

|  |
| --- |
| 소프트웨어 설계서 |
| 포트폴리오 관리 소프트웨어 |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **제출일** | 2018. 11. 02. |  | **전공** | 컴퓨터공학과 |
| **과목** | 소프트웨어공학 |  | **조장** | 김진우 |
| **담당교수** | 안동혁 |  | **조원** | 김태환, 문형진, 임창우, 강동현 |

CONTENTS

[1. 개요 4](#_Toc528939086)

[1.1. 시스템 개요 4](#_Toc528939087)

[1.2. 범위 4](#_Toc528939088)

[1.3. 목적 4](#_Toc528939089)

[2. 소프트웨어 모델 5](#_Toc528939090)

[2.1. 상호작용 모델 5](#_Toc528939091)

[2.2. 구조 모델 9](#_Toc528939094)

[2.3. 동작 모델 11](#_Toc528939097)

[3. 소프트웨어 아키텍처 15](#_Toc528939100)

[3.1. MVC(Model-View-Controller) 패턴 15](#_Toc528939101)

[4. 참조(Reference) 16](#_Toc528939105)

[4.1. 서적 16](#_Toc528939106)

[4.2. 웹사이트 및 PDF 16](#_Toc528939108)

1. 개요
   1. 시스템 개요

Portfolio Manager는 사용자가 편리하게 포트폴리오를 구성할 수 있도록 하기 위해 제작할 포트폴리오 관리 시스템이다. 사용자가 원하는 카드를 만들고 관리하며 카드에 각종 정보를 추가 가능하다.

* 1. 범위

본 문서는 Portfolio Manager의 구현을 위한 설계 사양을 기술한다.

* 1. 목적

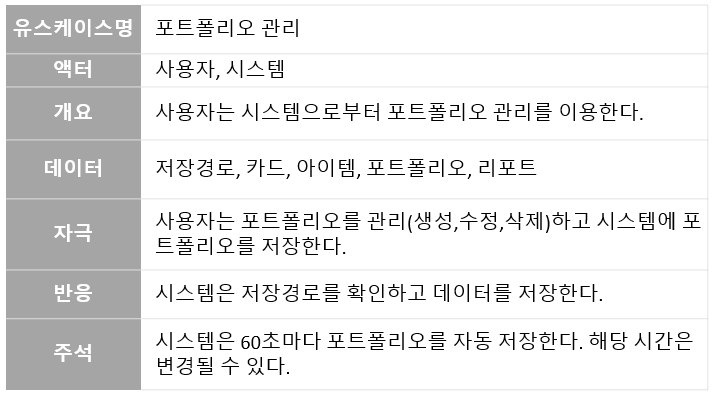
본 문서의 목적은 소프트웨어 요구사항 명세서에 따라 Portfolio Manager 시스템을 개발하기 위함에 있다. 소프트웨어 아키텍처, 모델에 따른 구조도 및 UML 등으로 소프트웨어를 설계한다. 또한 소프트웨어 구현 도중 발생 가능한 변경 사항들에 대해서 구조 수정을 용이하기 하기 위함에 있다.

1. 소프트웨어 모델
   1. 상호작용 모델

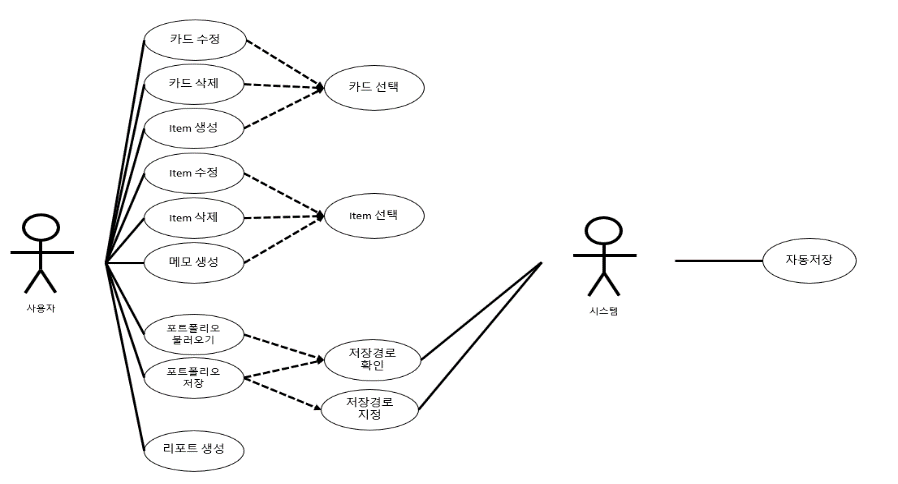
상호작용 모델은 사용자 입출력과 관련된 사용자 상호작용 또는 개발되는 소프트웨어와 그 환경의 다른 시스템과의 상호작용, 소프트웨어 시스템의 컴포넌트들 간의 상호작용 등 여러 가지로 표현된다. 유스케이스 모델링과 시퀀스 다이어그램으로 Portfolio Manager 시스템에 대한 상호 작용 모델을 표현한다.

* + 1. 유스케이스 모델링(Usecase Modeling)

유스케이스 모델링을 통해 Portfolio Manger 시스템과 외부 에이전트(사용자)와의 상호작용을 단순하게 표현한다. 표 형식을 통해 유스케이스 다이어그램을 상세하게 표현한다.



<그림 1> 포트폴리오 관리 유스케이스[표 형식]



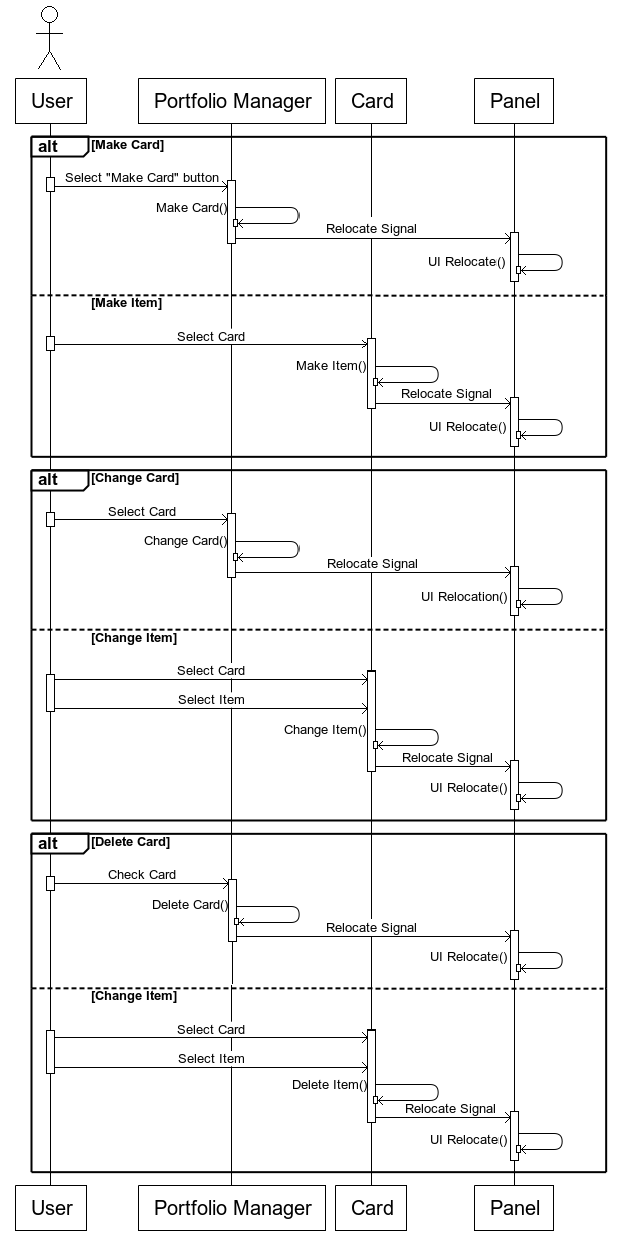
<그림 2> 유스케이스 다이어그램(Diagram)

* + 1. 시퀀스 다이어그램(Sequence Diagram)

시퀀스 다이어그램을 통해 사용자와 시스템의 객체들 간의 상호 작용과 객체들 간의 상호작용을 모델링한다.

* **1. Portfolio 관리**

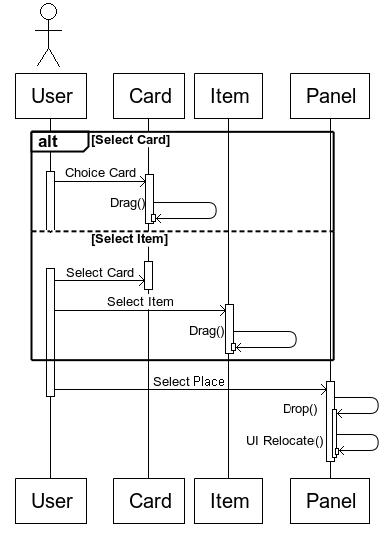
사용자는 포트폴리오 관리를 위해 카드를 수정, 삭제, 생성한다. 또한 카드 하위 항목인 아이템(Item)을 수정, 상제, 생성한다. 해당 객체들을 수정, 삭제하기 위해서는 카드나 아이템을 선택(Select)한다.



<그림 3> 포트폴리오 관리 시퀀스 다이어그램

* **2. UI Drag & Drop**

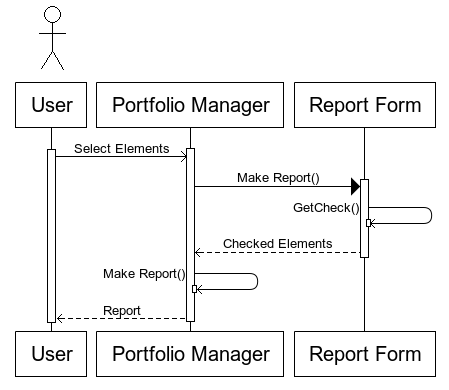
사용자는 카드와 아이템의 위치를 자신이 원하는 위치로 변경한다. 드래그를 하고 패널(Panel) 위 원하는 위치로 놓으면 패널은 전체 UI를 재배치(UI Relocation)한다.



<그림 4> UI 재배치 시퀀스 다이어그램

* **3. Report 관리**

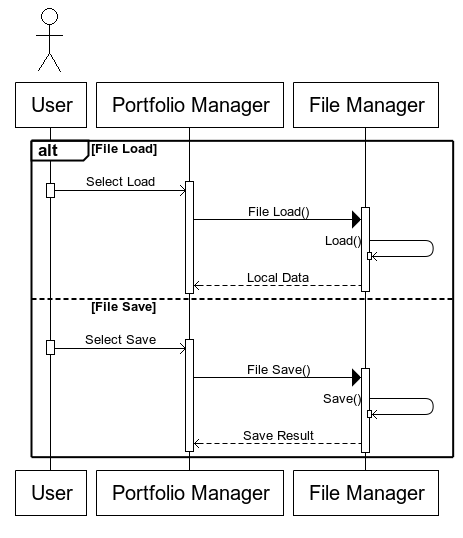
사용자는 Report에 나타날 카드들을 선택하여 Portfolio Manager로 객체들을 전달한다. Portfolio Manger는 객체를 확인한 후 새로운 폼(Form)에 전달하여 리포트를 생성한다.



<그림 5> 리포트 관리 시퀀스 다이어그램

* **4. 파일 관리**

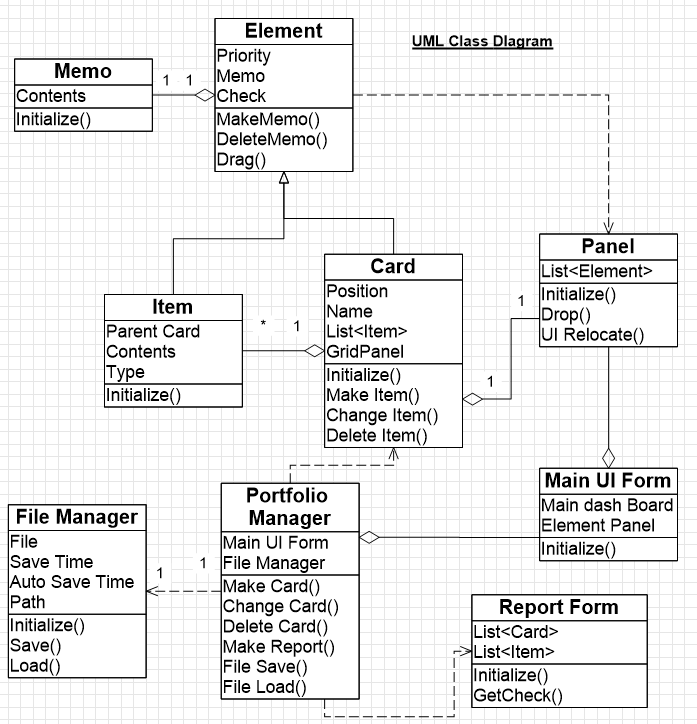
사용자의 저장, 불러오기를 통해 파일 경로를 전달한다. File Manager는 파일 저장, 불러오기를 직접 수행하는 객체이다.



<그림 6> 파일 관리 시퀀스 다이어그램

* 1. 구조 모델

구조 모델은 시스템을 구성하는 컴포넌트들을 정의하고 컴포넌트들의 관계로 시스템 구성을 표현한다. 시스템이 실행될 때의 동적 모델을 클래스 다이어그램으로 표현한다.

* + 1. 클래스 다이어그램(Class Diagram)

<그림 7> 클래스 다이어그램

* + 1. 컴포넌트와 관계 설명

Ⅰ. Element, Item, Card, Memo

Ⅰ-1 Element

* Card와 Item의 부모 클래스
* Memo Class를 가지고 있으며 생성하거나 제거를 할 수 있다.
* Card와 Item의 공통적인 기능을 가지고 있다.

Ⅰ-2 Card

* 포트폴리오의 최상위 요소(자기소개서, 활동 내역 등등)
* 각각의 Card마다 세부사항으로 Item의 리스트를 가진다.
* 생성된 Card를 수정, 삭제 가능하다.

Ⅰ-3 Item

* 사용자가 실질적으로 내용을 입력하는 UI
* 자신이 속한 부모 Card를 알고 있다.

Ⅰ-4 Memo

* Card와 Item에 입력 가능한 메모
* 리포트 생성시에 나타나지 않는다.

Ⅱ. File Manager

* 파일의 입출력과 자동 저장 기능을 수행하는 클래스

Ⅲ. Portfolio Manager

* 사용자와 각각의 Manager와 연결시키는 역할을 한다.
* Main UI Form을 가지고 있으며 각각의 Manager를 참조한다.

Ⅳ. Main UI Form, Panel

Ⅳ-1 Main UI Form

* Panel을 가진다.
* 프로그램의 메인 폼이며 각 메뉴 버튼들이 포함된다.

Ⅳ-2 Panel

* Card와 Item이 위치하는 폼이다.
* Drag되어진 Card나 Item을 Drop하는 장소로, Drop된 위치와 유사한 레이아웃의 위치에 자리 잡고 다른 Element들은 재배치된다.
  1. 동작 모델

동작 모델은 시스템이 실행될 때의 직접적인 동작에 대한 모델이다. 시스템의 사용자의 자극에 의해 어떤 반응을 일으키는지 직접적으로 표현한다.

* + 1. 이벤트 주도 모델링
* **시스템 상태 및 자극**

시스템의 상태와 자극을 표 형식으로 표현한다.

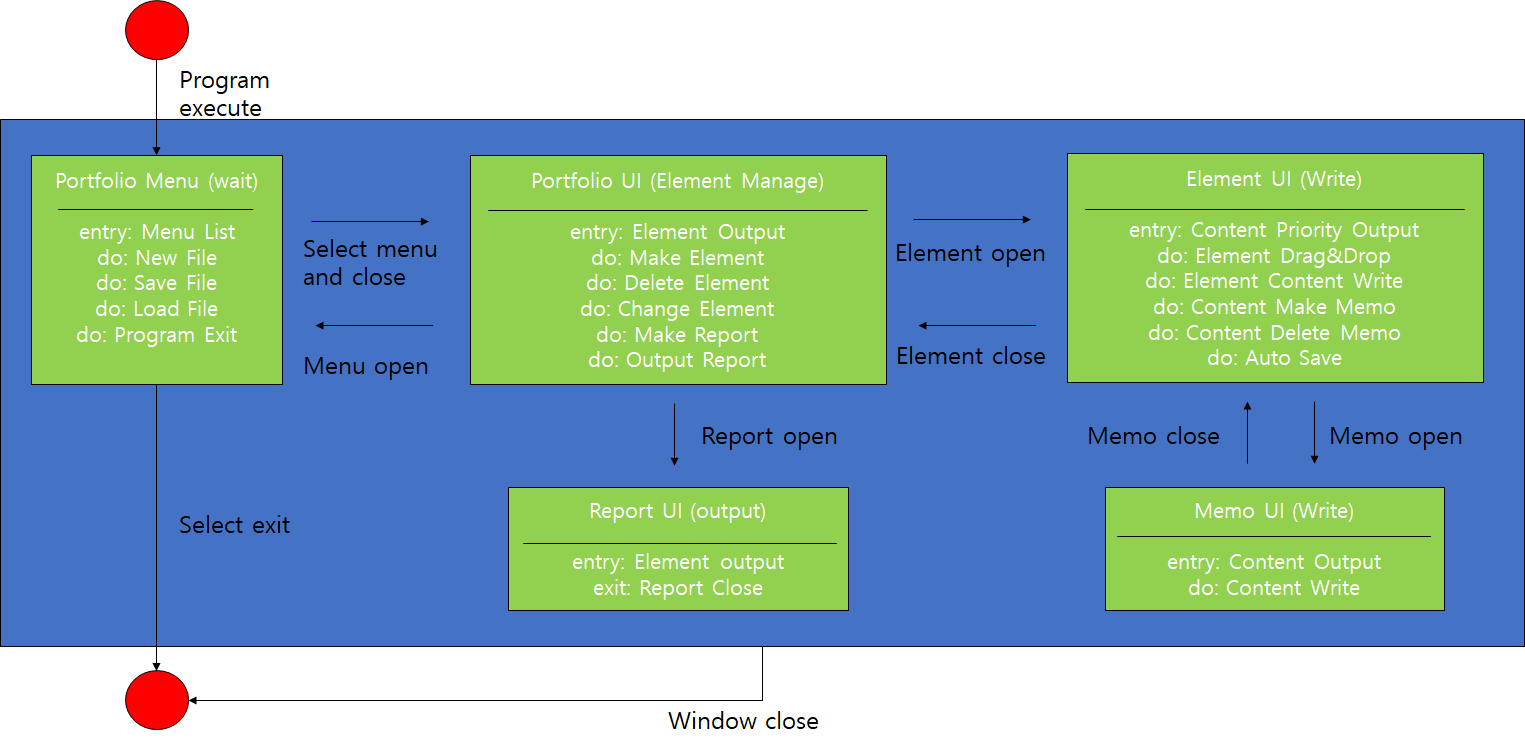
|  |  |
| --- | --- |
| **상태** | **설명** |
| **대기** | **메뉴 선택을 기다리고 메뉴 리스트를 보여줌** |
| **Element 관리** | **Element들을 보여주며 관리 기능을 보여줌** |
| **출력** | **선택된 Element의 내용을 보여줌** |
| **작성** | **Element의 내용 또는 Memo를 작성** |

<그림 8> 상태 표

|  |  |
| --- | --- |
| **자극** | **설명** |
| **프로그램 실행** | **사용자가 프로그램을 실행한다.** |
| **새로운 파일** | **사용자가 메뉴에서 New File 선택** |
| **파일 저장** | **사용자가 메뉴에서 Save File 선택** |
| **파일 불러오기** | **사용자가 메뉴에서 Load File 선택** |
| **프로그램 종료** | **사용자가 종료 버튼을 선택** |
| **Element 생성** | **사용자가 메뉴에서 Make Card 또는 Card에서 Make Item 선택** |
| **Element 삭제** | **사용자가 메뉴에서 Delete Card 또는 Card에서 Delete Item 선택** |
| **Element 수정** | **사용자가 메뉴에서 Change Card 또는 Card에서 Change Item 선택** |
| **Report 생성** | **사용자가 메뉴에서 Make Report 선택** |
| **Drag & Drop** | **사용자가 Element를 Drag & Drop** |
| **Memo 생성** | **사용자가 Element에 Memo 생성** |
| **Memo 삭제** | **사용자가 Element의 Memo 삭제** |
| **자동 저장** | **일정 시간마다 시스템이 파일 저장** |

**<그림 9> 상태 표**

* **상태 다이어그램(State Diagram)**

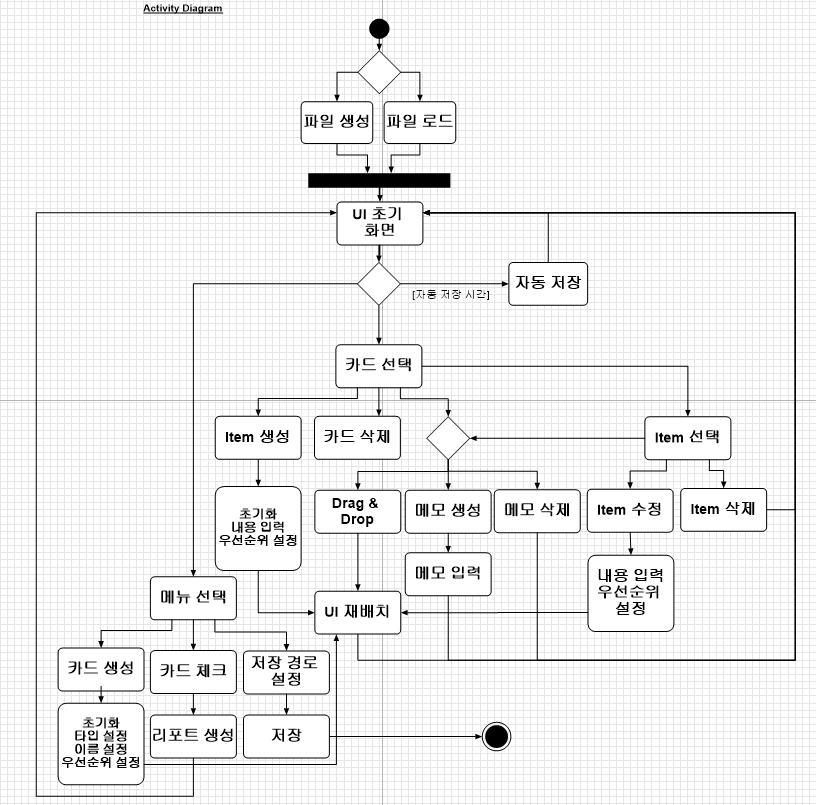


**<그림 10> 상태 다이어그램**

* + 1. 데이터 주도 모델링

입력데이터의 처리과정을 단계적으로 표현한다.

* **액티비티 다이어그램(Activity Diagram)**



**<그림 11> 액티비티 다이어그램**

* **상세 활동 및 순서**
  + 활동 순서

1. 사용자는 기존 파일을 사용할지 새로운 파일을 생성할 지 선택한다
2. 선택이 완료되면 UI 초기 화면이 나타난다
3. 사용자는 자신의 활동을 선택하여 수행한다
   1. 활동 목록은 다음과 같다
      1. 메뉴
         1. 카드 생성
         2. 카드 체크
         3. 저장
      2. 카드 선택
         1. Item 관리
         2. 카드 삭제
         3. UI 재배치
         4. 메모 관리
   2. 자동 저장은 강제적으로 수행된다.
4. 프로그램의 종료는 저장이후 가능하다.
   * 모든 활동은 사용자의 선택에 기반하여 동작한다.
5. 소프트웨어 아키텍처
   1. MVC(Model-View-Controller) 패턴

본 프로그램은 데이터와 데이터 처리, 사용자에게 보일 부분이 각각 나눠져 동작하게 된다. 계층화 패턴, 클라이언트-서버 패턴, 마스터-슬레이브 패턴 등 여러 소프트웨어 아키텍처 패턴 중 해당 과정은 MVC 패턴에 가장 적합하다.

* + 1. Model

Ⅰ. 사용자가 입력한 데이터들이다.

Ⅱ. 별도 DB 없이 직렬화 파일로 관리된다.

Ⅲ. 저장 작업이 수행되지 않았을 경우 UI각각의 Field 데이터를 사용한다.

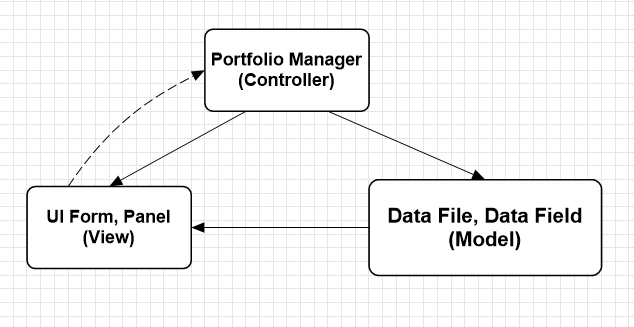
* + 1. View

Ⅰ. 메인 UI Form과 Card와 Item이 나타날 Panel이다.

UI를 사용자가 Drag & Drop으로 위치를 수정할 수 있기에 일시적으로 Controller와 관계를 가지게 된다.

* + 1. Controller

Ⅰ. Portfolio Manager로 전반적인 이벤트에 대한 처리를 담당한다.

Ⅱ. File Manager 또한 Controller에 속한다.

<그림 12> MVC Model 구조도

1. 참조(Reference)
   1. 서적
      1. Sommervile 소프트웨어 공학 10판
   2. 웹사이트 및 PDF
      1. <http://www.sw-eng.kr/common/attachfile/FileDown.do?fileId=00000000000000033373>
      2. <https://www.cisp.or.kr/wp-content/uploads/2016/10/20161020_142553.pdf>
      3. <http://myeonguni.tistory.com/792>
      4. <http://sfeg.tistory.com/339>
      5. <http://www.jidum.com/jidums/view.do?jidumId=982>
      6. <https://msdn.microsoft.com/ko-kr/library/dd409377.aspx?f=255&MSPPError=-2147217396>