소프트웨어 요구사항 명세서

|  |  |
| --- | --- |
| 과 목 | 소프트웨어 공학 |
| 담 당 교 수 | 안동혁 |
| 작 성 자 | 김진우(조장), 김태환, 문형진, 임창우, 강동현 |

**변경 이력**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 버전 | 변경 내역 | 작성 | | 검토 | | 승인 | |
| 일자 | 담당자 | 일자 | 담당자 | 일자 | 담당자 |
| 1.1 | 파일 관리, 항목 관리 분리 | 11/02 | 김진우 |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**목차**

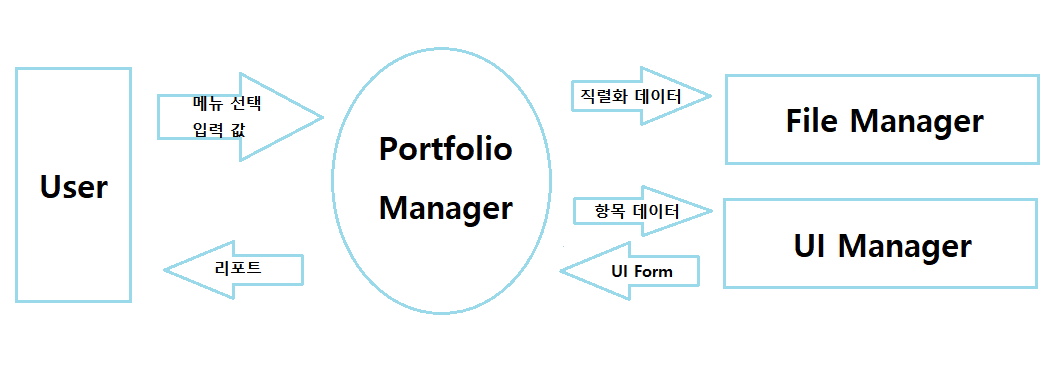
1. 문서 개요
2. 시스템 개요
   1. 시스템 컨텍스트
   2. 입/출력 개요
3. 소프트웨어 기능 요구사항
   1. 기능 요구사항 목록
   2. 기능 흐름
   3. 개념적 데이터 모델
   4. 기능 요구사항 정의
4. 소프트웨어 품질 요구사항
   1. 품질 요구사항 목록
   2. 품질 요구사항 정의
5. 인터페이스 요구사항
6. 제약 사항
7. **문서 개요**

본 문서는 고객 및 이해관계자의 관점으로 기술된 고객 요구사항을 바탕으로 소프트웨어 제품 관점의 요구사항을 구체화한 결과로서 소프트웨어 요구사항을 기술한다.

본 문서의 주요 구성 항목은 다음과 같다.

* 시스템 개요: 시스템의 Context로서 개발 대상 시스템과 외부와의 입/출력을 정의한다.
* 소프트웨어 기능 요구사항: 소프트웨어 기능 요구사항을 정의한다.
* 소프트웨어 품질 요구사항: 소프트웨어 품질 요구사항을 정의한다.
* 인터페이스 요구사항: 시스템과 외부 및 내부 요소 간의 인터페이스에 대한 요구사항을 정의한다.
* 제약 사항: 표준, 법규, 규정 등과 같이 프로젝트 외부의 조직에 의해서 요구되는 제약을 정의한다.

1. **시스템 개요**
   1. 시스템 컨텍스트



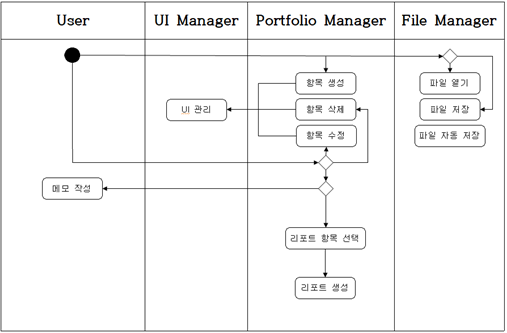
* 1. 입/출력 개요

|  |  |  |
| --- | --- | --- |
| 데이터 명 | 속성 | 설명 |
| 입력 값 | 입력 | User가 Portfolio Manager에 입력하는 데이터 값 |
| 메뉴 선택 | 입력 | User가 Portfolio Manager에서 선택한 메뉴 |
| 리포트 | 출력 | Portfolio Manager에서 User에게 보여주는 양식을 갖춘 데이터 |
| 직렬화 데이터 | 출력 | Portfolio Manager가 File Manger에 전달하는 직렬화 데이터 스트림 |
| 항목 데이터 | 출력 | Portfolio Manager에서 UI Manager로 전달하는 데이터 값 |
| UI Form | 입력 | UI Manager에서 Portfolio Manager로 입력하는 UI Form 양식 |

1. **소프트웨어 기능 요구사항**
   1. 기능 요구사항 목록

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 기능  요구사항 ID | 기능  요구사항 명 | 우선순위 | | |
| 중요도 | 난이도 | 우선  순위 |
| FR-001 | 파일 저장 | 4 | 3 | 5 |
| FR-002 | 파일 열기 | 4 | 3 | 6 |
| FR-003 | 리포트 생성 | 4 | 5 | 7 |
| FR-004 | 항목 생성 | 6 | 5 | 1 |
| FR-005 | 항목 수정 | 5 | 5 | 2 |
| FR-006 | 항목 삭제 | 5 | 5 | 3 |
| FR-007 | UI 관리 | 4 | 5 | 4 |
| FR-008 | 파일 자동저장 | 3 | 2 | 11 |
| FR-009 | 메모 작성 | 2 | 2 | 10 |
| FR-010 | 리포트 항목 선택 | 2 | 2 | 9 |
| FR-011 | 데이터 타입 정의 | 2 | 3 | 8 |

* 1. 기능 흐름



* 1. 개념적 데이터 모델

시스템에서 저장/관리되는 데이터로서 기능 흐름에서 각 기능 간의 공유 데이터를 정의한다.

|  |  |  |
| --- | --- | --- |
| 데이터 명 | 속성 | 설명 |
| 직렬화 데이터 파일 | 파일 경로 | 직렬화 파일의 저장 위치와 이름 |
| Class 정보 | 각 데이터와 UI정보가 담긴 클래스의 정보 |
| UI Form | UI 리스트 | 사용되는 UI의 종류 |
| UI 종류 | 관리하기 쉽도록 연결된 리스트 형식으로 저장 |
| UI 배치 정보 | Window창에서 나타날 위치를 설정 |
| 데이터 | 항목/세부사항 데이터가 저장 |
| 항목 데이터 | 항목 명 | User가 입력한 데이터로 이름을 정의 |
| 중요도 | 항목의 배치 기준 |
| 세부사항 데이터 | 세부사항 내용 | 사용자가 입력/수정 가능한 데이터 |
| 중요도 | 세부사항의 배치 기준 |
| 타입 | 포함하는 데이터의 타입(텍스트, 사진, 숫자 등) |
| 세부사항 검색 | 검색 데이터 | 세부사항 내에서 일치하는 데이터(텍스트, 숫자)를 검색 |

* 1. 기능 요구사항 정의
     1. FR-001: 파일 저장

|  |  |
| --- | --- |
| 개요 | Portfolio Manager에 있는 사용자 입력 데이터를 Local에 저장한다. |
| 선행  조건 | * Portfolio Manager에 사용자 입력 데이터가 존재해야 한다. |
| 후행  조건 | * Portfolio Manager에 Local에 저장된 파일을 불러온다 |
| 기본  시나리오 | 1. User는 메뉴에서 “파일”의 “저장” 버튼을 선택한다. 2. Portfolio Manager는 File Manager에게 파일 저장 대화상자를 요청한다. 3. File Manager는 파일 저장 대화상자를 띄운다. 4. User는 경로를 선택한다. 5. File Manager는 Portfolio Manager가 가진 사용자 입력 데이터를 직렬화 데이터로 변환한다. 6. File Manager는 User가 선택한 경로에 사용자 입력 데이터를 저장한다. |
| 대안  시나리오 | * 저장을 하지 않은 시나리오  1. 알림 창을 띄운다. 2. 저장을 선택 시 저장 시나리오 진행한다. 3. 알림 창을 띄우기 전 수행하려 한 작업을 수행한다. |

* + 1. FR-002: 파일 열기

|  |  |
| --- | --- |
| 개요 | User가 사용하고자 하는 데이터를 Local에서 불러온다. |
| 선행  조건 | * Local에서 불러올 파일은 올바른 형식이어야 한다. |
| 후행  조건 | * Local에 Portfolio Manager의 사용자 입력 데이터 파일이 생성된다. |
| 기본  시나리오 | 1. User는 메뉴에서 “파일”의 “열기” 버튼을 선택한다. 2. Portfolio Manager는 File Manager에게 파일 불러오기 대화상자를 요청한다. 3. File Manager는 파일 불러오기 대화상자를 띄운다. 4. User는 파일을 선택한다. 5. File Manager는 User가 선택한 파일을 읽어온다. |
| 대안  시나리오 | * 손상된 파일  1. File Manager는 User에게 잘못된 형식임을 알린다. 2. 열기 기본시나리오 2번으로 돌아간다. |

* + 1. FR-003: 리포트 생성

|  |  |
| --- | --- |
| 개요 | User는 선택한 항목에 대하여 프로그램 내에서 볼 수 있는 리포트를 생성한다. |
| 선행  조건 | * Portfolio Manager에 사용자 입력 데이터가 존재해야 한다. |
| 후행  조건 | * 새로운 Window 창으로 리포트가 화면에 출력된다. |
| 기본  시나리오 | 1. User는 메인 폼에서 선택 가능한 항목 중 하나를 선택한다. 2. Portfolio Manager는 선택한 항목을 리포트로 생성한다. 3. Portfolio Manager는 새로운 Window한다. 4. Portfolio Manager는 새로 만들어진 Window창에 생성한 리포트를 출력한다. |
| 대안  시나리오 | * 선택 항목 없을 경우 * 기본 값으로 전체 항목을 선택한 것으로 간주한다. |

* + 1. FR-004: 항목 생성

|  |  |
| --- | --- |
| 개요 | Portfolio Manager에 있는 자료를 생성이 가능하다. |
| 선행  조건 | * Portfolio Manager에 사용자 입력 데이터가 존재해야 한다. |
| 후행  조건 | * Portfolio Manager 데이터의 생성이 정상적으로 처리된다. |
| 기본  시나리오 | * 항목 생성  1. User는 메인 폼에서 “항목 생성” 버튼을 선택한다. 2. User는 항목의 이름을 입력한다. 3. User는 “생성” 버튼을 누른다. 4. UI Manager에서 생성된 폼에 User가 입력한 데이터를 넣는다. 5. 메인 폼에 생성된 폼을 추가한다. 6. UI 관리 기능의 UI 재배치 시나리오를 진행한다.  * 세부사항 생성  1. User는 항목 창에서 “세부사항 생성” 버튼을 선택한다. 2. 데이터 타입 정의 시나리오를 진행한다. 3. User는 “생성” 버튼을 누른다. 4. UI관리 기능의 UI 재배치 시나리오를 진행한다. |
| 대안  시나리오 | * 항목 이름 중복 시나리오  1. 항목 생성 2번 시나리오에서 중복된 이름을 가진 항목의 이름을 입력한다. 2. 3번 시나리오 이후 항목 이름 중복 알림 창을 띄운다. 3. User는 항목 생성 2번 시나리오로 돌아간다. |

* + 1. FR-005: 항목 수정

|  |  |
| --- | --- |
| 개요 | Portfolio Manager에 있는 자료를 수정 가능하다. |
| 선행  조건 | * Portfolio Manager에 사용자 입력 데이터가 존재해야 한다. |
| 후행  조건 | * Portfolio Manager 데이터의 수정이 정상적으로 처리된다. |
| 기본  시나리오 | * 항목 수정  1. User는 메인 폼에서 수정할 항목을 선택한다. 2. User는 항목 내의 내용을 수정한다. 3. User는 항목 창에서 “항목 수정” 버튼을 누른다. 4. 재확인 알림 창을 띄운다. 5. Portfolio Manager에 저장된 데이터를 수정한다.  * 세부사항 수정  1. User는 항목 창에서 수정할 세부사항을 선택한다. 2. User는 세부사항 내의 내용을 수정한다. 3. User는 세부사항 창에서 “세부사항 수정” 버튼을 누른다. 4. 재확인 알림 창을 띄운다. 5. “수정” 버튼 선택 시 Portfolio Manager에 저장된 데이터를 수정한다. |
| 대안  시나리오 | * 세부사항 타입 변경 시나리오  1. 세부사항 수정 2번 시나리오에서 “타입 수정” 버튼을 누른다. 2. 하위 데이터는 모두 제거된다는 알림 창을 띄운다. 3. “확인” 버튼 선택 시 하위 데이터를 제거한다. 4. UI 관리 기능의 UI 재배치 시나리오를 진행한다. 5. 데이터 타입 정의 시나리오를 진행한다. |

* + 1. FR-006: 항목 삭제

|  |  |
| --- | --- |
| 개요 | Portfolio Manager에 있는 자료를 생성/수정/삭제 가능하다. |
| 선행  조건 | * Portfolio Manager에 사용자 입력 데이터가 존재해야 한다. |
| 후행  조건 | * Portfolio Manager 데이터의 삭제가 정상적으로 처리된다. |
| 기본  시나리오 | * 항목 삭제  1. User는 메인 폼에서 삭제할 항목을 선택한다. 2. User는 항목 창에서 “항목 삭제” 버튼을 선택한다. 3. “삭제” 버튼 선택 시 Portfolio Manager에 저장된 선택 항목을 제거한다. 4. UI 관리 기능의 UI 재배치 시나리오를 진행한다.  * 세부사항 삭제  1. User는 항목 창에서 삭제할 세부사항을 선택한다. 2. User는 항목 창에서 “세부사항 삭제” 버튼을 선택한다. 3. 세부항목의 데이터 목록과 재확인 알림 창을 띄운다. 4. “삭제” 버튼 선택 시 Portfolio Manager에 저장된 세부사항 데이터를 삭제한다. 5. UI관리 기능의 UI 재배치 시나리오를 진행한다. |
| 대안  시나리오 | * 항목 세부사항 존재 시나리오  1. 항목 삭제 2번 시나리오 후에 하위 항목과 함께 재확인 알림 창을 띄운다 2. 항목 삭제 3번 시나리오 시 세부사항 데이터를 함께 제거한다. |

* + 1. FR-007: UI 관리

|  |  |
| --- | --- |
| 개요 | UI Form을 생성하거나 배치를 재설정 한다. |
| 선행  조건 | * Portfolio Manager에 사용자 입력 데이터가 존재해야 한다. |
| 후행  조건 | * 항목의 중요도와 개수에 따라 배치 순서가 달라진다. |
| 기본  시나리오 | * UI Form 생성  1. User는 항목 생성을 선택한다. 2. Portfolio Manager는 UI Manager에게 중요도와 세부사항 데이터, 전체 항목의 개수를 전송한다. 3. UI Form을 만들고 데이터를 입력한다. 4. UI 재배치 시나리오를 진행한다.  * UI 재배치  1. 생성되어 있는 UI Form을 중요도 순으로 정렬한다. 2. Window에 나타날 수 있는 항목의 최대 제한인 5개 기준으로 페이지를 나눈다. |
| 대안  시나리오 | * UI Form의 우선 순위가 같은 경우 * 먼저 생성된 항목을 우선적으로 배치한다. |

* + 1. FR-008: 파일 자동 저장

|  |  |
| --- | --- |
| 개요 | 파일 손실을 막기 위해 정해진 Timeout 시간마다 자동 저장을 실행한다. 저장이 이루어지는 위치는 마지막으로 저장된 로컬 경로이다. |
| 선행  조건 | * Portfolio Manager에 사용자 입력 데이터가 존재해야 한다. * Portfolio Manager의 사용자 입력 데이터를 적어도 한 번 저장해야 한다. * 저장은 백그라운드에서 실행된다. * Timeout 시간이 정의되어 있어야 한다. |
| 후행  조건 | * Local에 Portfolio Manager의 사용자 입력 데이터 파일이 업데이트 된다. |
| 기본  시나리오 | 1. File Manager가 Timeout이 되었음을 안다. 2. File Manager는 마지막으로 저장된 경로로 파일 관리의 저장 시나리오를 사용한다. |
| 대안  시나리오 | * 저장을 하지 않은 파일의 경우  1. File Manager는 임시파일 경로를 생성한다. 2. File Manager는 임시파일 경로로 파일관리 기능의 저장 시나리오를 사용한다. 3. 프로그램 종료 후 다시 시작하는 경우 저장하지 않은 데이터 불러오기 선택 창을 띄운다. 4. 해당 파일의 경우 하나의 파일만이 저장된다. |

* + 1. FR-009: 메모 작성

|  |  |
| --- | --- |
| 개요 | 각 데이터 별로 상세내용을 기록하기 쉽도록 메모를 작성할 수 있다. 이미 만들어져 있는 경우 수정 가능하다. |
| 선행  조건 | * Portfolio Manager에 사용자 입력 데이터가 존재해야 한다. |
| 후행  조건 | * 세부사항에 대한 메모 데이터가 Portfolio Manager에 생성/수정된다. * 생성/수정된 메모 데이터를 카드 설정 창에서 볼 수 있다. |
| 기본  시나리오 | 1. User는 항목 창에서 선택 가능한 세부사항 중 하나를 선택한다. 2. User는 “메모 작성” 버튼을 선택한다. 3. Portfolio Manager는 메모 작성 창을 띄운다. 4. User는 메모 작성 창에서 메모를 작성한다. 5. User는 “작성” 버튼을 선택한다. 6. Portfolio Manager는 메모를 생성한다. |
| 대안  시나리오 | * 메모가 생성되어 있는 경우  1. User는 만들어진 메모 창에서 내용을 수정한다. 2. User는 메모 창에 있는 “수정” 버튼을 선택한다. 3. Portfolio Manager는 메모 수정을 실행한다. |

* + 1. FR-010: 리포트 항목 선택

|  |  |
| --- | --- |
| 개요 | User는 메인 폼에서 원하는 데이터를 선택할 수 있다. |
| 선행  조건 | * Portfolio Manager에 사용자 입력 데이터가 존재해야 한다. |
| 후행  조건 | * User가 원하는 데이터를 선택한다. |
| 기본  시나리오 | 1. User는 원하는 항목을 선택한다. 2. 리포트를 출력한다. |
| 대안  시나리오 | * 선택된 항목이 없는 경우 * 기본 값으로 전체 선택을 한다. |

* + 1. FR-011: 데이터 타입 정의

|  |  |
| --- | --- |
| 개요 | 세부사항에 입력되는 데이터의 타입을 정의한다. 기존 정의된 타입을 수정할 경우 해당 항목의 데이터는 모두 제거된다. |
| 선행  조건 | * Portfolio Manger에 사용자 입력 데이터가 존재해야 한다. |
| 후행  조건 | * 세부사항에 새로운 데이터 타입이 정의 혹은 기존의 데이터가 수정된다. * Portfolio Manager의 사용자 입력 데이터에 세부사항이 생성 혹은 수정된다. |
| 기본  시나리오 | 1. User는 데이터 타입의 이름을 정의한다. 2. User는 텍스트, 숫자, 사진 중 하나를 데이터 타입으로 선택한다. |
| 대안  시나리오 | * 데이터 타입 변경 시나리오  1. Portfolio Manager는 해당 항목의 기존 데이터 삭제를 실행한다. 2. 기본 시나리오를 진행한다. |

1. **소프트웨어 품질 요구사항**
   1. 품질 요구사항 목록

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 품질  요구사항 ID | 유형 | 품질 요구사항 명 | 우선순위 | | |
| 중요도 | 난이도 | 우선  순위 |
| QR-001 | 기능성 | 원하는 데이터 입력 | 3 | 3 | 1 |
| QR-002 | 효율성 | 데이터 처리 속도 | 3 | 2 | 2 |
| QR-003 | 복구성 | 데이터 복구 | 2 | 1 | 3 |

* 1. 품질 요구사항 정의
     1. QR-001: 원하는 데이터 입력

|  |  |
| --- | --- |
| 항목 | 내용 |
| 설명 | 개인 이력 관리 기능에 대한 요구사항 |
| 전제 조건 | 사용자의 인적사항, 자격증, 이력 사항, 자기소개서 작성 등 개인의 이력을 기록 |
| 충족 기준 | 사용자가 원하는 기능, 항목별로 선택하여 기록 |

* + 1. QR-002: 데이터 처리 속도

|  |  |
| --- | --- |
| 항목 | 내용 |
| 설명 | 기록한 데이터들의 처리 속도에 대한 요구사항 |
| 전제 조건 | 항목은 수십 개가 있고 텍스트, 숫자, 사진 등 여러 가지 데이터가 있음 |
| 충족 기준 | 텍스트, 숫자, 사진 등 입력된 기록들을 바이너리 형식 파일로 최대 5초 내에 저장 |

* + 1. QR-003: 데이터 복구

|  |  |
| --- | --- |
| 항목 | 내용 |
| 설명 | 기록한 데이터들의 복구에 대한 요구사항 |
| 전제 조건 | 시스템의 결함, 다운, 예기치 못한 현상으로 인한 에러 |
| 충족 기준 | 5분 간격의 자동 저장 |

1. **인터페이스**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 인터페이스  요구사항 ID | 인터페이스  요구사항 명 | 인터페이스 요구사항 설명 | 우선순위 | | |
| 중요도 | 난이도 | 우선  순위 |
| IR-001 | 화면 표시 제한 | Window창에 표시되는 리스트의 항목 수는 5개로 제한 | 2 | 3 | 2 |
| IR-002 | 최소 사양 | 윈도우7 이상에서 작동 가능 | 1 | 1 | 3 |
| IR-003 | 저장 형식 | 직렬화 된 로컬 파일 형식으로 저장 | 3 | 2 | 1 |

1. **제약 사항**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 제약사항  ID | 제약사항  명 | 제약사항  설명 | 우선순위 | | |
| 중요도 | 난이도 | 우선  순위 |
| CR-001 | 코딩 규칙 준수 | Microsoft에서 제공하는 C# 코딩 규칙”C# Coding Conventions”을 준수하여 개발 | 2 | 2 | 1 |
| CR-002 | 사용자 정보 보호 | 개인정보 보호법에 따라 사용자의 인적사항 정보 보호 필요 | 1 | 1 | 2 |

* 참고사항: 코딩 규칙에 대한 참고 사항은 <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions>를 참고하도록 한다.