

화면설계(3)

학습 2 UI 설계하기

2-1. UI 흐름 설계

① UI 설계서 구성에 따른 작성 방법

UI 설계서 구성은 UI 설계서 표지, UI 설계서 개정 이력, UI 요구사항 정의, 시스템 구조, 사이트 맵, 프로세스 정의, 화면 설계 등으로 이루어진다.

1. UI 설계서 표지

UI 설계서에 포함될 프로젝트 명 또는 시스템 명을 포함시킨다.

2. UI 설계서 개정 이력

UI 설계서 처음 작성 시에는 첫 번째 항목으로 ‘초안 작성’을 포함시키고 그에 해당되는 초기 버전(version)을 1.0으로 설정한다. 변경 또는 보완이 충분히 이루어져 완성이 되었다고 판단할 경우 버전을 x.0 으로 바꾸어 설정한다.

3. UI 요구사항 정의

UI 요구사항들을 재확인하고 정리한다.

4. 시스템 구조

- UI 프로토타입을 재확인한다.
- UI 요구사항들과 UI 프로토타입에 기초해 UI 시스템 구조를 설계한다.

5. 사이트 맵(Site Map)

- UI 시스템 구조의 내용을 사이트 맵의 형태로 작성한다.
- 사이트 맵 상세 내용(Site Map Detail)을 표 형태로 작성한다.

6. 프로세스(Process) 정의

사용자 관점에서 요구되는 프로세스들을 진행되는 순서에 맞추어 정리한다.

7. 화면 설계

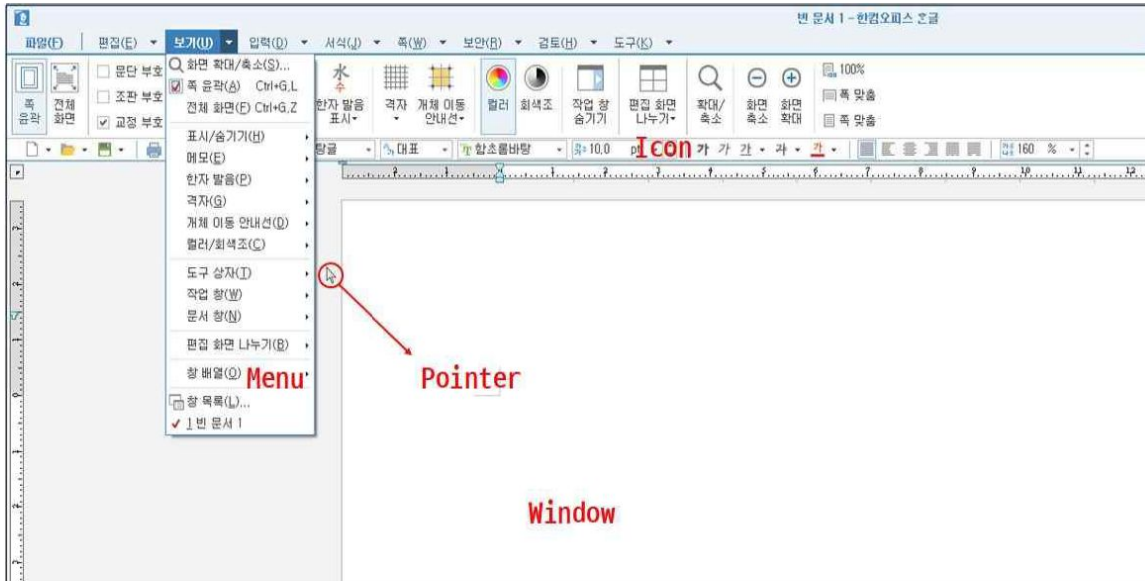
UI 프로토타입과 UI 프로세서 정의를 참고해 각 페이지별로 필요한 화면을 설계한다.

-각 화면별로 구분되도록 각 화면별 고유 ID를 부여하고 별도의 표지 페이지를 작성한다.

-각 화면별로 필요한 화면 내용을 설계한다.

2 UI 화면 설계의 기본 구성 요소

1. 윈도우(Window)
2. 메뉴(Menu)
3. 아이콘(Icon)
4. 포인터(Pointer)



[그림 2-1] UI 화면 설계의 기본 구성 요소: Window, Menu, Icon, Pointer

③ 유용성 개념 및 적용 원리 파악

1. 유용성 개념

(1) 유용성

사용자가 시스템을 통해 원하는 목표를 얼마나 효과적으로 달성할 수 있는가에 대한 척도이다.

(2) 유용한 시스템을 설계할 때 고려사항

사용자가 시스템의 구조, 기능, 가치 등에 대해 가지고 있는 마음 속 모형인 사용자 모형과 시스템 설계자가 만들려고 하는 개발자 모형 사이의 차이를 최소화하려는 노력이 필요하다.

(3) 사용자 모형과 개발자 모형 간의 차이 발생

- 시스템에서의 실행 차, 즉 실행 가능한 기능과 사용자의 원래 목적이 서로 상이하기 때문에 발생한다.

2. 실행 차를 줄이기 위한 UI 설계 원리

(1) 사용 의도 파악

- 사용자의 원래 목적을 명확히 파악하여 불필요한 부가 기능 때문에 시스템 성능이 떨어지지 않도록 해야 한다.
- 사용자가 보다 관심을 가지는 항목을 눈에 띄는 위치에 배치하고 적절한 시점에 해당 기능이 제공되도록 하여야 한다.

(2) 행위 순서 규정

- 사용자 행위의 순서를 세분화시킨 뒤 순서대로 명확하게 제시하여 선택할 수 있도록 해야 하고 또한 의도에 따라 행위의 순서를 사용자가 임의로 변경 가능하도록 해야 한다.
- 하나의 작업을 수행하기 위한 단계 수를 최소화시키고, 동일한 결과를 여러 가지 다른 방법으로도 달성 가능하도록 설계 시 고려해야 하며, 행위의 순서가 사용자의 기존 경험에 비추어 가능한 한 친숙하도록 설계해야 한다.

(3) 행위의 순서대로 실행

- 프로세스의 흐름을 UI를 통해 직관적으로 알 수 있게 제공함으로써 사용자가 의도한 행위의 순서를 실제 실행으로 옮기는 데 어려움을 최소화해야 한다.
- 과도한 상호 작용으로 인해 작업이 원활히 진행되지 못하는 상황이 발생되지 않도록 고려해야 한다.
- 사용자가 의도한 행위와 가장 잘 어울리는 입력 장치를 선택하고, 사용자의 행위에 대해 적절한 피드백과 취소 기능을 제공해 주며, 디폴트 값을 적절하게 설정함으로써 불필요한 조작을 최소화하여 사용자가 의도한 행위를 효율적으로 실행할 수 있도록 설계해야 한다.

3. 평가 차를 줄이기 위한 UI 설계 원리

(1) 수행한 키 조작의 결과를 사용자가 빠르게 지각하도록 유도

- 사용자가 수행한 행위에 대해 아무런 변화된 결과가 제공되지 않는다면 사용자는 자신이 제대로 조작하였는지 의심하게 되므로, 이러한 평가 차가 발생하지 않도록 설계해야 한다.
- 가능한 한 빠른 처리를 통해 즉각적으로 반응되도록 해야 하며, 즉각적인 반응이 힘들더라도 가능한 한 반응 속도를 높이도록 노력해야 한다. 또한 사용자가 수행한 행위로 인해 현재 시스템의 변화가 이루어졌음을 가능한 한 직관적으로 피드백해 주어야 한다.

(2) 키 조작으로 변화된 시스템의 상태를 사용자가 쉽게 인지하도록 유도

사용자가 수행한 행위로 인해 변화된 시스템의 상태를 사용자가 직관적으로 인지할 수 있도록 시스템을 설계해야 한다. 이를 위해 시스템의 상태 정보를 가능한 한 단순하게, 그리고 이해하기 쉽게 제시해야 한다.

(3) 사용자가 가진 원래 의도와 시스템 결과 간의 유사 정도를 사용자가 쉽게 파악하도록 유도

사용자가 가진 원래 의도가 시스템을 통해 충족되었는지 또는 충족될 수 있는지를 사용자가 쉽게 파악할 수 있도록 설계해야 한다. 이를 위해 미리보기 기능처럼 예상 결과를 사전에 제시할 수 있다면 제공해 주는 것이 대부분의 경우 바람직하다.

4 스토리보드 작성 기법

스토리보드는 디자이너와 개발자가 최종적으로 참고하는 산출 문서이며, 정책이나 프로세스 및 콘텐츠의 구성, 와이어프레임(UI, UX), 기능에 대한 정의, 데이터베이스의 연동 등 구축하는 서비스를 위한 대부분의 정보가 수록되어 있는 문서이다.

1. 메뉴 구성도 만들기(스토리보드 1단계)

전체적인 메뉴 구성도이며, 어떤 것을 보여주고 결정된 사항을 표현하기 위한 메뉴의 순서와 구성 단계, 용어를 정의한다.

2. 스타일 확정(스토리보드 2단계)

레이아웃이나 글자 모양, 크기, 색상, 그래픽에서의 일관성을 유지해야 한다.

3. 설계하기(스토리보드 3단계)

화면에 보여지는 시각적인 디자인 콘셉트를 잡는다.

수행 순서

① 유용성을 적용한 UI 설계안의 적정성을 확인한다.

1. 실행 차를 줄이기 위한 UI 설계 원리를 확인한다.

(1) 사용 의도를 정확히 확인한다.

(가) UI 설계서를 통해 각 화면 설계 결과에서 불필요한 부가 기능이 있는지 찾아본다.

(나) UI 설계서를 통해 각 화면 설계 결과에서 중복되는 기능이 있는지 찾아본다.

(2) 행위 순서를 규정한다.

(가) UI 설계서를 통해 사용자가 도서 대출 예약 시스템의 각 기능을 사용하기 위해 어떤 사전 행위들이 수행되어야 하는지 나열한다.

(나) UI 설계서를 통해 각 기능들의 상대적 중요도를 상, 중, 하로 나눈다.

(다) 각 기능을 사용하기 위해 필요한 사전 행위들을 바탕으로 중요도가 상 등급인 기능들이 수행되기 위해 필요한 단계 수가 최소화되어 있는지 확인한다.

(라) 전반적으로 기능을 사용하기 위해 필요한 단계 수가 적절한지 확인한다.

② UI 요구사항과 UI 표준 및 지침에 따라, 화면과 폼의 흐름을 설계한다.

1. 화면에 표현되어야 할 기능을 작성한다.

(1) 기능적 요구사항을 검토한다.

(가) 구축할 시스템에서 각각의 기능적 요구사항이 무엇일지 정리한다.

- 1) 시스템의 입력으로 무엇이 포함되어야 하는지 분석한다.
- 2) 시스템의 출력으로 무엇이 포함되어야 하는지 분석한다.
- 3) 시스템이 어떤 데이터를 저장해야 하는지 분석한다.
- 4) 시스템이 어떤 연산을 수행해야 하는지 분석한다.
- 5) 기타 요구사항(예: 동기화 등) 으로 어떤 것들이 있는지 조사한다.

(나) 구축할 시스템에서 각각의 기능적 요구사항의 설명을 정리한다.

- 1) 회원 등록, 삭제, 수정 사항에 대해 정리한다.
- 2) 이벤트 발생 시 필요한 기능에 대해 숙지한다.
- 3) 검색은 쉽고 빠르게 가능한지 조사한다.
- 4) 입력 및, 출력 기능에 대해 검토한다.
- 5) 특정 시스템의 기능에 대해 조사한다.
- 6) 기타 사용자의 편의성을 높이는 기능에 대해 조사한다.
- 7) 메모 내용의 전체 선택이 가능한지 파악한다.

(가) 구축할 시스템에서 각각의 비기능적 요구사항이 무엇일지 정리한다.

- 1) 사용성, 효율성, 신뢰성, 유지 보수성, 재사용성 등 품질에 관한 요구사항으로 어떤 것들이 있는지 검토한다.
- 2) 플랫폼, 사용 기술 등 시스템 환경에 관한 요구사항으로 어떤 것들이 있는지 분석한다.
- 3) 비용, 일정 등 프로젝트 계획에 관한 요구사항으로 어떤 것들이 있는지 조사한다.

(나) 구축할 시스템에서 각각의 비기능적 요구사항의 설명을 정리한다.

- 1) 구축할 시스템은 운영 체제에 대해 종속적이지 않게 작동이 가능한지 분석한다.
- 2) 일부 오류로 인해 구축할 시스템 전체가 작동 불능 상태로 전환하는지 분석한다.
- 3) 데이터베이스관리시스템(Data Base Management System : DBMS)은 안정적인지 검토한다.
- 4) 이벤트 발생 시 처리 속도는 2초가 넘지 않도록 측정한다.
- 5) 사용자의 제약사항을 검토한다.

2. 화면의 입력 요소를 확인한다.

(1) 화면에 표현되어야 할 기능을 확인한다.

도서 대출 예약 시스템의 화면 표현 기능 확인 사례

(가) 사용자가 자신의 계정인 아이디와 패스워드를 입력할 수 있어야 한다.

(나) 시스템이 도서 탐색 화면을 제공해 주어야 한다.

(다) 시스템이 색인 정보를 제공해 주어야 한다.

(라) 사용자가 책 이름을 입력할 수 있어야 한다.

(마) 시스템은 일치하는 책을 검색하여 사용자가 선택 가능한 리스트를 제공해 주어야 한다.

(바) 사용자가 책을 선택할 수 있어야 한다.

(사) 시스템이 예약 여부를 제공해 줄 수 있어야 한다.

(아) 예약이 이루어진 경우 예약 내역 또는 대출 내역이 색인에 반영되어야 한다.

(2) 화면의 입력 요소를 확인한다.

도서 대출 예약 시스템의 화면 입력 요소 확인 사례

- (가) 사용자 계정 아이디 입력
- (나) 사용자 계정 패스워드 입력
- (다) 책 이름 입력

(3) 추가적으로 필요한 화면 요소를 확인한다.

도서 대출 예약 시스템의 추가 화면 요소 확인 사례

- (가) 도서 탐색 화면
- (나) 도서명, 색인 정보
- (다) 도서 리스트
- (라) 대출 가능 / 대출 중 표시

(4) 기능을 표현하기 위해 필요한 페이지를 확인한다.

도서 대출 예약 시스템의 기능 표현 페이지 확인 사례

(가) 로그인 화면

(나) 도서검색 화면

(5) 각 화면 간 이동과 흐름을 확인한다.

도서 대출 예약 시스템의 화면 간 이동 흐름 확인 사례

(가) 로그인 성공 시 도서검색 화면으로 이동한다.

(나) 로그아웃 시 로그인 화면으로 다시 이동한다.

3. 유스케이스를 통한 UI 요구사항을 확인한다.

(1) 화면에 표현되어야 할 기능을 확인한다.

도서 대출 예약 시스템의 화면 표현 기능 확인 사례

앞서 작성한 도서 대출 예약 시스템 중 도서 반납에 관한 사용 사례를 바탕으로 화면에 표현되어야 할 기능들을 확인한다.

(2) 화면의 입력 요소를 확인한다.

도서 대출 예약 시스템의 화면 입력 요소 확인 사례

앞서 작성한 도서 대출 예약 시스템 중 도서 반납에 관한 사용 사례를 바탕으로 화면의 입력 요소들을 확인한다.

(3) 추가적으로 필요한 화면 요소를 확인한다.

도서 대출 예약 시스템의 추가 필요 화면 요소 확인 사례

앞서 작성한 도서 대출 예약 시스템 중 도서 반납에 관한 사용 사례를 바탕으로 추가적으로 필요한 화면 요소들을 확인한다.

(4) 기능을 표현하기 위해 필요한 페이지를 확인한다.

도서 대출 예약 시스템의 기능 표현 페이지 확인 사례

앞서 작성한 도서 대출 예약 시스템 중 도서 반납에 관한 사용 사례를 바탕으로 각 기능들을 표현하기 위해 필요한 페이지들을 확인한다.

(5) 각 화면 간 이동과 흐름을 확인한다.

도서 대출 예약 시스템의 화면 간 이동 흐름 사례

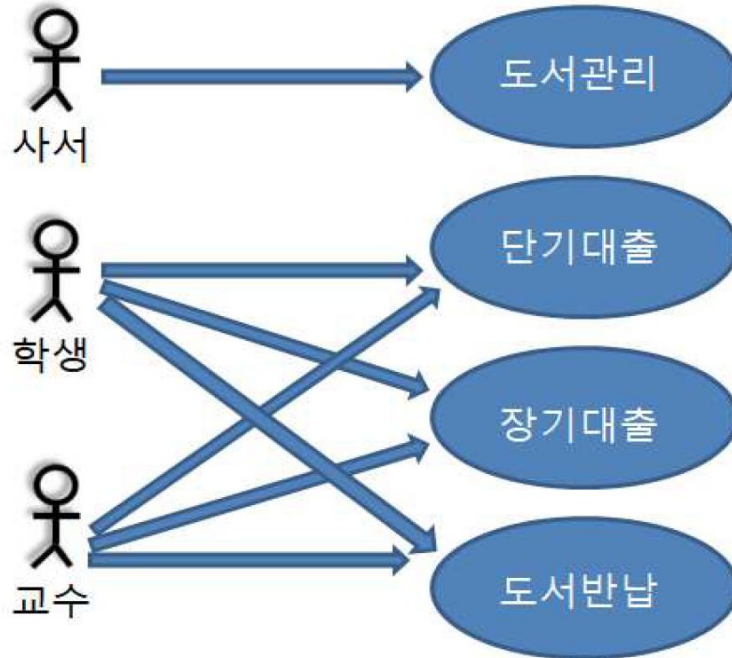
앞서 작성한 도서 대출 예약 시스템 중 도서 반납에 관한 사용 사례를 바탕으로 각 화면 간 이동과 흐름들을 확인한다.

4. 유스케이스를 설계한다.

앞서 작성한 시스템 중 기능에 관한 사용 사례를 바탕으로 파워포인트 또는 한글 프로그램을 사용하여 UI 유스케이스 설계를 수행한다.

(1) UI 요구사항을 바탕으로 액터별 시나리오를 구상한다.

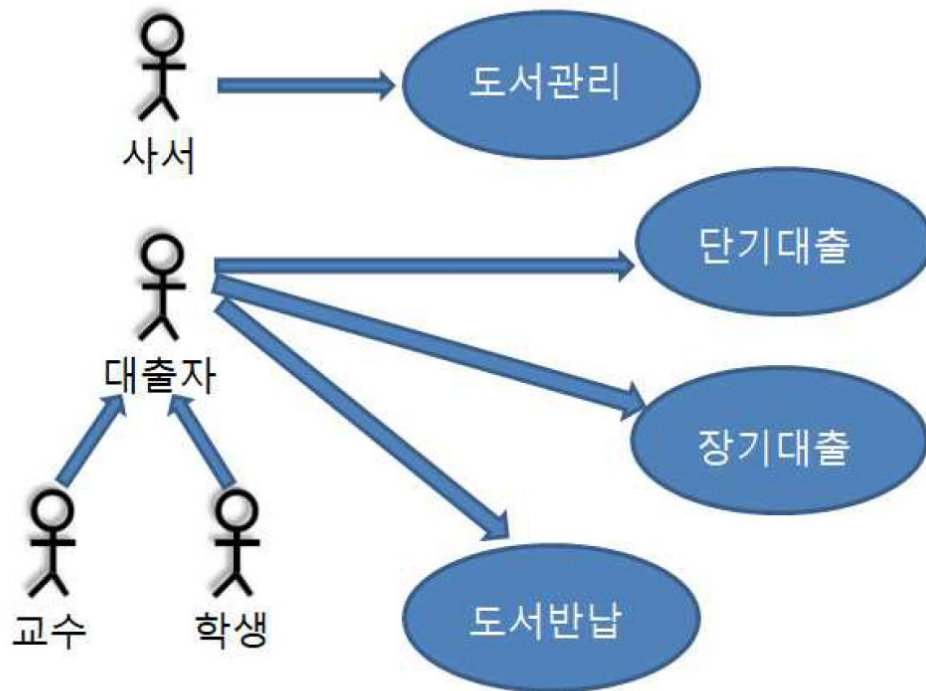
각각의 액터가 어떤 행위를 하는지 작성해 본다.



[그림 2-2] 액터의 행위 사례(도서 관리 시스템)

(2) UI 요구사항을 바탕으로 액터를 세분화한다.

액터의 상호 작용을 생각하고 비슷한 행위를 하는 액터를 그룹으로 지정하여 부모 액터를 정의한다. 아래의 [그림 2-3]은 도서 관리 시스템에서 대출자 액터가 학생과 교수의 부모 액터임을 나타내는 사례이다.



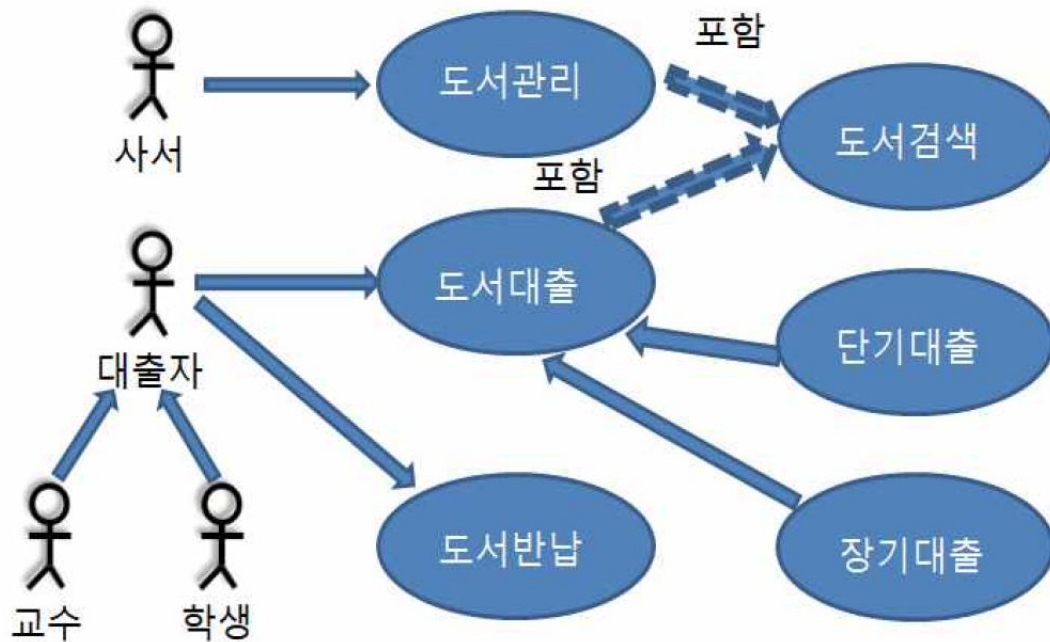
[그림 2-3] 액터의 관계 사례(도서 관리 시스템)

(3) 유스케이스를 설계한다.

(가) 각 입력 요소들의 형태와 입력 방법에 대해서 구상한다.

(나) 각 화면 요소들의 배치에 대해서 구상한다.

(다) 각 요소들을 설계한다.



[그림 2-4] 액터의 확장 사례(도서 관리 시스템)

5. 기능 및 양식(Form)을 확인한다.

(1) Input Box를 확인하고, 규칙을 정의한다.

아이디	<input type="text"/>
-----	----------------------

아이디	<input type="text"/>
-----	----------------------

*아이디	<input type="text"/>
------	----------------------

[그림 2-5] Input Box