. 또한 위젯이 기본적으로 가져야 할 기능을 사용할 수 있도록 하는 '위젯 위치 이동' 버튼, '위젯 데이터 수정' 버튼, '위젯 삭제' 버튼을 제공한다.
- ⑬ 위젯 위치 이동 버튼 : 드래그 앤 드롭을 통하여 프로그래밍 인터페이스 내부에서 위젯의 위치를 옮긴다.
- ⑭ 위젯 데이터 수정 버튼 : 위젯 생성시 입력한 데이터를 수정한다.
- ⑮ 위젯 삭제 버튼 : 위젯을 삭제한다. 위젯 삭제시엔 "해당 위젯이 연결 대상인가?, 연결 위젯인가?" 등의 조건을 검사하여 삭제가 가능한지 불가능한지 판단한 후에 삭제를 실행하거나 취소한다.

5. 소프트웨어 요구사항(Software Requirements)

- 기능적 요구사항 (Functional Requirements)

개발 톨 및 각 인터페이스

1. 사용 가능한 '위젯 파일'을 표시하는 '위젯 리스트 인터페이스'와 '위젯'이 추가될 '프로그래밍 인터페이스' 표시
2. 사용 가능한 '위젯 파일'을 확인하여 위젯 리스트 인터페이스에 표시
3. 새로운 '위젯 파일'을 제작하여 위젯 리스트 인터페이스에 추가
4. 기존에 생성된 '위젯 파일'의 코드를 수정
5. 위젯 리스트 인터페이스의 특정 '위젯 파일'에 대한 '위젯'을 생성하여 프로그래밍 인터페이스에 추가
6. '위젯'을 프로그래밍 인터페이스의 적절한 위치에 배치
7. 현재 프로그래밍 인터페이스에서 사용중인 위젯들, 즉 프로그램의 정보를 저장 또는 불러오기

위젯

1. 위젯의 기능에 대한 정보 출력
2. 위젯 생성 시 필수적으로 필요한 정보를 입력 받은 후 생성
3. 위젯의 위치 이동

4. 위젯 데이터 수정 및 삭제
5. 저장 / 불러오기 기능을 위한 위젯의 정보 송신 / 수신 기능
6. 위젯 실행 시 다른 위젯의 실행에 영향을 주지 않도록 하기 위해 스레드 사용

- 비기능적 요구사항 (Non-functional Requirements)

1. 효율성 요구 사항(efficiency requirements)

위젯을 표시하는 프로그래밍 인터페이스의 크기가 한정되어 있고, 제한없이 생성되는 위젯은 메모리의 과도한 사용을 유발하므로, 이를 방지하기 위하여 위젯이 특정 개수 이상 만들어지지 않도록 제어한다.

2. 신뢰성 요구 사항 (reliability requirements)

위젯 데이터 수정 및 삭제 시, 연결 되어있는 다른 위젯의 기능 수행에 오류가 발생할 수 있으므로, 다음과 같은 확인 과정을 거친다.

1. 자신을 연결 대상으로 사용하는 '연결 위젯'이 있는가?
 - a. 연결 대상으로 사용되는 위젯인 경우, 수정 및 삭제 시 연결 위젯에 기능 오류가 발생할 수 있으므로 수정 및 삭제가 불가능하다.

b. 연결 대상으로 사용되지 않는 경우, 자신이 '연결 위젯'인가?

- i. '연결 위젯'이라면, 연결중인 위젯들의 실행에 악영향을 줄 수 있음을 확인한 후, 사용자의 동의에 따라 수정 및 삭제가 가능하다.
- ii. '연결 위젯'이 아니라면, 수정 및 삭제가 영향을 미칠 위젯이 없다고 판단하여 수정 및 삭제가 가능하다.

'연결 위젯'에서 위젯을 연결하려 할 시, 그 위젯이 이미 다른 연결 위젯에서 사용중인 경우, 기능 수행에 오류가 발생할 수 있으므로 연결 대상으로 설정되지 않도록 한다.

3. 사용자 요구사항 (usability requirement)

위젯 리스트 인터페이스에서 각 '위젯 파일'에 대한 정보를 제공하여 사용자가 어떤 기능을 수행하는 위젯인지 파악하기 쉽도록 한다.

사용자가 새로운 '위젯 파일'을 제작하는 경우, 위젯이 필수적으로 가져야 하는 라이브러리, 변수, 함수 등이 미리 작성되어 있는 Python 파일을 제공하여 사용자의 코딩 작업을 최소화한다.

4. 이식성 요구 사항 (portability requirement)

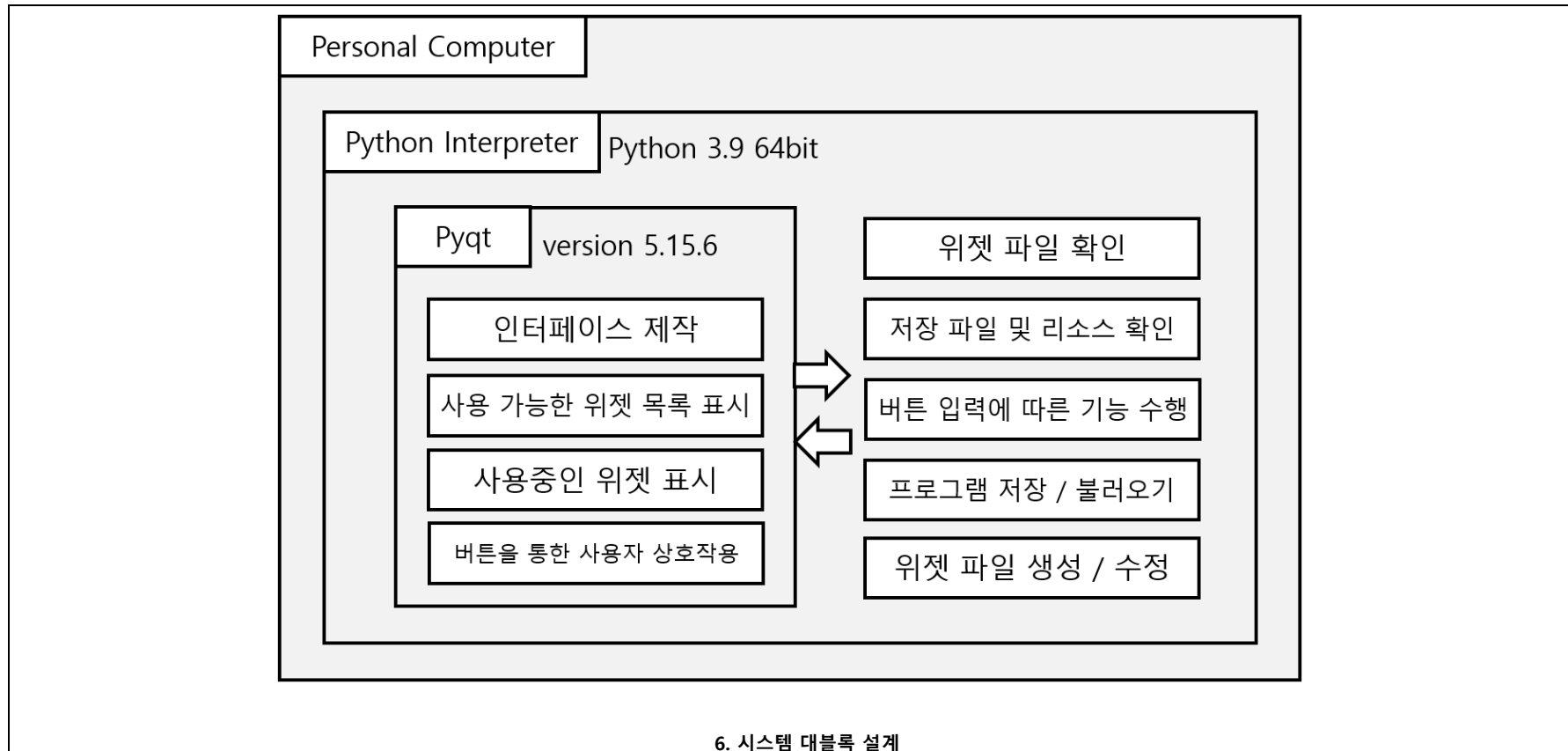
Python 기반으로 작성된 프로그램이므로, Python 이 설치되어 있는 PC 환경에서 사용할 수 있다.

5. 보안 요구 사항 (Security requirement)

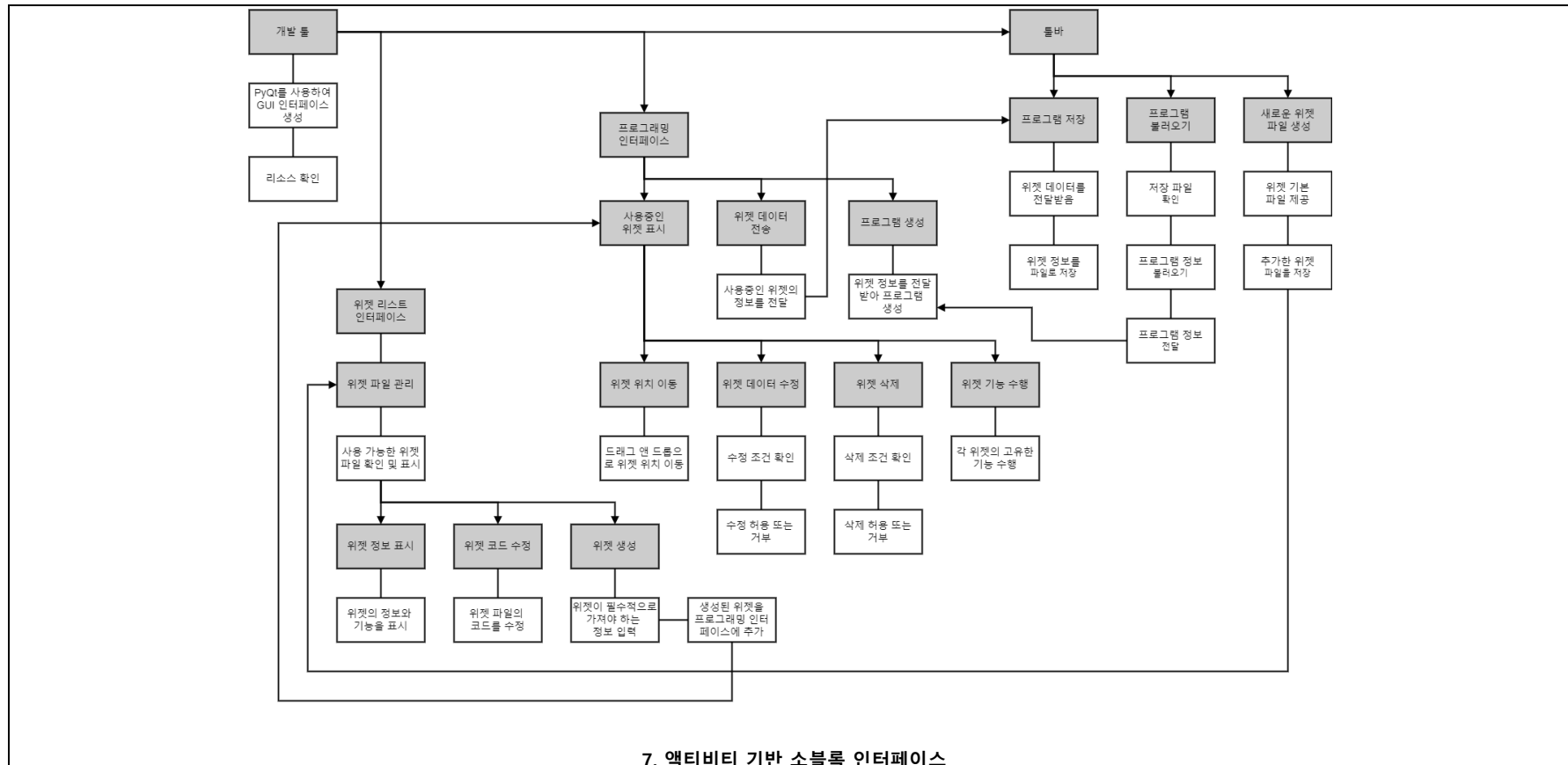
기본 리소스 파일이나 프로그램 저장 파일이 손상된 경우, 이를 인식하여 실행을 종료한다.

'위젯 파일'이 손상되거나 올바르지 않게 작성된 경우, 이를 인식하여 위젯이 사용되는 것을 방지한다.

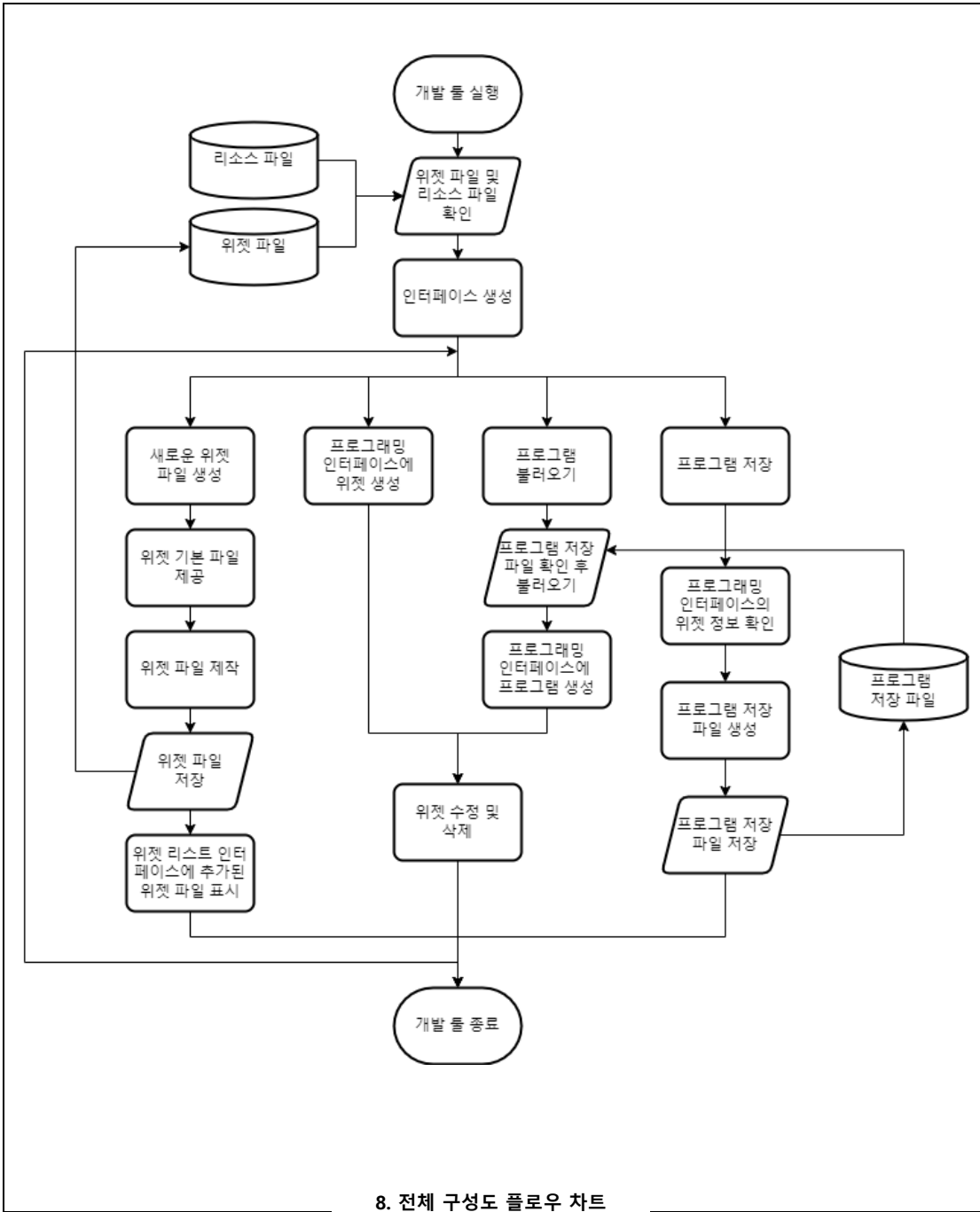
6. 시스템 대블록 설계

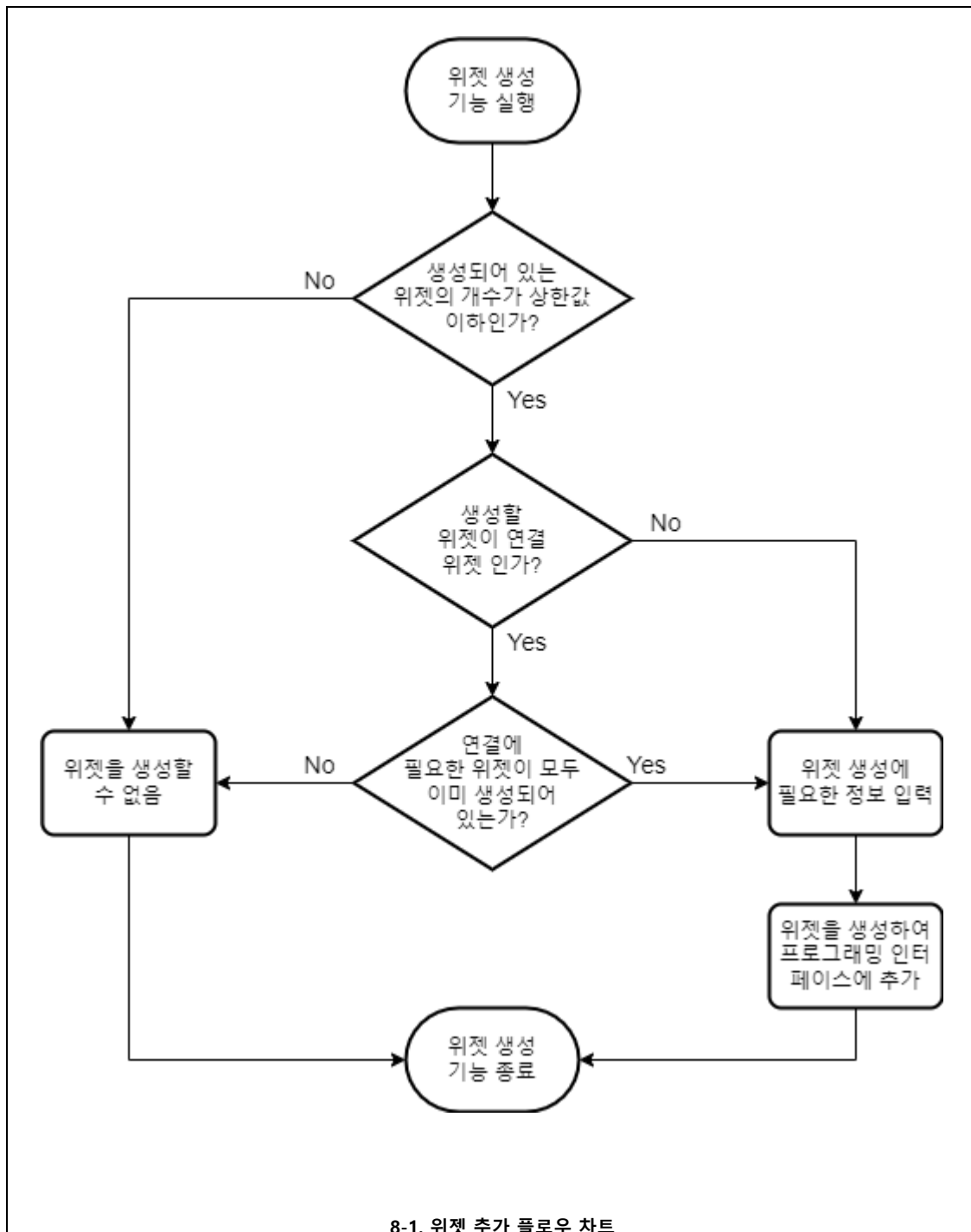


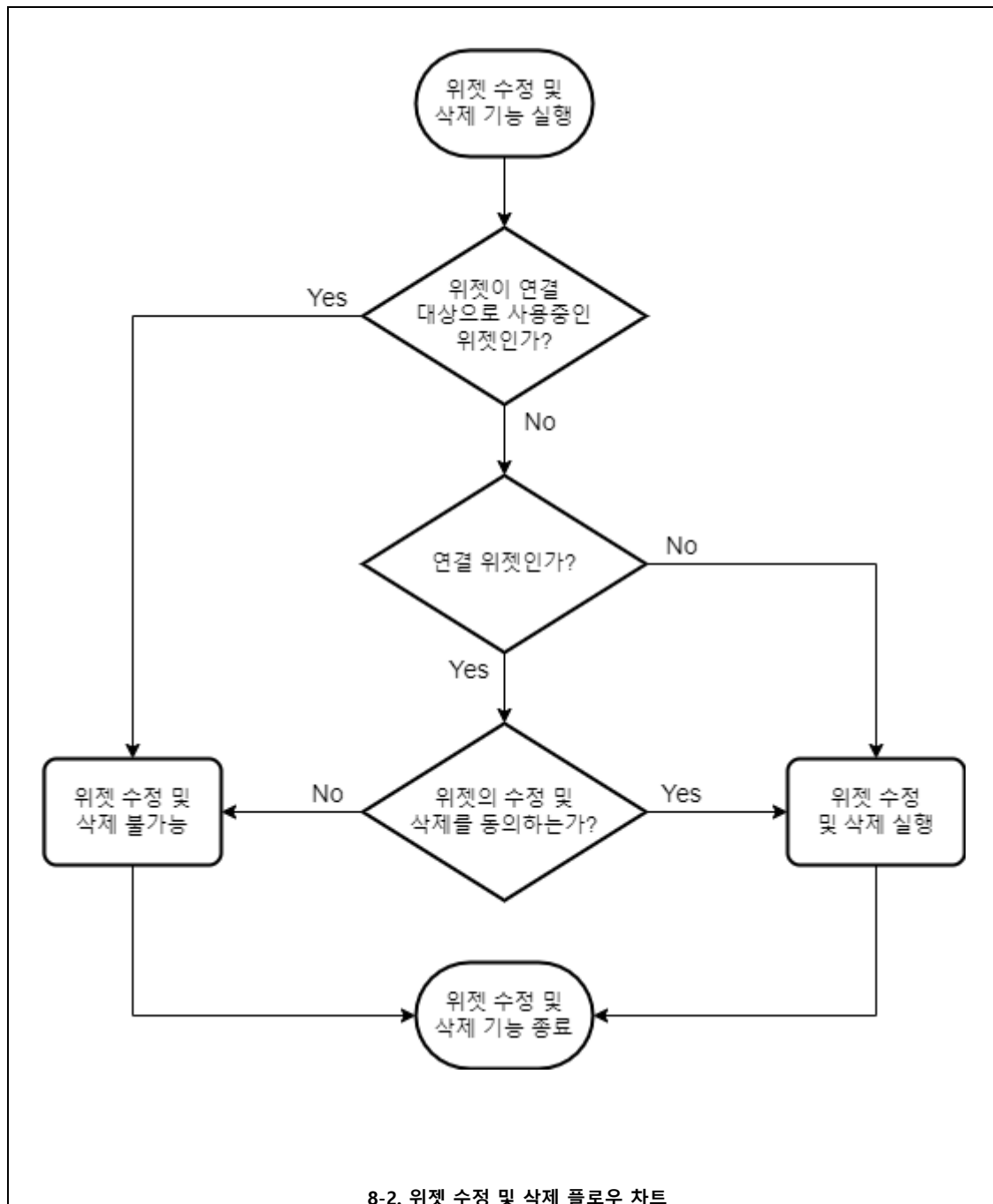
7. 액티비티 기반 소블록 인터페이스



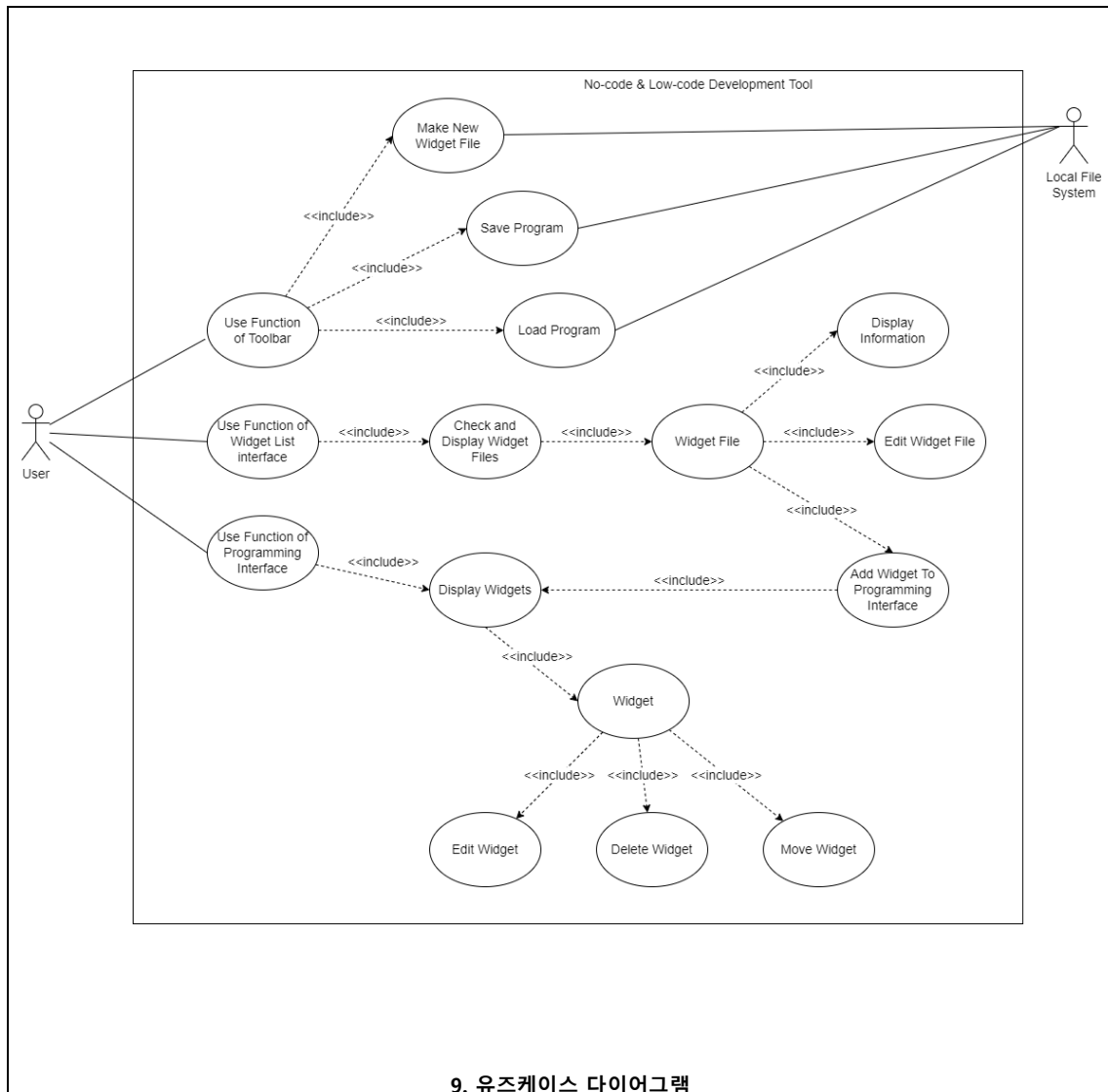
8. 구성도 (Flow Chart)



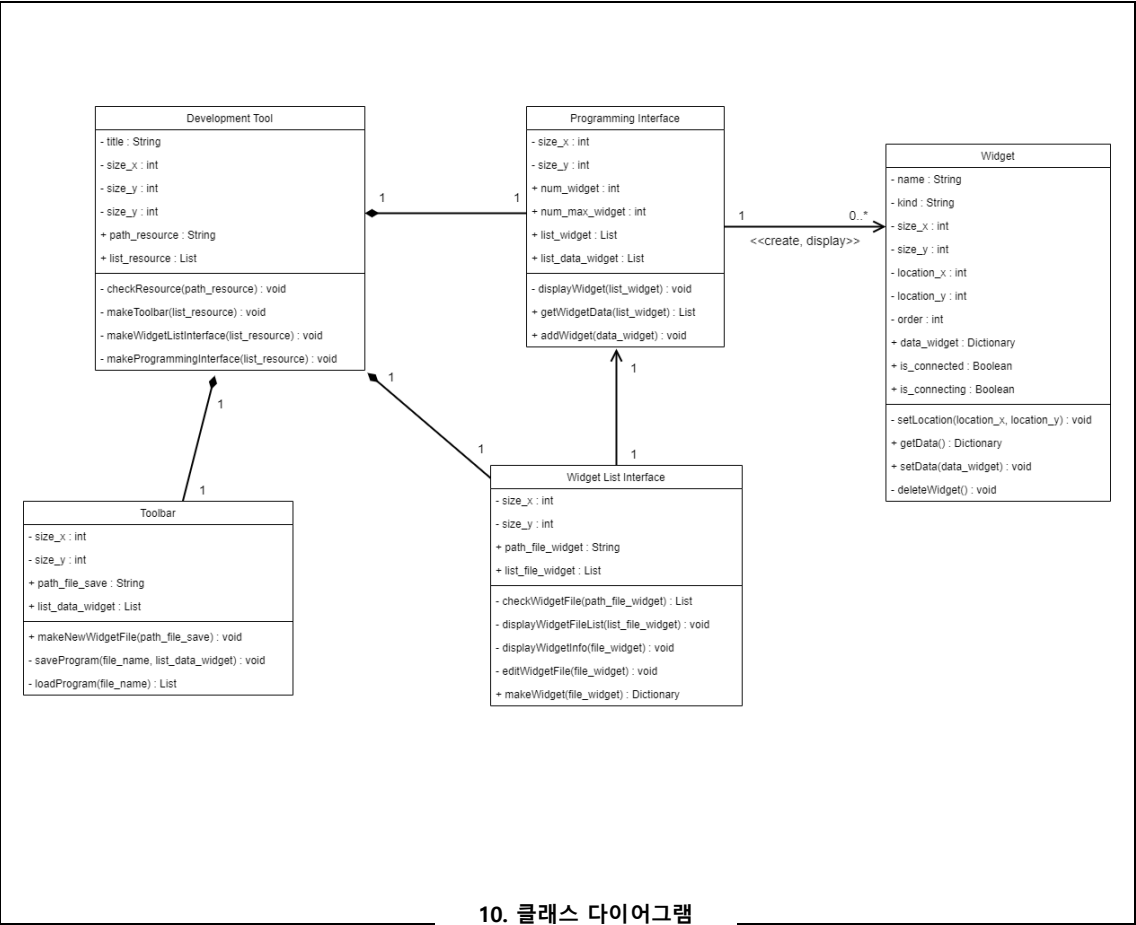




9. 유즈케이스 다이어그램 (Usecase Diagram)

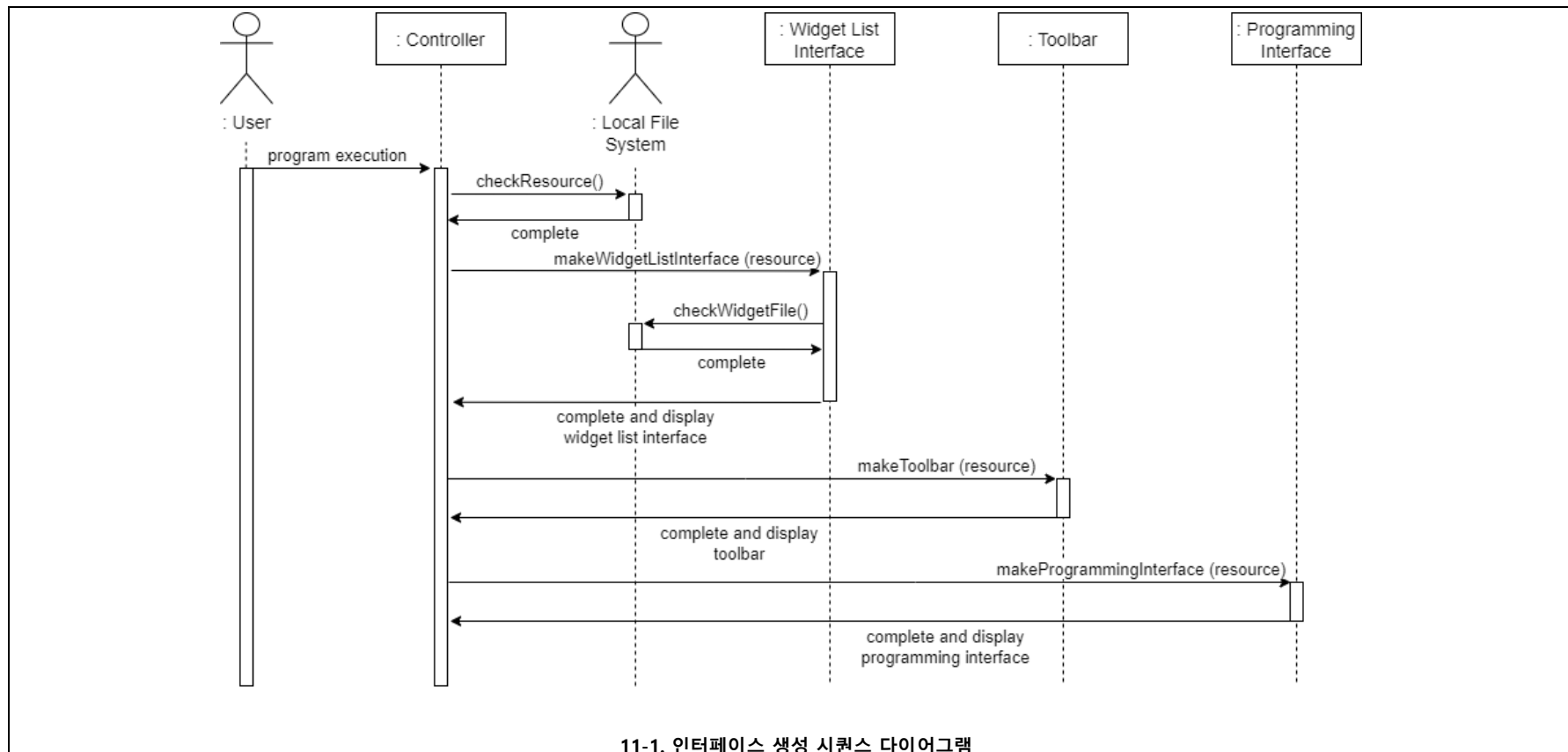


10. 클래스 다이어그램 (Class Diagram)

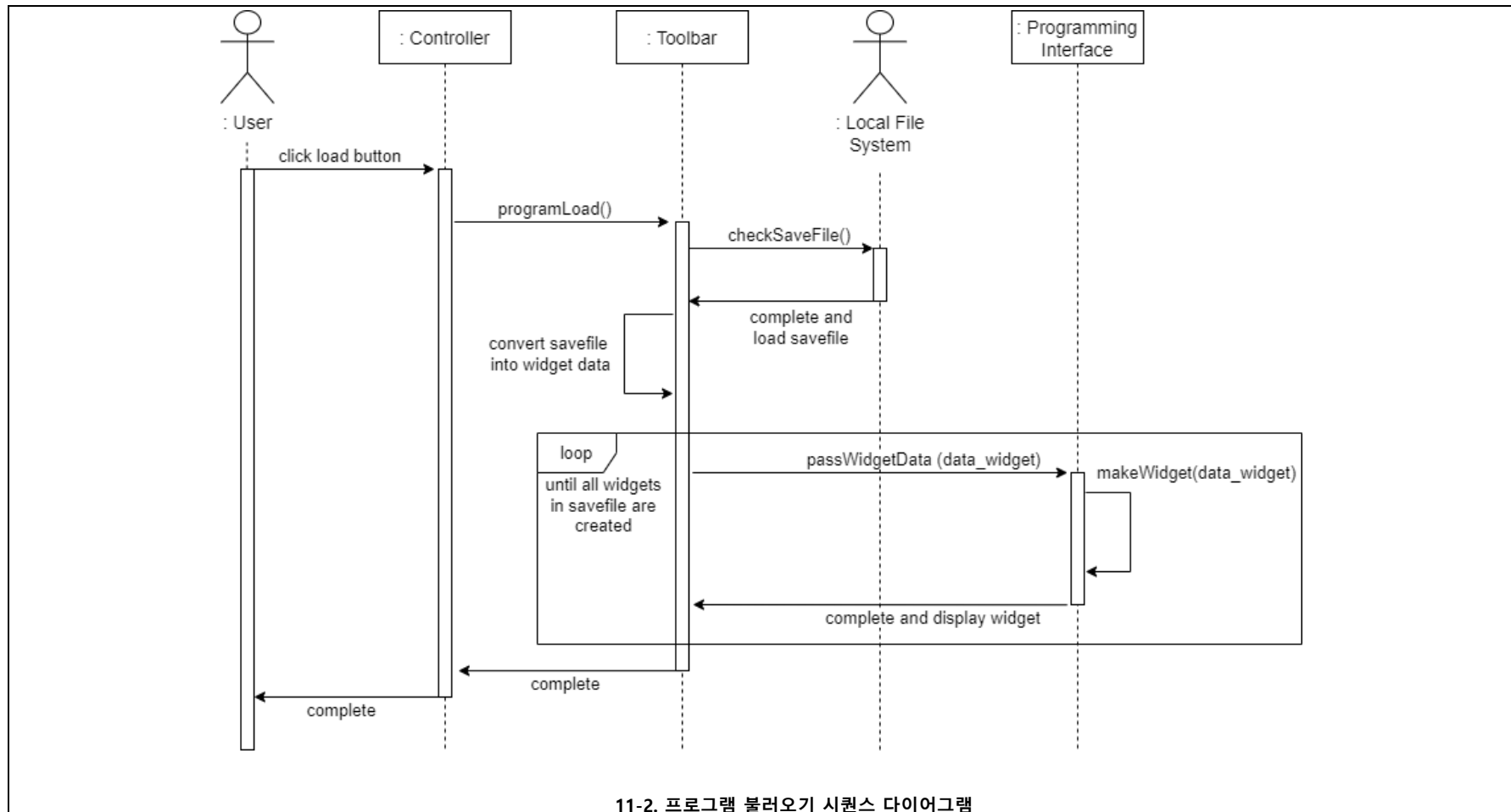


10. 클래스 다이어그램

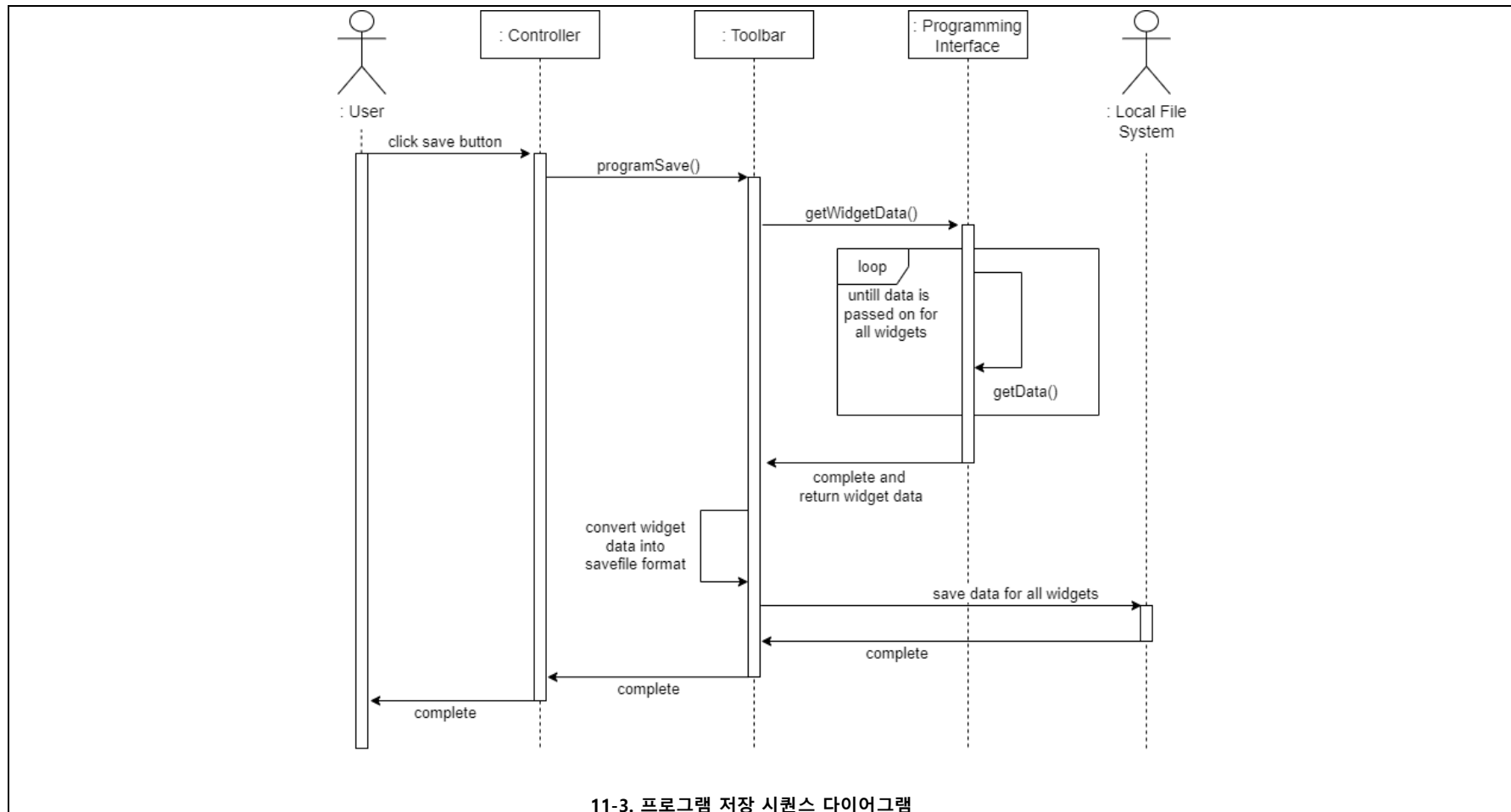
11. 시퀀스 다이어그램 (Sequence Diagram)



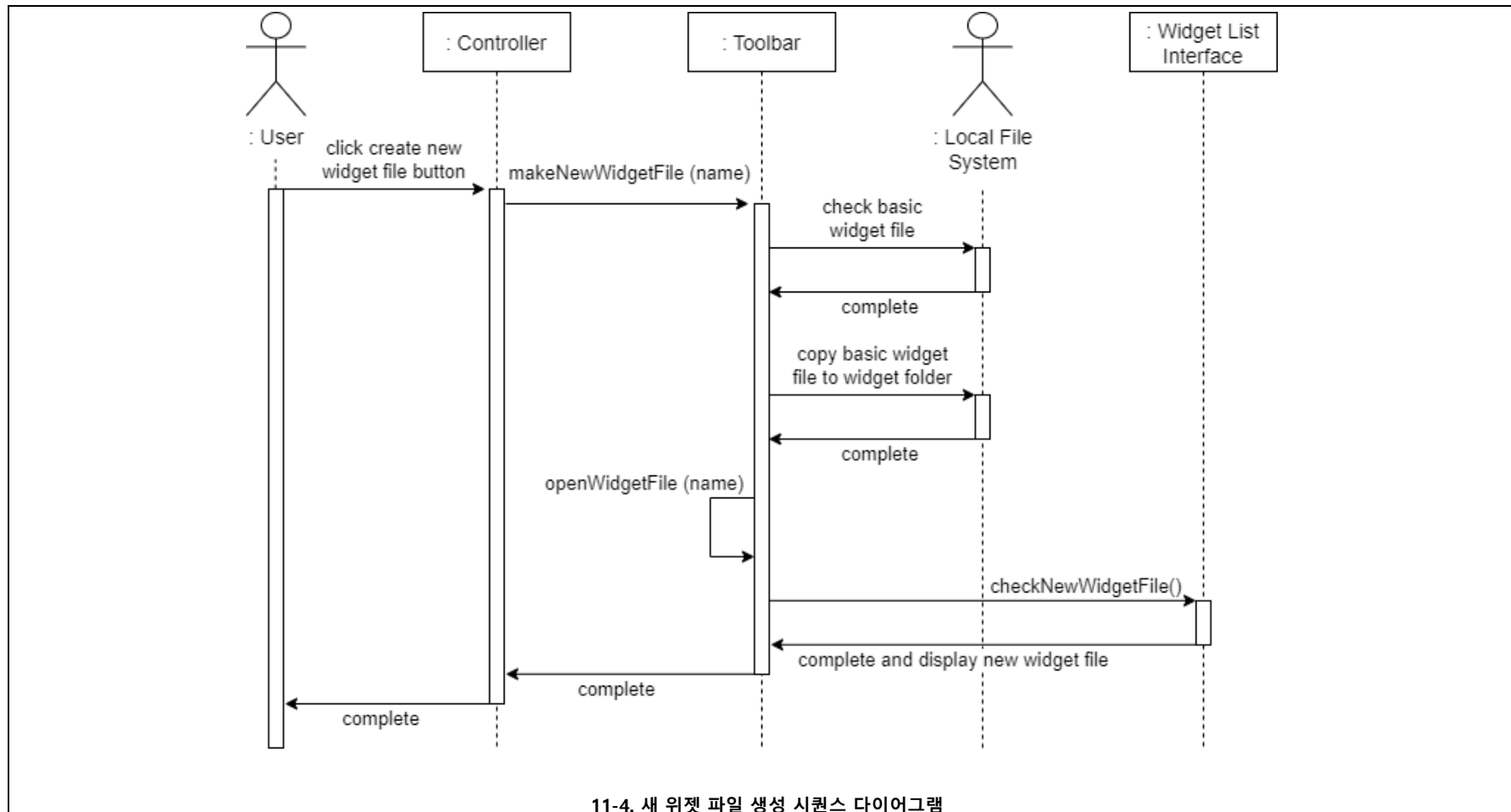
11-1. 인터페이스 생성 시퀀스 다이어그램

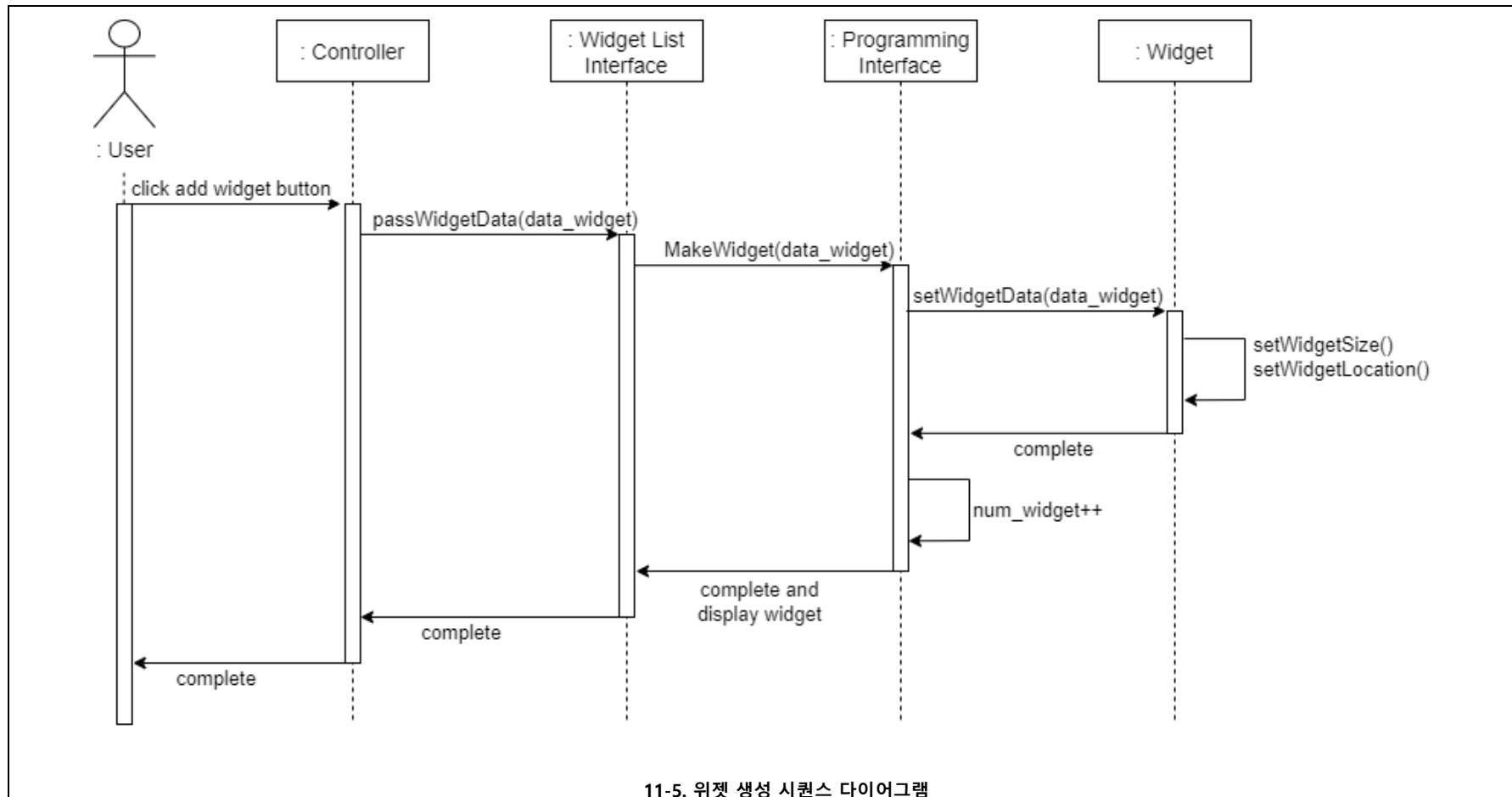


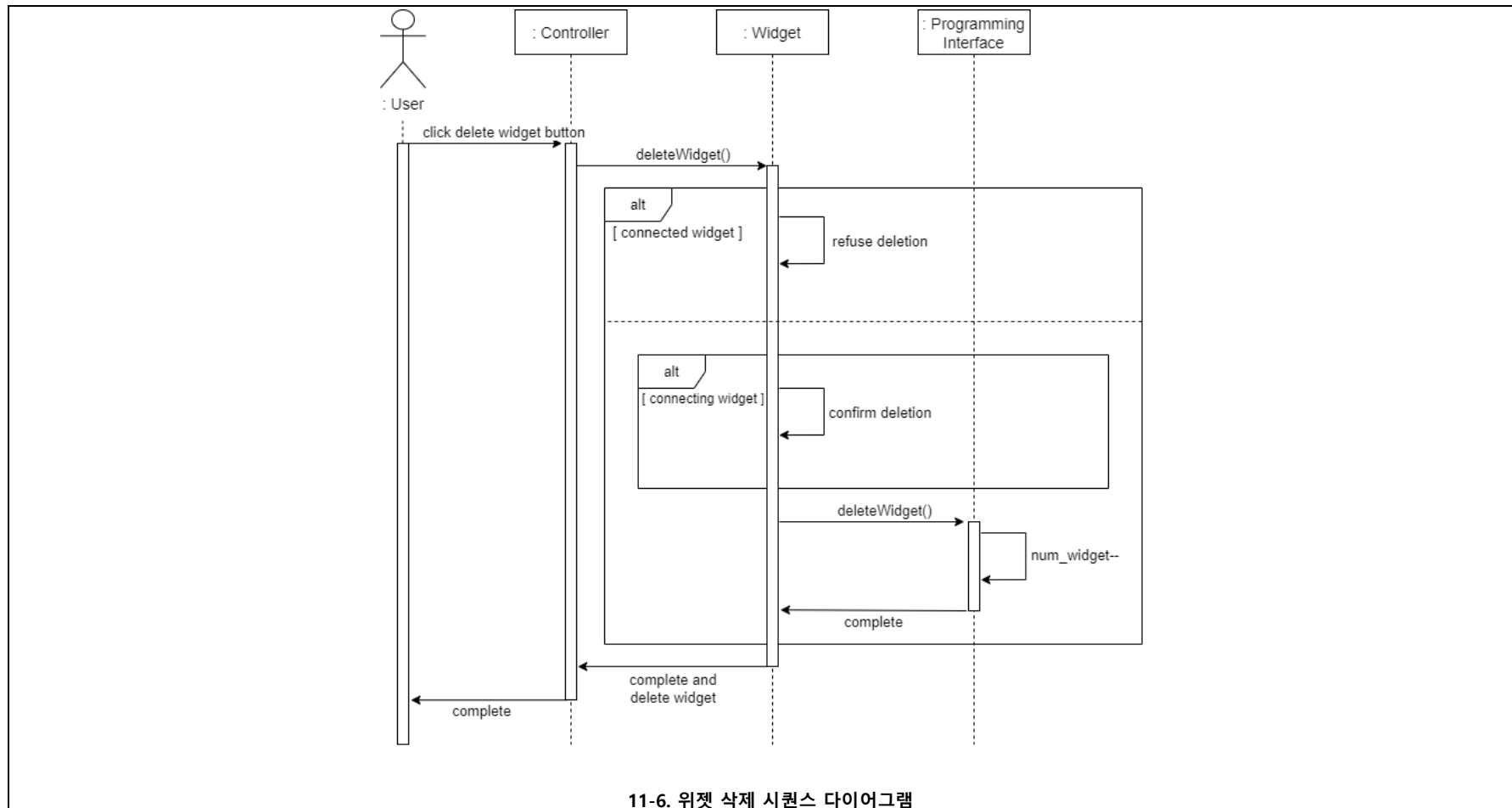
11-2. 프로그램 불러오기 시퀀스 다이어그램



11-3. 프로그램 저장 시퀀스 다이어그램







11-6. 위젯 삭제 시퀀스 다이어그램

12. 알고리즘 구성 (Algorithm)

프로그램 저장 알고리즘

1. 프로그래밍 인터페이스에서 현재 생성되어 있는 위젯의 목록을 확인한다.
2. 생성되어 있는 각 위젯의 정보를 얻어, 생성된 순서대로 List 형태로 만들어 저장한다.
3. 툴바로 위젯의 정보 List 를 전달하고 파일 이름을 입력 받는다. 이후 프로그램 저장 폴더에 저장 파일을 생성하고 데이터를 암호화하여 저장한다.

프로그램 불러오기 알고리즘

1. 불러올 프로그램 저장 파일의 이름을 선택한다. 프로그램 저장 폴더 위치에서 해당 이름을 가진 저장 파일을 탐색한다.
2. 저장 파일의 내용을 불러온 후, 암호화되어 저장된 내용을 복호화 하여 위젯의 데이터를 List 형태로 만든다.
3. 위젯의 데이터 List 를 프로그래밍 인터페이스에 전달한다. 프로그래밍 인터페이스는 기존에 생성되어 있던 위젯을 모두 제거하고, 전달받은 위젯 데이터 List 에 따라 순서대로 위젯을 생성하여 표시한다.

위젯 생성 알고리즘

1. 위젯 리스트 인터페이스에서 생성되어야 하는 위젯 파일을 확인한 후, 생성시 필수적으로 필요한 정보를 입력 받는다.
2. 입력 받은 위젯의 정보를 프로그래밍 인터페이스로 전달한다. 프로그래밍 인터페이스는 현재 생성된 위젯의 개수가 상한선 이하로, 위젯이 생성 가능한지 확인한다.
3. 위젯이 생성 가능한 경우, 전달받은 정보에 따라 위젯 객체를 순서에 맞게 생성하여 표시한다.

연결 위젯 생성 알고리즘

1. 위젯 리스트 인터페이스에서 생성되어야 하는 위젯 파일을 확인한 후, 생성시 필수적으로 필요한 정보를 입력 받는다. 또한 현재 생성되어 있는 위젯 중, 연결 대상으로 삼을 수 있는 위젯들을 확인하여 연결 대상으로 지정한다.
2. 연결 위젯의 정보를 프로그래밍 인터페이스로 전달한다. 프로그래밍 인터페이스는 현재 생성된 위젯의 개수가 상한선 이하로, 위젯이 생성 가능한지 확인한다.
3. 위젯 생성이 가능한 경우, 전달받은 정보에 따라 연결 위젯 객체를 생성하고 연결 대상으로 선택된 위젯을 연결시킨다. 위젯의 생성과 연결이 성공적으로 이루어진 경우, 위젯 객체를 표시한다.

13. 데이터 및 파일 시스템 구조

13-1. 데이터 구조

Python 은 key 와 value 의 쌍으로 이루어져 있는, Hash 형태의 데이터를 다루기 용이하도록 'Dictionary' 데이터 타입을 지원한다. Dictionary 타입은 key 와 value 가 일대일로 대응하며, key 를 통해 value 를 탐색할 수 있는 형태로, 한 Dictionary 타입에 여러 개의 key, value 쌍을 입력하여 사용할 수 있으며, 이러한 방식으로 데이터베이스 테이블에서의 레코드와 유사하게 사용할 수 있다.

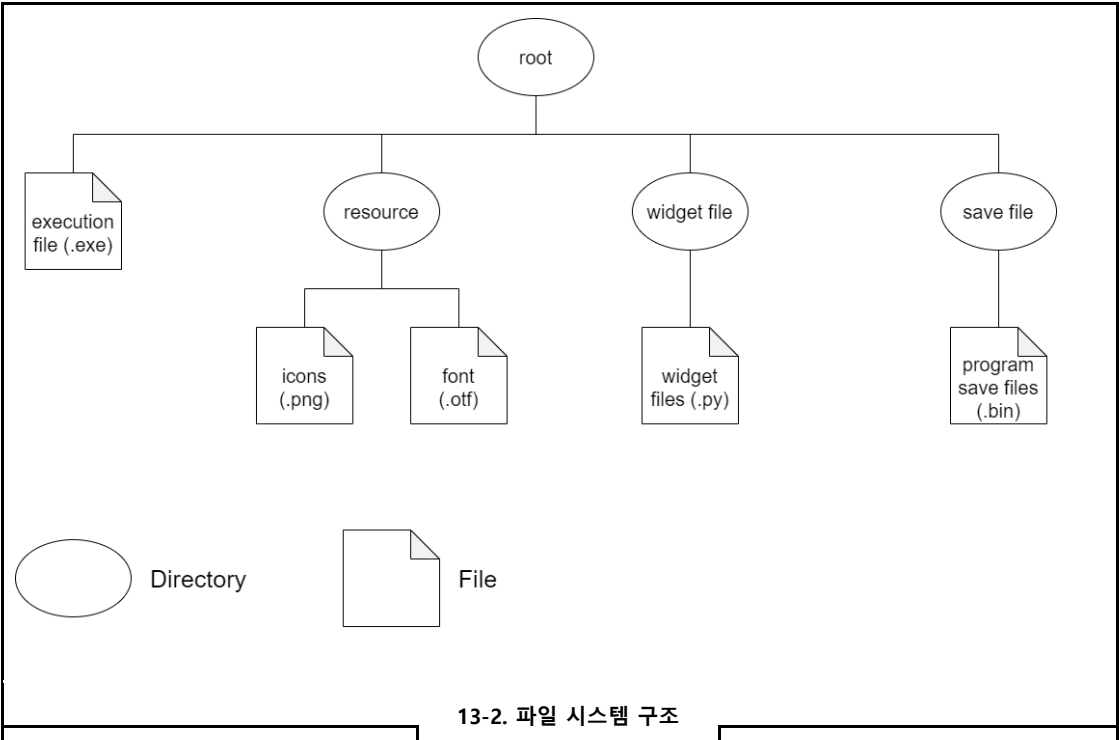
또한 Python 에서 지원하는 List 자료구조는 다른 프로그래밍 언어의 배열(Array)과 유사하게 사용할 수 있으며, 하위 타입에 제한이 없다. 따라서 Dictionary 타입을 인자로 가지는 List 는 데이터베이스 레코드를 모아 구성한 테이블과 유사하게 사용할 수 있다.

본 프로젝트에서는 위젯의 데이터를 담는 변수 `data_widget` 을 Dictionary 타입으로 설정하여, 위젯의 각 데이터를 key 와 value 쌍의 레코드와 유사한 형태로 사용한다. 또한 `list_data_widget` 변수는 `data_widget` 을 모아 List 타입으로 구성한 것으로, 현재 생성중인 위젯들의 데이터를 모아 놓은 테이블과 유사한 형태로 사용한다.

data_widget : Dictionary		
key 명칭	value 타입	비고
name	String	위젯의 이름
kind	String	위젯의 종류, 위젯 파일의 이름
location_x	int	위젯의 프로그래밍 인터페이스 상 위치의 x 좌표
location_y	int	위젯의 프로그래밍 인터페이스 상 위치의 y 좌표
order	int	위젯이 생성된 순서
is_connected	Boolean	해당 위젯이 연결 위젯으로 사용중인지 확인
is_connecting	Boolean	해당 위젯이 연결 위젯인지 확인

13-1. data_widget 구조

13-2. 파일 시스템 구조



14. 시나리오 (Scenario)

[사용자 위젯 생성]

1. 위젯 추가

툴바에서 '새로운 위젯을 추가' 버튼을 선택 후, Python 코딩을 통해 '위젯 파일'을 작성하여 좌측 위젯 리스트 인터페이스에 추가

2. 위젯 선택

위젯 리스트에서 원하는 위젯을 선택

3. 사용

사용자가 선택한 '위젯'의 기능이 우측 프로그래밍 인터페이스에 나타나게 되며, 만들어진 위젯 추가 후 기능 수행

[위젯 사용]

대표 예시)

메모장-우측 프로그래밍 인터페이스에 메모장 추가 후 사용

1. 데이터 수정

데이터 수정 버튼을 사용하여 '위젯'의 기존에 입력해놓은 데이터를 수정할 수 있다

2. 삭제

사용자가 삭제 버튼을 선택하면, 해당 위젯이 단독으로 존재하는 위젯인지, 조합된 위젯이 존재하는 '연결 위젯'인지 조건을 검사하여 판단 후 삭제한다.

3. 이동

프로그래밍 인터페이스에서 위젯 위치 이동을 원할 경우 이동 버튼을 눌러 드래그 앤 드롭 형태로 사용 가능하다.

[프로그램 저장, 불러오기]

1. 저장 버튼을 통해 현재 프로그래밍 인터페이스에 사용중인 위젯들의 정보 저장
2. 저장된 프로그램 정보가 있는 경우 불러오기 버튼을 통해 프로그램 정보를 가져옴

[연결 위젯을 통한 위젯 조합 사용]

1. 생성하고자 하는 연결 위젯이 필요로 하는 위젯을 우측 프로그래밍 인터페이스에 추가
2. '연결 위젯'을 생성하여 프로그래밍 인터페이스의 위젯 연결 후 사용자의 의도에 따라 이용 가능

15. 역할 분담 및 개발 일정

		개발구분	3월				4월				5월				6월			
			1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
기획	김동현, 장혜진	아이디어 제안&구체화																
	김동현, 장혜진	기술 분석 및 자료																
	장혜진	use case diagram&flow chart 제작																
	김동현	최종제안서 보완 및 정리																
개발	김동현, 장혜진	기본 인터페이스 구성하기																
	장혜진	위젯리스트 인터페이스 구현																
	김동현	프로그래밍 인터페이스 구현																
	김동현, 장혜진	위젯 추가기능 구현																
최종결과	김동현, 장혜진	최종 테스트																
	김동현, 장혜진	수정																
	김동현, 장혜진	최종보고서 제출																

15. 역할 분담 및 개발 일정

16. 관련 기술 및 개념

- No-code

간단한 사용자 인터페이스 방식을 제공하여 복잡한 코딩과정 없이 응용프로그램들을 구축할 수 있도록 하는 방식으로 SW 자동화 기술을 이용하여 코딩 없이 소프트웨어를 구축할 수 있는 것을 일컫는다.

자체적인 어플리케이션을 구축하고자 하는 비전문가 사용자들에게 유망한 대안책으로 떠오르는 기술이다.

- Low-code

복잡한 코딩 과정을 단순화해서 소프트웨어를 빠르게 개발 및 배포하도록 만든 일종의 개발 환경으로 코드의 작성을 최소화하여 필요한 기능을 간단한 코딩 작업만으로 구현하여 소프트웨어를 개발하는 것을 말한다.

코딩이 한 줄도 필요하지 않는 No-code와는 다르게 코딩이 필요하지만, 이에 대한 가이드라인이 존재하는 경우가 많다.

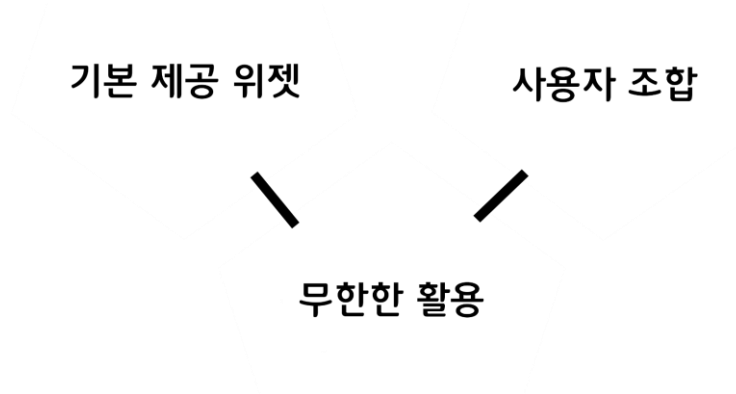
- PyQt

Qt Company 에서 개발한 오픈소스 크로스 플랫폼 C++ GUI 프로그래밍 라이브러리인 Qt 를 Riverbank Computing Limited 에서 Python 환경에서 실행되도록 바인딩한 라이브러리로, 1000 여개의 GUI 프로그래밍 클래스를 지원한다. 라이선스로는 GPL 과 동시에 상업적 사용이 가능한 라이선스를 판매 중이므로 사용자의 목적에 맞게 라이선스를 선택하여 사용할 수 있다.

17. 기대효과

- 프로그래밍 입문자, 비전공자들이 배경지식의 부담감 없이 GUI 프로그래밍을 익숙한 PC 환경에서 접할 수 있다.
- 모바일 앱, 웹사이트, 비즈니스 등에만 특화 되어있는 기존의 No-code, Low-code 플랫폼을 익숙한 PC 플랫폼을 통해 일반인들에게도 제공할 수 있다.
- 쉽고, 간단하고, 효율적인 GUI 프로그래밍 툴을 통해 복잡하지 않다는 인식을 심어주고, 사용자들에게 프로그램 개발에 대한 동기부여와 의욕 고취를 기대 할 수 있다.

- 사용자가 개인적 기호에 맞게 프로그램을 조합하여 새로운 프로그램을 만드는 방식을 통해 사용자의 다양한 욕구를 만족시킬 수 있다.



18. 개발도구 및 개발환경

개발도구

- Visual Studio 2019 v16.11.13
 - Microsoft 에서 개발한 통합 개발 환경으로, C++, C#, Visual Basic, Python 등 수많은 언어를 지원한다. 또한 사용 목적에 따라 기능과 가격을 나누어 개인 사용자용 Community 버전, 전문가용 Professional 버전, 기업용 Enterprise 버전을 지원한다. 본 프로젝트에서는 Python 프로그래밍 도구로 Visual Studio 2019 를 활용하며, 제안서 작성 시점 상의 최신 버전인 16.11.13 Community 버전을 사용한다.

개발환경

- OS : Microsoft Windows 10.0 - 64 bit
 - Microsoft 에서 개발한 운영체제로, 현재 가장 광범위하게 사용되는 운영체제이다. 프로그래밍 비전공자와 입문자를 위한 No-code & Low-code 프로그래밍 툴을 만드는 것이 본 프로젝트의 목적인 만큼, 비전공자와 입문자가 주로 사용하는 OS 인 Windows 10.0 의 64bit 버전을 본 프로젝트의 개발 환경으로 설정한다.
- Python 3.9.7 - 64 bit
 - Python 은 1991 년 귀도 반 로섬이 개발한 인터프리터 방식의 프로그래밍 언어이다. 문법이 간편하고 다양한 자료구조와 라이브러리를 지원하여 어플리케이션 개발, 게임 제작, 머신 러닝 등 수많은 분야에 활용된다. 본 프로젝트에서는 Python 특유의 편의성과 확장성, 그리고 많은 교육기관에서 프로그래밍 교육을 Python 으로 시작한다는 것에 주목하여, 개발 프로그래밍 언어로 Python 을 사용한다.

19. 참고문헌

- 1) Codebots | The pros and cons of no code for software development. <<https://codebots.com/low-code/what-is-no-code-the-pros-and-cons-of-no-code-for-software-development>>
- 2) Python Wiki | PyQt <<https://wiki.python.org/moin/PyQt>>
- 3) Riverbank Computing | PyQt Introduction
<<https://riverbankcomputing.com/software/pyqt/intro>>
- 4) Visual Studio 2019 버전 16.11 릴리스 정보 | Microsoft Build
<https://docs.microsoft.com/ko-kr/visualstudio/releases/2019/release-notes>
- 5) Python about <https://www.python.org/about/>