소프트웨어 요구사항 명세서

<베이직 노트>

* 메모 프로그램 -

|  |  |
| --- | --- |
| 과 목 명 | 소프트웨어 공학 |
| 담당교수 | 강동현 |
| 학 과 | 국어국문학과, 컴퓨터 공학과 |
| 조 명 | D조 |
| 조 원 | 우지연 권기영 김종민 김호진 정민수 |
| 문서 버전 | 1.7 |
| 제 출 일 | 2021.11.20 |



목차

[1. 서문 2](#_Toc88214438)

[**1.1** **설계 목적** 2](#_Toc88214439)

[**1.2** **유사 소프트웨어** 2](#_Toc88214440)

[**1.3** **차이점** 3](#_Toc88214441)

[**1.4** **기대 효과** 3](#_Toc88214442)

[**1.5** **예상 사용자 계층** 3](#_Toc88214443)

[2. 도입 4](#_Toc88214444)

[**2.1** **시스템 기능** 4](#_Toc88214445)

[**2.2** **실행환경** 4](#_Toc88214446)

[**2.3** **설계 및 실행 제약사항** 4](#_Toc88214447)

[**2.4** **가정 및 의존 사항** 5](#_Toc88214448)

[**2.4.1** **가정** 5](#_Toc88214449)

[**2.4.2** **의존 사항** 6](#_Toc88214450)

[3. 구조적 명세 6](#_Toc88214451)

[**3.1 구조적 명세** 6](#_Toc88214452)

[4. 사용자 요구사항 정의 9](#_Toc88214453)

[**4.1 요구하는 소프트웨어 구상도** 9](#_Toc88214454)

[**4.2 기능적 요구사항** 9](#_Toc88214455)

[**4.3 비기능적 요구사항** 10](#_Toc88214456)

[5. Use Case Diagram 10](#_Toc88214457)

[**5.1 Use Case Diagram** 10](#_Toc88214458)

[**5.2 Use Case Description** 11](#_Toc88214459)

[6. Class Diagram 14](#_Toc88214460)

[**6.1 Class Diagram** 14](#_Toc88214461)

[**6.2 Class Diagram Description** 14](#_Toc88214462)

[7. 참조 22](#_Toc88214463)

그림 목차

[[그림 1‑1] 노션 실행 화면 2](#_Toc88214464)

[[그림 4‑1] 요구하는 소프트웨어 구상도 9](#_Toc88214465)

[[그림 5‑1] ‘베이직 노트’의 Use Case Diagram 10](#_Toc88214466)

[[그림 6‑1] ‘베이직 노트’의 Class Diagram 14](#_Toc88214467)

표 목차

[[표 1‑1] ‘노션’과 ‘베이직 노트’의 차이점 3](#_Toc88214468)

[[표 5‑1] 페이지 수정 11](#_Toc88214469)

[[표 5‑2] 글자 크기 변경 11](#_Toc88214470)

[[표 5‑3] 페이지 생성 11](#_Toc88214471)

[[표 5‑4] 페이지 저장 12](#_Toc88214472)

[[표 5‑5] 페이지 명 중복 검사 12](#_Toc88214473)

[[표 5‑6] 페이지 목록 12](#_Toc88214474)

[[표 5‑7] 페이지 삭제 13](#_Toc88214475)

[[표 5‑8] 페이지 열기 13](#_Toc88214476)

[[표 6‑1] Main 클래스의 변수 15](#_Toc88214477)

[[표 6‑2] Main 클래스의 메서드 15](#_Toc88214478)

[[표 6‑3] ViewController 클래스의 변수 15](#_Toc88214479)

[[표 6‑4] ViewController 클래스의 메서드 16](#_Toc88214480)

[[표 6‑5] PageMenu 클래스의 변수 16](#_Toc88214481)

[[표 6‑6] PageMenu 클래스의 메서드 17](#_Toc88214482)

[[표 6‑7] PageListViewer 클래스의 변수 18](#_Toc88214483)

[[표 6‑8] PageListViewer 클래스의 메서드 18](#_Toc88214484)

[[표 6‑9] PageList 클래스의 변수 18](#_Toc88214485)

[[표 6‑10] PageList 클래스의 메서드 19](#_Toc88214486)

[[표 6‑11] Page 클래스의 변수 20](#_Toc88214487)

[[표 6‑12] Page 클래스의 메서드 20](#_Toc88214488)

[[표 6‑13] PageEditor 클래스의 변수 20](#_Toc88214489)

[[표 6‑14] PageEditor 클래스의 메서드 21](#_Toc88214490)

개정 이력

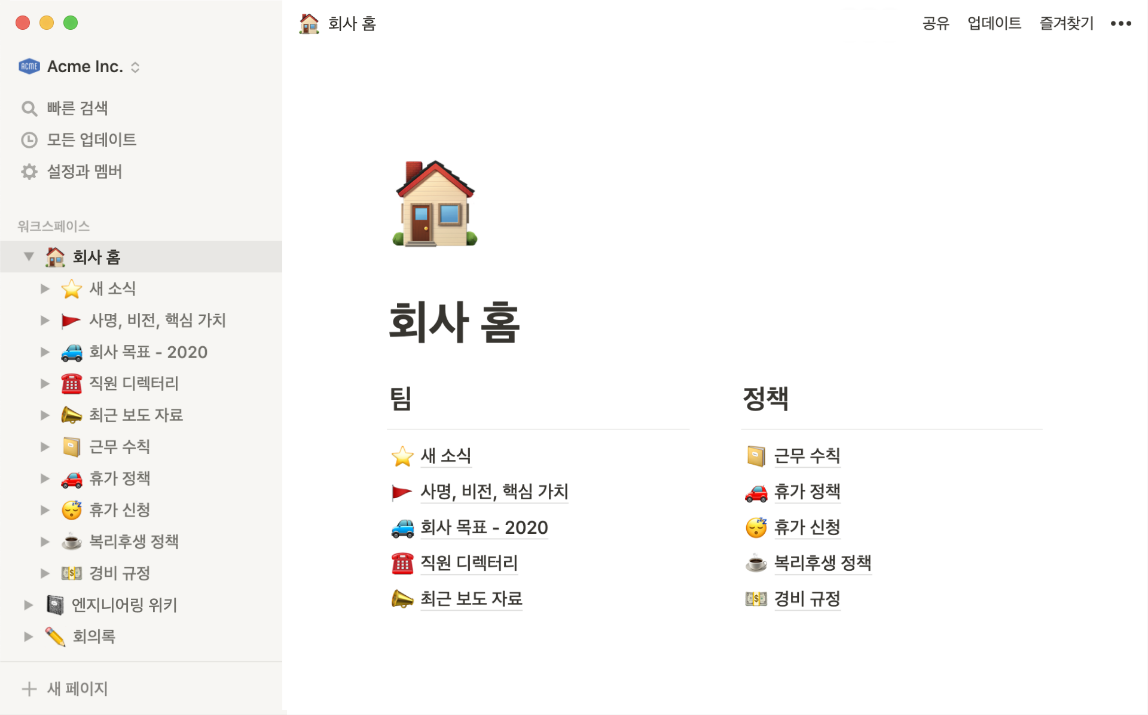
|  |  |  |
| --- | --- | --- |
| **날짜** | **변경이유** | **Version** |
| 10.16.21 | 서문, 도입, 사용자 요구사항 정의가 작성되었다. | 1.0 |
| 11.05.21 | Use Case 작성 및 사용자 요구사항이 변경되었다. | 1.1 |
| 11.08.21 | 유사 소프트웨어와의 차이점 및 기대효과가 추가로 작성되었다. | 1.2 |
| 11.10.21 | 서문의 기대효과 및 도입이 수정되었다. | 1.3 |
| 11.14.21 | 설계 목적, 차이점 및 Use Case Diagram이 수정되었다. | 1.4 |
| 11.16.21 | 설계 목적과 기대효과가 수정되었다. | 1.5 |
| 11.18.21 | Class Diagram이 작성되었다. | 1.6 |
| 11.20.21 | 사용자 요구사항에 기능적,비기능적 요구사항이 추가되었다. | 1.7 |

# **서문**

## **설계 목적**

현대인의 업무량이 늘어남에 따라 메모의 필요성은 지속적으로 강조되고 있으며, 메모를 위한 다양한 컴퓨터 프로그램들이 존재한다. 대표적인 메모 프로그램인 '노션'은 이미지 삽입, 일정 관리와 같은 다양한 기능을 제공한다. 그러나, 지나치게 다양한 기능은 간단한 텍스트 기록만을 원하는 사용자에게 불필요하며, 오히려 프로그램 사용을 어렵게 하는 원인이 되기도 한다. 본 프로젝트에서 개발할 '베이직 노트'는 메모에 필수적인 텍스트 작성 기능에 트리뷰(Tree view) 기능을 추가하여 페이지 관리를 간편하게 하고자 한다.

## **유사 소프트웨어**



[그림 1‑1] 노션 실행 화면

[그림 1-1]은 ‘노션(Notion)’ 프로그램의 실행 화면이다. ‘노션’은 노트 프로그램이자 프로젝트 관리 소프트웨어로, 업무 관리, 프로젝트 관리, 메모를 위해 사용되는 프로그램이다. ‘노션’은 일정표, 갤러리, 캘린더, 칸반 보드 등의 기능과 영상, 음성, 코드와 같은 다양한 양식의 파일 업로드를 지원한다. ‘노션’에서 생성한 페이지는 Markdown이나 HTML 형태로 내보낼 수 있다.

## **차이점**

‘베이직 노트’는 메모를 위한 텍스트 작성 기능만을 제공하여 사용자가 쉽고 간편하게 기록할 수 있도록 도와주는 프로그램이며, ‘노션’은 웹 기반의 소프트웨어로 서버 동기화를 통해 메모를 저장하는 복합적인 프로그램이다. [표 1-1]은 ‘노션’과 ‘베이직 노트’의 차이점을 보여준다.

[표 ‑] ‘노션’과 ‘베이직 노트’의 차이점

|  |  |  |
| --- | --- | --- |
|  | **노션** | **베이직 노트** |
| **서비스 형태** | 웹 기반 서비스 | Stand-alone 서비스 |
| **저장 방식** | 서버 동기화 이용 | 파일 시스템 이용 |
| **기능** | 복합적인 기능 | 텍스트 작성 기능 |

## **기대 효과**

해당 프로그램은 다음과 같은 기대효과를 가진다.

* 네트워크 트래픽을 유발하지 않기 때문에 동시에 동작하는 **다른 온라인 서비스의 속도에 영향을 미치지 않는다**.
* 프로그램 실행 시 CPU와 메모리 자원을 적게 소모하므로 **다중작업 환경에서 유리**하다.
* 민감한 파일을 제3자의 서버에 저장하지 않으므로 **보안적인 측면에서 유리**하다.
* 간단한 기능만을 제공하는 사용자 친화적인 프로그램이므로, 프로그램 적응을 위한 **시간적 비용이 감소**한다.

## **예상 사용자 계층**

해당 프로그램은 다음과 같은 계층에 의해 사용될 것으로 예상된다.

* 오프라인 환경에서 메모 프로그램이 필요한 사용자 계층.
* 텍스트 작성 기능만 필요로 하는 사용자 계층.
* 프로그램 적응에 많은 시간을 소비하고 싶지 않은 사용자 계층.
* 컴퓨터에 대한 배경지식이 풍부하지 않은 사용자 계층.

# **도입**

## **시스템 기능**

다음은 ‘베이직 노트’의 시스템 기능에 대한 설명이다.

* 페이지 수정: 사용자가 생성된 페이지에 글을 추가, 삭제하여 데이터를 작성하고, 사용자가 작성한 데이터를 ‘베이직 노트’가 임시로 저장하는 기능.
* 글 작성 및 크기 변경: 사용자가 생성한 페이지에 글을 작성할 수 있고, 페이지에 작성한 글자 크기를 사용자가 지정한 만큼 변경하는 기능.
* 페이지 생성: 사용자가 빈 페이지를 생성하는 기능.
* 페이지 저장: 사용자가 작성한 페이지의 이름을 지정하여 txt 파일로 저장하는 기능.
* 페이지 명 중복 검사: 사용자가 저장하려는 페이지 명이 존재하는지 검사하는 기능.
* 페이지 삭제: 사용자가 작성한 페이지를 삭제하는 기능.
* 페이지 목록: 사용자가 저장한 페이지 명을 보여주는 기능.
* 페이지 열기: 사용자가 선택한 페이지의 내용을 보여주는 기능.

## **실행환경**

해당 프로그램의 실행환경은 다음과 같이 설정한다.

* Window 10
* Java JRE 17

## **설계 및 실행 제약사항**

해당 프로그램은 다음과 같은 제약사항을 가진다.

* 사용자에게 친화적인 인터페이스를 구현해야 하고, CLI로 개발해서는 안 된다.
* Java SE 17 문법을 표준으로 프로그램이 작성된다.
* 시스템은 객체 모델링 기법으로 설계하며 설계 과정에서 나온 산출물은 UML로 기록한다.
* 소프트웨어 실행 시 5초 내에 창이 나타나야 하고, 모든 기능은 1초 내에 수행되어야 한다.
* 문서를 저장하면 실제 디스크에 바로 저장되어야 하며, 파일을 다시 불러올 수 있어야 한다.
* 창의 크기는 사용자가 손쉽게 조절할 수 있어야 한다. 이때, 텍스트 상자의 크기는 창의 크기에 따라 자동으로 바뀌어야 한다.
* 시스템 내 기능으로 문서 삭제 시 복구할 수 없다.
* 파일 저장 시 확장자는 '.txt'로 한다.
* 사용자가 프로그램을 실행하는 경로에 save\_file 폴더를 만들고, 저장한 페이지는 해당 폴더에 저장된다.
* 글자 크기는 최소 1pt, 최대 50pt로 제한한다.

## **가정 및 의존 사항**

### **가정**

해당 프로그램은 다음과 같은 가정 아래 설계된다.

* 요구 명세서가 설명하는 소프트웨어는 프로그램 및 시스템으로 설명될 수 있으며 세 단어는 같은 소프트웨어를 의미한다.
* Java에서 제공하는 함수를 제외한 모든 함수는 저작권 문제를 피하기 위해 새로 작성한다. 다만 블로그와 GitHub 상의 코드를 참고할 수는 있다.
* 글자 크기는 사용자가 변경할 수 있고, 한 문서에서 글자간 크기가 다를 수 있다.
* 창의 크기를 변경하여도 글자 크기는 변하지 않으며, 마찬가지로 글자 크기를 변경해도 창의 크기는 변하지 않는다.

### **의존 사항**

해당 프로그램은 다음과 같은 의존 사항을 가진다.

* Java SE 17 표준으로 작성되며 이전, 이후 표준에서 완벽하게 동작하지 않을 수 있다.
* Windows 10 기준에서 컴포넌트 색상이 결정되며, 타 운영체제에서는 색상이 다를 수 있다.

# **구조적 명세**

## **3.1 구조적 명세**

그냥 요구사항으로 큰 목차하고 3.1 구조적명세 3.2 기능적 요구사항 3.3비기능적 요구사항이 좋을까요?

|  |  |
| --- | --- |
| **기능** | 페이지 생성 |
| **설명** | 사용자가 글 작성을 목적으로 페이지를 생성한다. |
| **입력** | 사용자가 페이지 생성 버튼 클릭 |
| **출처** | 메인 화면 |
| **출력** | 빈 페이지 |
| **목적지** | 메인 페이지? 로컬? |
| **동작** | 빈 페이지를 메인 화면에 출력한다. |
| **필요사항** | 사용자가 버튼을 클릭한다. |
| **사전조건** | 없음 |
| **사후조건** | 메인 화면이 글을 작성할 수 있는 페이지로 변경된다. |

|  |  |
| --- | --- |
| **기능** | 페이지 삭제 |
| **설명** | 사용자가 원하는 페이지를 삭제할 수 있다. |
| **입력** | 사용자가 삭제하려는 페이지의 이름을 클릭 |
| **출처** | 사용자가 선택한 페이지 |
| **출력** | DB(어떤걸로 통일하죠?) 에서 가져온 페이지 목록 데이터 |
| **목적지** | 페이지 목록? 메인 페이지? |
| **동작** | DB로부터 선택한 페이지와 관련된 모든 정보를 삭제한다. |
| **필요사항** | 사용자가 삭제 버튼을 클릭한다 |
| **사전조건** | DB에 삭제하려는 페이지의 정보가 존재해야 한다. |
| **사후조건** | DB에 존재하는 페이지 데이터를 제외하고 저장한다. |

|  |  |
| --- | --- |
| **기능** | 페이지 수정 |
| **설명** | 사용자가 원하는 페이지를 수정할 수 있다. |
| **입력** | 사용자가 수정하고자 하는 페이지의 이름을 클릭 |
| **출처** | 사용자가 선택한 페이지 |
| **출력** | 수정하고자 하는 페이지로 이동 |
| **목적지** | 수정할 페이지 화면 |
| **동작** | DB에 저장된 기존의 페이지 내용을 보여주고, 사용자가 수정하고자 하는 부분을 수정한다. |
| **필요사항** | 페이지 수정 버튼을 클릭해야 한다. |
| **사전조건** | DB에 수정할 페이지가 존재해야 한다. |
| **사후조건** | 수정된 페이지의 내용이 출력된다. |

|  |  |
| --- | --- |
| **기능** | 페이지 저장 |
| **설명** | 사용자가 페이지 저장을 클릭할 경우 페이지 내의 기록한 내용들이 DB에 저장되어야 한다. |
| **입력** | 사용자가 페이지를 선택하고 페이지 저장 버튼을 클릭한다. |
| **출처** | 사용자가 클릭한 페이지 저장 버튼 |
| **출력** | DB에 페이지를 저장한다. |
| **목적지** | DB |
| **동작** | 페이지를 선택하고 페이지 저장버튼을 클릭할 경우, DB에 저장하고 결과를 출력한다. |
| **필요사항** | 저장하려는 페이지가 선택되어야 한다. |
| **사전조건** | 사용자가 선택할 페이지가 생성되어 있어야 한다. |
| **사후조건** | 사용자가 저장한 페이지가 정상적으로 DB에 저장되어야 한다. |

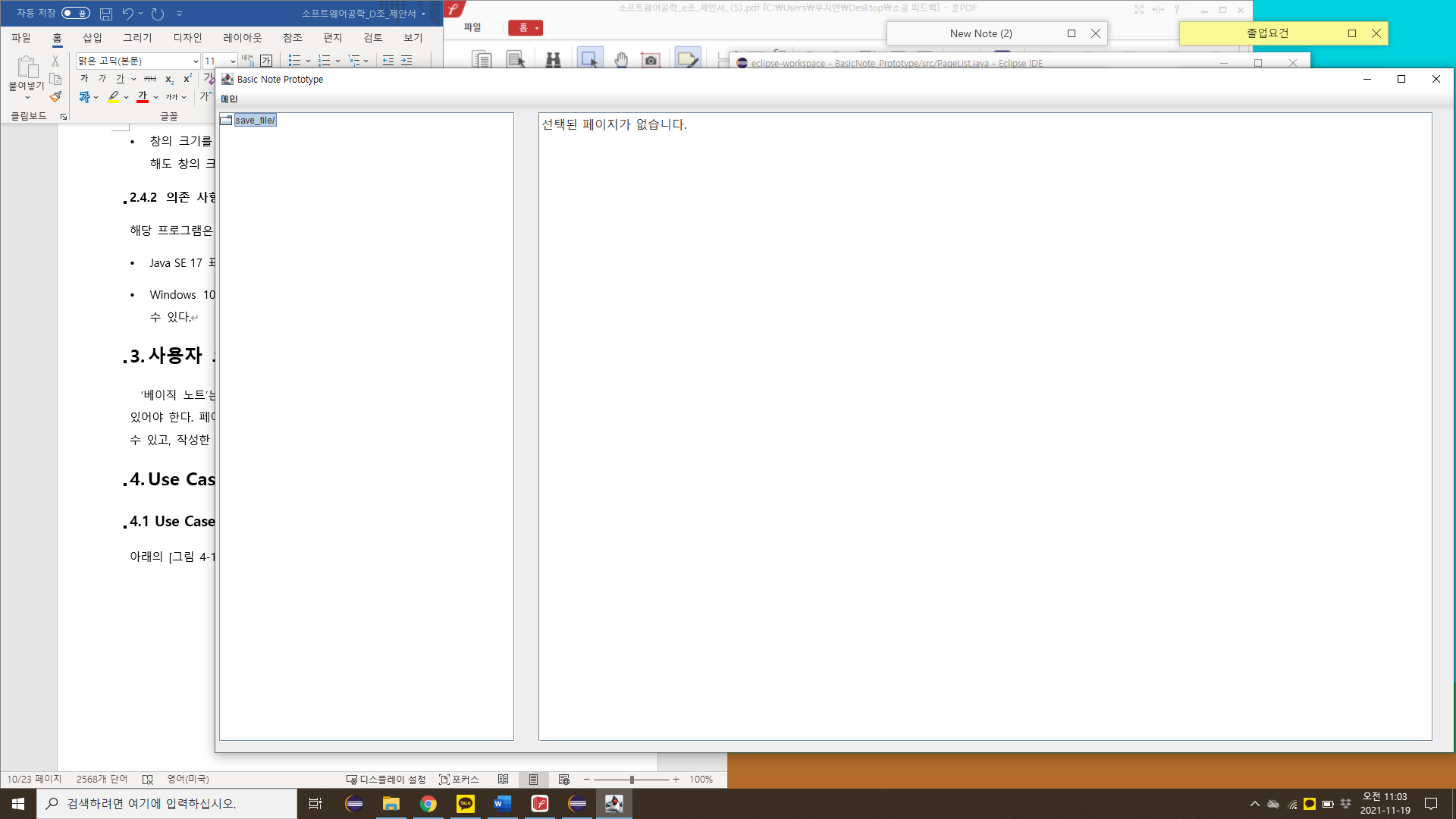
|  |  |
| --- | --- |
| **기능** | 페이지 열기 |
| **설명** | 사용자가 선택한 페이지의 정보를 DB에서 가져와 메인 화면에 출력해준다. |
| **입력** | 사용자가 페이지 목록에서 열고자 하는 페이지를 클릭한다. |
| **출처** | DB |
| **출력** | 사용자가 열고자 하는 페이지 내용 |
| **목적지** | 메인 화면 |
| **동작** | 열고자 하는 페이지를 페이지 목록에서 클릭할 경우, DB에서 저장된 해당 페이지의 내용을 메인 화면에 출력한다. |
| **필요사항** | 사용자가 페이지 목록에서 열고자 하는 페이지를 클릭한다. |
| **사전조건** | 사용자가 선택할 페이지가 DB에 저장되어 있어야 한다. |
| **사후조건** | 사용자가 선택한 페이지가 정상적으로 메인 화면에 출력되어야 한다. |

|  |  |
| --- | --- |
| **기능** | ???????? |
| **설명** |  |
| **입력** |  |
| **출처** |  |
| **출력** |  |
| **목적지** |  |
| **동작** |  |
| **필요사항** |  |
| **사전조건** |  |
| **사후조건** |  |

# **사용자 요구사항 정의**

## **4.1 요구하는 소프트웨어 구상도**

아래 [그림3-1]은 요구하는 소프트웨어의 구상도이다.



[그림 ‑] 요구하는 소프트웨어 구상도

## **4.2 기능적 요구사항**

해당 프로그램의 기능적 요구사항은 6가지로 나눌 수 있다.

* ‘베이직 노트’는 페이지를 생성할 수 있다.
* 생성된 페이지에 대한 수정을 할 수 있다.
* 작성한 페이지를 저장할 수 있다.
* 상단 좌측의 버튼 중 페이지 삭제 버튼을 클릭하면 생성된 페이지에 대한 삭제가 가능하다.
* 생성된 페이지에 대한 글자크기 수정을 할 수 있다.
* 화면 좌측에 저장된 페이지의 목록을 볼 수 있다.

(‘베이직 노트’는 페이지를 생성, 삭제, 수정할 수 있고, 생성된 페이지의 목록을 볼 수 있어야 한다. 페이지는 삭제 시 복구되지 않는다. 페이지 내에서는 글 작성 및 저장을 할 수 있고, 작성한 글의 크기를 변경할 수 있어야 한다.)

## **4.3 비기능적 요구사항**

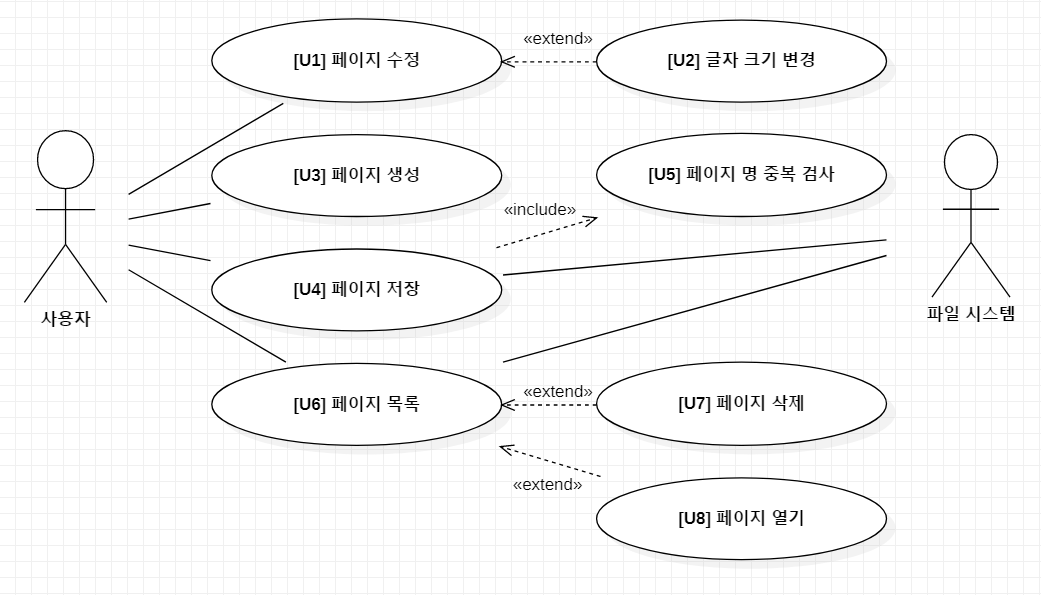
해당 프로그램의 비기능적인 요구사항은 다음과 같다.

* 모든 동작은 1초내에 수행되어야 한다.
* 서버를 사용하지 않으므로 한 번에 하나의 페이지만 수정가능하다.
* 저장된 모든 페이지는 페이지 목록에 존재해야 한다.

# **Use Case Diagram**

## **5.1 Use Case Diagram**

아래의 [그림 5-1]은 ‘베이직 노트’의 Use Case Diagram이다.



[그림 ‑] ‘베이직 노트’의 Use Case Diagram

## **5.2 Use Case Description**

아래의 표들은 [그림 4-1] Use Case Diagram에 대한 명세이다.

[표 ‑] 페이지 수정

|  |  |
| --- | --- |
| **System** | 베이직 노트 |
| **Use Case** | [U1] 페이지 수정 |
| **Actor** | 사용자 |
| **Description** | 사용자가 생성된 페이지에 글을 추가, 삭제하여 데이터를 작성하고, 사용자가 작성한 데이터를 ‘베이직 노트’가 임시로 저장한다. |
| **Stimulus** | 사용자가 ‘베이직 노트’를 통해 페이지 수정을 요청한다. |
| **Response** | 사용자가 작성한 데이터를 임시 메모리에 저장한다. |
| **Comments** | 사용자는 한 번에 하나의 페이지만 수정할 수 있다. |

[표 ‑] 글자 크기 변경

|  |  |
| --- | --- |
| **System** | 베이직 노트 |
| **Use Case** | [U2] 글자 크기 변경 |
| **Actor** | 사용자 |
| **Description** | 사용자가 글 수정 시 원하는 글자 크기를 지정한다. |
| **Stimulus** | 사용자가 페이지 수정 화면에서 글자 크기 변경을 요청한다. |
| **Response** | 글자 크기 메타 데이터를 수정하고 변경된 글자 크기를 시각화한다. |
| **Comments** | 글자 크기의 최소 크기는 1pt이며, 최대 크기는 50pt로 제한한다. |

[표 ‑] 페이지 생성

|  |  |
| --- | --- |
| **System** | 베이직 노트 |
| **Use Case** | [U3] 페이지 생성 |
| **Actor** | 사용자 |
| **Description** | 사용자가 빈 페이지를 생성한다. |
| **Stimulus** | 사용자가 ‘베이직 노트’를 통해 페이지 생성을 요청한다. |
| **Response** | 빈 페이지가 임시로 생성된다. |
| **Comments** |  |

[표 ‑] 페이지 저장

|  |  |
| --- | --- |
| **System** | 베이직 노트 |
| **Use Case** | [U4] 페이지 저장 |
| **Actor** | 사용자, 파일 시스템 |
| **Description** | 사용자가 수정한 데이터를 지정한 이름으로 파일 시스템에 저장한다. |
| **Stimulus** | 사용자가 페이지 수정 화면에서 페이지 저장을 요청한다. |
| **Response** | 사용자가 작성한 문자열 데이터와 메타 데이터를 파일 시스템에 저장한다. |
| **Comments** | save\_file 폴더에 “memo\_”라는 이름과 사용자가 정한 페이지 이름을 더하여 txt 파일로 저장한다. |

[표 ‑] 페이지 명 중복 검사

|  |  |
| --- | --- |
| **System** | 베이직 노트 |
| **Use Case** | [U5] 페이지 명 중복 검사 |
| **Actor** | 사용자, 파일 시스템 |
| **Description** | 사용자가 저장하려는 페이지 명이 존재하는지 검사한다. |
| **Stimulus** | 페이지 저장을 요청하고 새로운 페이지 명을 입력한다. |
| **Response** | 파일 시스템에서 동일한 페이지 명이 존재하는지 검사하여 사용자에게 전달한다. |
| **Comments** | 페이지 명이 중복될 경우 사용자에게 다시 입력하도록 요청한다. |

[표 ‑] 페이지 목록

|  |  |
| --- | --- |
| **System** | 베이직 노트 |
| **Use Case** | [U6] 페이지 목록 |
| **Actor** | 사용자, 파일 시스템 |
| **Description** | 사용자가 저장한 페이지를 확인할 수 있다. |
| **Stimulus** | 사용자가 ‘베이직 노트’를 실행 시 페이지 목록을 확인 가능하고, 새 페이지를 저장하거나 삭제할 때 업데이트 된다. |
| **Response** | 파일 시스템에서 저장되어 있는 페이지 명을 화면에 출력한다. |
| **Comments** | save\_file 폴더에 저장되어 있는 페이지 명을 생성 일자 기준으로 화면에 출력한다. |

[표 ‑] 페이지 삭제

|  |  |
| --- | --- |
| **System** | 베이직 노트 |
| **Use Case** | [U7] 페이지 삭제 |
| **Actor** | 사용자, 파일 시스템 |
| **Description** | 파일 시스템에 저장된 페이지를 삭제한다. |
| **Stimulus** | 사용자가 ‘베이직 노트’를 통하여 페이지 삭제를 요청한다. |
| **Response** | 선택된 페이지가 삭제된다. |
| **Comments** | 삭제된 페이지는 복원할 수 없다. |

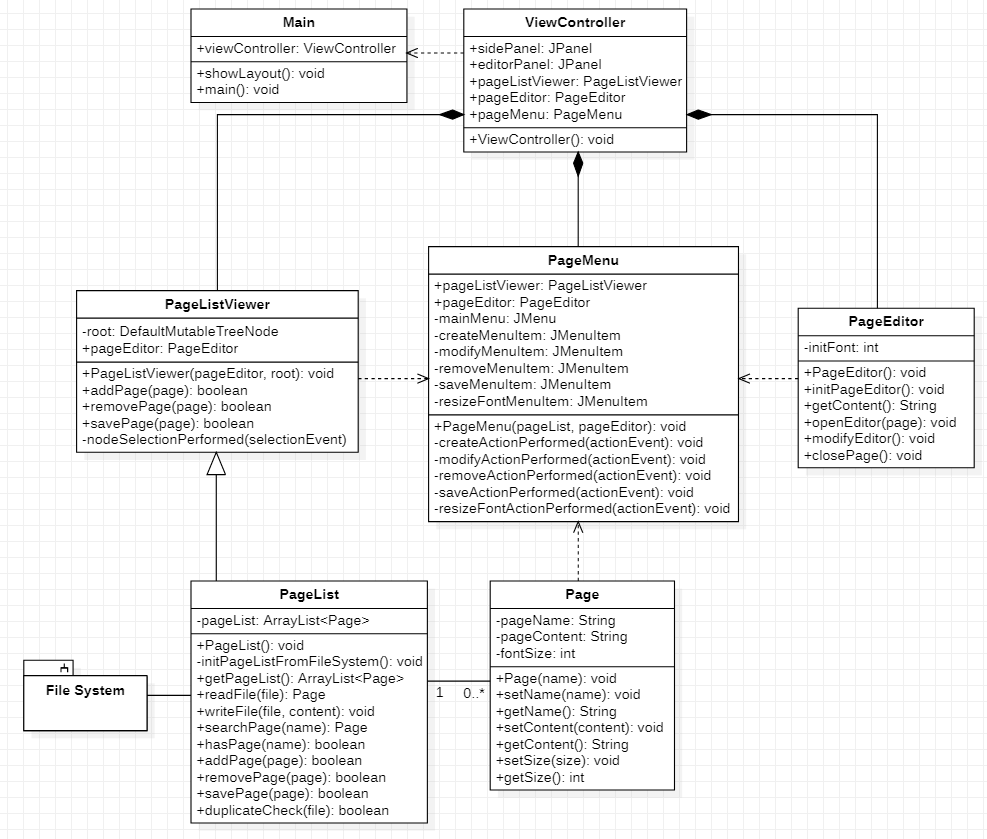
[표 ‑] 페이지 열기

|  |  |
| --- | --- |
| **System** | 베이직 노트 |
| **Use Case** | [U8] 페이지 열기 |
| **Actor** | 사용자, 파일 시스템 |
| **Description** | 사용자가 페이지 목록에서 선택한 페이지의 내용을 확인할 수 있다. |
| **Stimulus** | 사용자가 페이지 목록에서 확인하기 원하는 페이지를 선택한다. |
| **Response** | 파일 시스템에서 사용자가 요청한 페이지의 내용을 화면에 출력한다. |
| **Comments** | save\_file 폴더에 저장되어 있는 파일을 가져온다. |

# **Class Diagram**

## **6.1 Class Diagram**

아래의 [그림 5-1]은 ‘베이직 노트’의 Class Diagram이다.



[그림 ‑] ‘베이직 노트’의 Class Diagram

## **6.2 Class Diagram Description**

아래의 표들은 [그림 5-1] Class Diagram에 대한 명세이다.

1. **Main**

* 프로그램의 진입점으로, 메인 프레임을 구성하는 ViewController 객체를 생성한다.
* ViewController와 의존관계이다.

[표 ‑] Main 클래스의 변수

|  |  |  |  |
| --- | --- | --- | --- |
| **변수 명** | **권한** | **자료형** | **설명** |
| viewController | public | ViewController | ViewController 객체를 참조하는 변수이다. |

[표 ‑] Main 클래스의 메서드

|  |  |  |  |
| --- | --- | --- | --- |
| **메서드 명** | **권한** | **파라미터** | **설명** |
| main | public | args | 프로그램의 진입점으로 viewController 객체를 생성한다. |

1. **ViewController**

* JFrame을 상속받는 UI 클래스로 트리뷰를 포함하는 좌측 패널, 텍스트 에디터를 포함하는 에디터 패널, 상단의 메뉴바로 구성된다.
* 메인 프레임을 담당하며, Main 클래스에서 ViewController 클래스의 객체를 생성할 때 동작된다.

[표 ‑] ViewController 클래스의 변수

|  |  |  |  |
| --- | --- | --- | --- |
| **변수 명** | **권한** | **자료형** | **설명** |
| sidePanel | public | JPanel | 트리뷰를 포함하는 패널로 메인 프레임의 좌측에 존재한다. |
| editorPanel | public | JPanel | 텍스트 에디터를 포함하는 패널로 메인 프레임의 중앙에 존재한다. |
| pageListViewer | public | PageListViewer | 사이드 패널을 구성하는 트리뷰 컴포넌트로서 페이지 목록을 보여준다. |
| pageEditor | public | PageEditor | 에디터 패널을 구성하는 텍스트 에디터 컴포넌트로서 텍스트 작성을 위해 사용된다. |
| pageMenu | public | PageMenu | 메인 프레임 상단에 존재하여, 새 페이지 생성 및 페이지 수정 기능 등을 호출하기 위해 사용된다. |

[표 ‑] ViewController 클래스의 메서드

|  |  |  |  |
| --- | --- | --- | --- |
| **메서드 명** | **권한** | **파라미터** | **설명** |
| ViewController | public | void | ViewController의 생성자로, 모든 변수에 해당하는 객체를 생성하고, 컴포넌트를 올바른 패널에 부착한다. |

1. **PageMenu**

* 메인 프레임 상단 메뉴바를 담당하며 새 페이지 생성, 페이지 수정, 페이지 삭제, 페이지 저장, 글자 크기 변경 기능을 제공한다.

[표 ‑] PageMenu 클래스의 변수

|  |  |  |  |
| --- | --- | --- | --- |
| **변수 명** | **권한** | **자료형** | **설명** |
| pageListViewer | public | PageListViewer | ViewController 클래스에서 생성된 PageListViewer객체이다. ageListViewer 클래스의 메서드를 호출하기 위한 레퍼런스 변수로서 존재한다. |
| pageEditor | public | PageEditor | ViewController 클래스에서 생성된 PageEditor 객체이다. PageEditor 클래스의 메서드를 호출하기 위한 레퍼런스 변수로서 존재한다. |
| mainMenu | private | JMenu | 메뉴바의 메인 메뉴이다. 이 메뉴에 다섯 가지 기능의 메뉴 아이템이 포함된다. |
| createMenuItem | private | JMenuItem | 새 페이지 생성 기능을 호출하는 메뉴 아이템이다. |
| modifyMenuItem | private | JMenuItem | 페이지 수정 기능을 호출하는 메뉴 아이템이다. |
| removeMenuItem | private | JMenuItem | 페이지 삭제 기능을 호출하는 메뉴 아이템이다. |
| saveMenuItem | private | JMenuItem | 페이지 저장 기능을 호출하는 메뉴 아이템이다. |
| resizeFontMenuItem | private | JMenuItem | 폰트 크기 변경 기능을 호출하는 메뉴 아이템이다. |

[표 ‑] PageMenu 클래스의 메서드

|  |  |  |  |
| --- | --- | --- | --- |
| **메서드 명** | **권한** | **파라미터** | **설명** |
| PageMenu | public | pageList, pageEditor | PageMenu 클래스의 생성자로 메인 프레임 상단바에 메인 메뉴를 부착하고, 메인 메뉴에 다섯 개의 메뉴 아이템을 삽입한다. 또한, 각 메뉴 아이템에 ActionListener를 삽입하고 메뉴 아이템에 해당하는 메서드를 연결한다. |
| createActionPerformed | private | actionEvent | 새 페이지 생성 기능에 대한 메서드로createMeuItem이 호출되었을 때 동작한다. 새로운 페이지 객체를 생성하고 pageListViewer의 addPage를 호출한다. |
| modifyActionPerformed | private | actionEvent | 페이지 수정 기능에 대한 메서드로 modifyMenuItem이 호출되었을 때 동작한다. pageEditor에서 openEditor 메서드를 호출한다. |
| removeActionPerformed | private | actionEvent | 페이지 삭제 기능에 대한 메서드로 removeMenuItem이 호출되었을 때 동작한다. PageListViewer의 removePage 메서드와 pageEditor의 closePage 메서드를 호출한다. |
| saveActionPerformed | private | actionEvent | 페이지 저장 기능에 대한 메서드로 saveMenuItem이 호출되었을 때 동작한다. pageListViewer의 savePage 메서드를 호출한다. |
| resizeFontActionPerformed | private | actionEvent | 폰트 크기 변경 기능에 대한 메서드로 pageListViewer를 통해 선택된 페이지를 검색하고, 해당 페이지의 폰트 크기를 변경한다. 이는 UI에 바로 적용된다. |

1. **PageListViewer**

* 페이지 리스트를 저장하는 클래스인 PageList와 트리뷰를 구성하는 자바 클래스인 JTree를 상속받는 클래스이다.

[표 ‑] PageListViewer 클래스의 변수

|  |  |  |  |
| --- | --- | --- | --- |
| **변수 명** | **권한** | **자료형** | **설명** |
| root | private | DefaultMutableTreeNode | 트리뷰를 구성하는 최상단 노드이다. |
| pageEditor | public | PageEditor | PageEditor 메서드를 호출하기 위한 레퍼런스 변수이다. |

[표 ‑] PageListViewer 클래스의 메서드

|  |  |  |  |
| --- | --- | --- | --- |
| **메서드 명** | **권한** | **파라미터** | **설명** |
| PageListViewer | public | pageEditor,root | PageListViewer 클래스의 생성자로 트리뷰를 구성한다. |
| addPage | public | page | root 하단에 새 페이지 노드를 추가하고, 부모 클래스의 addPage 메서드를 호출한다. 실패할 경우 false를 반환한다. |
| removePage | public | page | 선택된 페이지 노드를 삭제하고, 부모 클래스의 removePage 메서드를 호출한다. 선택된 노드가 없거나 root 노드가 선택된 경우, 부모 클래스의 removePage 메서드가 실패한 경우 false를 반환한다. |
| savePage | public | page | 새로 저장할 페이지 이름을 입력받고, 이를 page에 적용한다. 이후, 부모 클래스의 savePage 메서드를 호출한다. |
| nodeSelectionPerformed | private | selectionEvent | 트리뷰를 통해 페이지가 선택되었을 때 해당 페이지의 내용이 pageEditor에 출력된다. |

1. **PageList**

* PageListViewer의 백그라운드 동작을 담당하며 List 변수인 pageList를 관리한다.

[표 ‑] PageList 클래스의 변수

|  |  |  |  |
| --- | --- | --- | --- |
| **변수 명** | **권한** | **자료형** | **설명** |
| pageList | private | ArrayList<Page> | Page 객체를 리스트로 저장하는 변수이다. |

[표 ‑] PageList 클래스의 메서드

|  |  |  |  |
| --- | --- | --- | --- |
| **메서드 명** | **권한** | **파라미터** | **설명** |
| PageList | public | void | PageList 클래스의 생성자로 pageList에 해당하는 ArrayList 객체를 생성한다.  그리고 initPageListFromFileSystem 메서드를 호출한다. |
| initPageListFromFileSystem | private | void | 파일 시스템으로부터 파일을 읽어온 후 readFile 메서드를 통해 Page를 생성하고 pageList에 삽입한다. |
| getPageList | public | void | pageList를 반환하는 get 메서드이다. pageList의 직접 참조를 막기 위해 존재한다. |
| readFile | public | file | 텍스트 파일을 읽어 내용과 폰트 크기 메타 데이터를 분리한 뒤, 새 Page 객체를 생성한다. |
| writeFile | public | file, content | 내용에 메타 데이터가 부착되었음을 가정하고 새 파일로 저장한다. 만약 duplicateCheck에서 true를 반환하거나 저장에 실패한 경우 IOException을 throw한다. |
| searchPage | public | name | pageList를 순차적으로 탐색하여 이름에 맞는 Page 객체를 리턴한다. |
| hasPage | public | name | pageList를 순차적으로 탐색하여 이름에 맞는 Page의 존재 여부를 리턴한다. |
| addPage | public | name | 새 Page 객체를 pageList에 삽입한다. 삽입에 성공한 경우 true를 리턴한다. |
| removePage | public | page | 입력받은 Page 객체를 pageList에서 삭제한다. 삭제에 성공한 경우 true를 리턴한다. |
| savePage | public | page | 입력받은 Page 객체를 파일 시스템에 저장한다. 내용에 폰트 크기 메타데이터를 삽입하고 새 File 객체를 생성하여 writeFile 메서드를 호출한다. 만약 writeFile 메서드를 호출하는 동안 에러가 검출되면 false를 리턴한다. |
| duplicateCheck | public | file | 중복 검사 메서드이다. 새 파일을 저장할 때 기존 파일 중 해당 파일이 있는지 검색한다. 존재한다면 true를 반환한다. |

1. **Page**

* 페이지에 해당하는 클래스이다. 세 속성 페이지 이름, 내용, 폰트 크기로 구성된다.

[표 ‑] Page 클래스의 변수

|  |  |  |  |
| --- | --- | --- | --- |
| **변수 명** | **권한** | **자료형** | **설명** |
| pageName | private | String | 페이지 이름을 의미한다. |
| pageContent | private | String | 페이지 내용을 의미한다. |
| fontSize | private | int | 폰트 크기를 의미한다. |

[표 ‑] Page 클래스의 메서드

|  |  |  |  |
| --- | --- | --- | --- |
| **메서드 명** | **권한** | **파라미터** | **설명** |
| Page | public | name | Page 클래스의 생성자로 페이지 이름을 입력받는다. |
| setName | public | name | 페이지 이름을 설정하는 메서드이다. pageName의 직접 참조를 막기 위해 존재한다. |
| getName | public | void | 페이지 이름을 반환하는 메서드이다. |
| setContent | public | content | 페이지 내용을 설정하는 메서드이다. pageContent의 직접 참조를 막기 위해 존재한다. |
| getContent | public | void | 페이지 내용을 반환하는 메서드이다. |
| setSize | public | size | 페이지의 폰트 크기를 설정하는 메서드이다. fontSize의 직접 참조를 막기 위해 존재한다. |
| getSize | public | void | 페이지의 폰트 크기를 반환하는 메서드이다. |

1. **PageEditor**

* 페이지를 작성하기 위한 클래스이며 텍스트 에디터 컴포넌트를 상속받는다.

[표 ‑] PageEditor 클래스의 변수

|  |  |  |  |
| --- | --- | --- | --- |
| **변수 명** | **권한** | **자료형** | **설명** |
| initFont | private | int | 기본 폰트 크기를 설정하는 변수이다. |

[표 ‑] PageEditor 클래스의 메서드

|  |  |  |  |
| --- | --- | --- | --- |
| **메서드 명** | **권한** | **파라미터** | **설명** |
| PageEditor | pbulic | void | PageEditor 클래스의 생성자로 initFont 변수를 설정하고, initPageEditor 메서드를 실행한다. |
| initPageEditor | public | void | 페이지가 선택되지 않았다는 메시지를 텍스트 에디터 컴포넌트에 표시한다. 그리고 텍스트 에디터 컴포넌트의 입력 가능 여부를 거부한다. |
| getContent | public | void | 텍스트 에디터 컴포넌트에 작성된 내용을 반환한다. |
| openEditor | public | page | 입력받은 페이지의 내용을 텍스트 에디터 컴포넌트에 입력하고, 페이지의 폰트 크기를 텍스트 에디터 컴포넌트에 설정한다. |
| modifyEditor | public | void | 텍스트 에디터 컴포넌트의 입력 가능 여부를 허용한다. |
| closePage | public | void | initPageEditor 메서드를 수행한다. |

# **참조**

* IEEE 소프트웨어 요구사항 명세서 표준 템플릿:   
  <https://goo.gl/nsUFwy>
* Software Engineering, Tenth Edition 제시 양식:  
  <https://iansommerville.com/software-engineering-book/static/web/requirements-standard/>