

대분류/20
정보통신

중분류/01
정보기술

소분류/02
정보기술개발

세분류/02
응용SW엔지니어링

능력단위/24

NCS학습모듈

화면 설계

LM2001020224_16v4



교육부

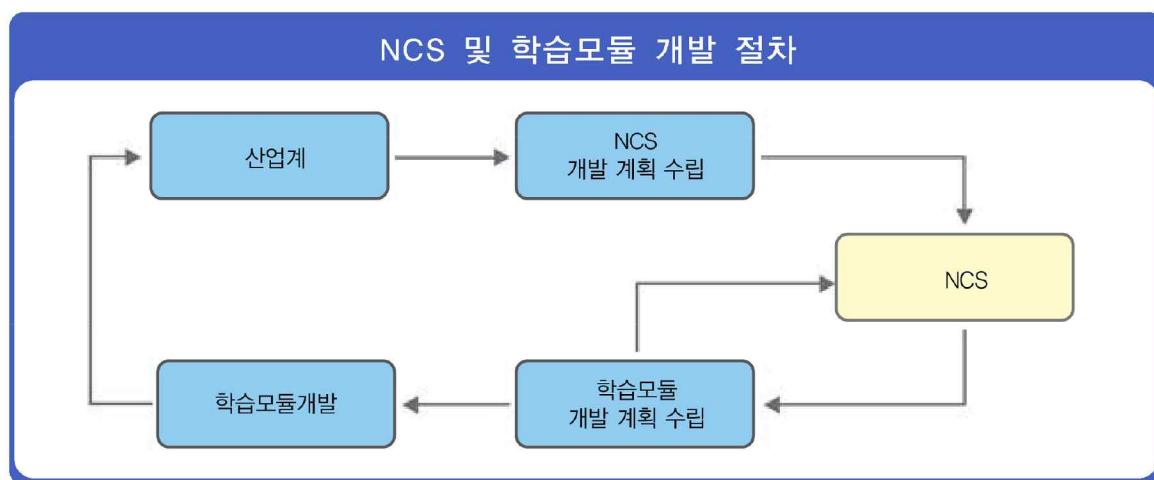
NCS 학습모듈은 교육훈련기관에서 출처를 명시하고 교육적 목적으로 활용할 수 있습니다. 다만 NCS 학습모듈에는 국가(교육부)가 저작재산권 일체를 보유하지 않은 저작물들(출처가 표기되어 있는 도표, 사진, 삽화, 도면 등)이 포함되어 있으므로 이러한 저작물들의 변형, 복제, 공연, 배포, 공중 송신 등과 이러한 저작물들을 활용한 2차 저작물의 생성을 위해서는 반드시 원작자의 동의를 받아야 합니다.

NCS 학습모듈의 이해

* 본 학습모듈은 「NCS 국가직무능력표준」 사이트(<http://www.ncs.go.kr>)에서 확인 및 다운로드 할 수 있습니다.

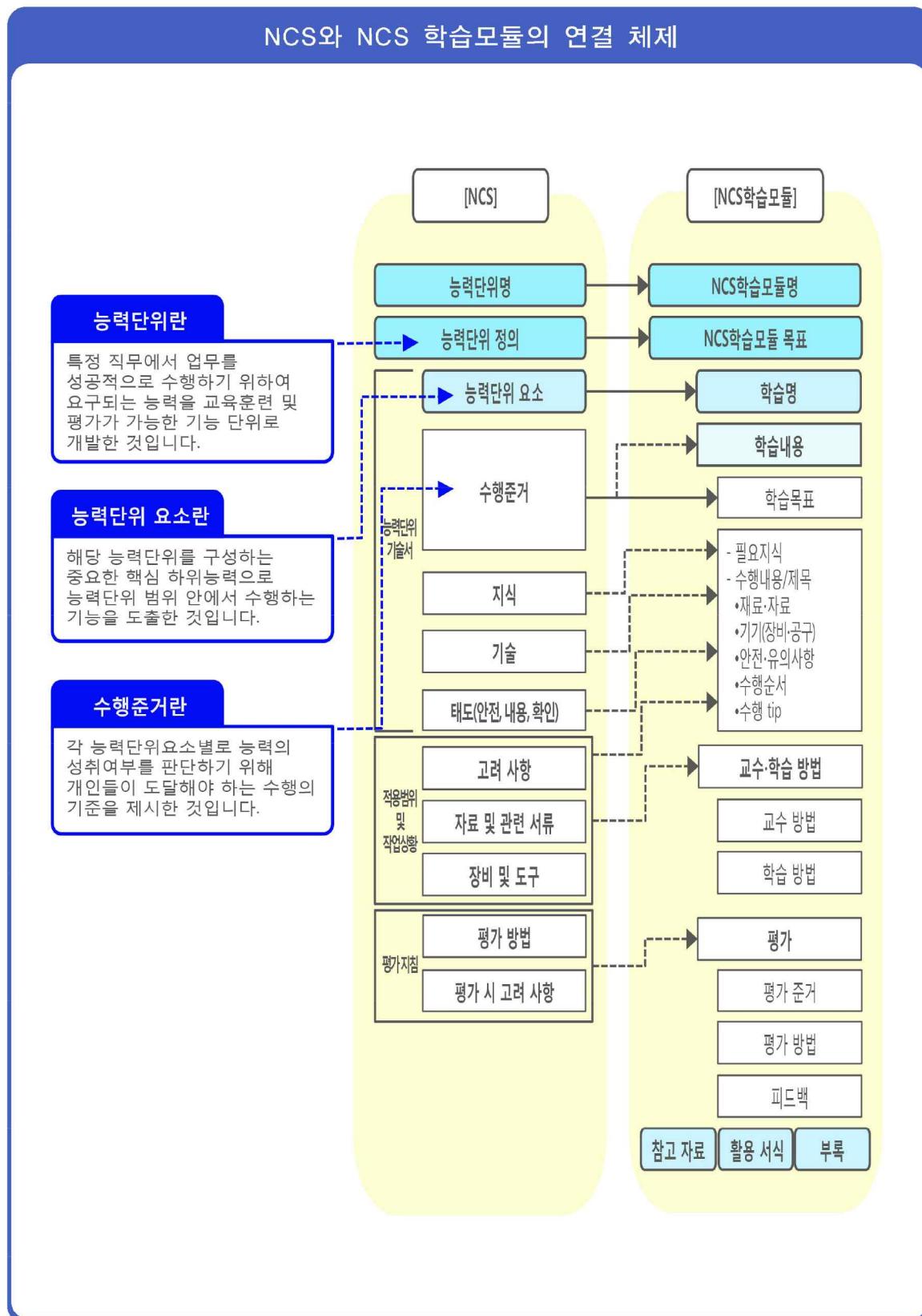
(1) NCS 학습모듈이란?

- 국가직무능력표준(NCS: National Competency Standards)이란 산업현장에서 직무를 수행하기 위해 요구되는 지식·기술·소양 등의 내용을 국가가 산업부문별·수준별로 체계화한 것으로 산업현장의 직무를 성공적으로 수행하기 위해 필요한 능력(지식, 기술, 태도)을 국가적 차원에서 표준화한 것을 의미합니다.
- 국가직무능력표준(이하 NCS)이 현장의 ‘직무 요구서’라고 한다면, NCS 학습모듈은 NCS의 능력단위를 교육훈련에서 학습할 수 있도록 구성한 ‘교수·학습 자료’입니다. NCS 학습모듈은 구체적 직무를 학습할 수 있도록 이론 및 실습과 관련된 내용을 상세하게 제시하고 있습니다.



- NCS 학습모듈은 다음과 같은 특징을 가지고 있습니다.
- 첫째, NCS 학습모듈은 산업계에서 요구하는 직무능력을 교육훈련 현장에 활용할 수 있도록 성취목표와 학습의 방향을 명확히 제시하는 가이드라인의 역할을 합니다.
- 둘째, NCS 학습모듈은 특성화고, 마이스터고, 전문대학, 4년제 대학교의 교육기관 및 훈련기관, 직장교육기관 등에서 표준교재로 활용할 수 있으며 교육과정 개편 시에도 유용하게 참고할 수 있습니다.

- NCS와 NCS 학습모듈 간의 연결 체계를 살펴보면 아래 그림과 같습니다.



(2) NCS 학습모듈의 체계

- NCS 학습모듈은 1.학습모듈의 위치, 2.학습모듈의 개요, 3.학습모듈의 내용 체계, 4.참고 자료, 5.활용 서식/부록으로 구성되어 있습니다.

1. NCS 학습모듈의 위치

- NCS 학습모듈의 위치는 NCS 분류 체계에서 해당 학습모듈이 어디에 위치하는지를 한 눈에 볼 수 있도록 그림으로 제시한 것입니다.

예시 : 이·미용 서비스 분야 중 네일미용 세분류

NCS-학습모듈의 위치

대분류	이용·숙박·여행·오락·스포츠
중분류	이·미용
소분류	아·미용 서비스

세분류

헤어미용

피부미용

메이크업

네일미용

이용

능력단위('17.4. 고시)

네일 샵 위생 서비스

네일 화장물 제거

네일 기본 관리

네일 렌

네일 팁

젠퐁 네일

아크릴릭 네일

평면 네일아트

융합 네일아트

네일 샵 운영관리

학습모듈명

네일숍 위생서비스

네일 화장물 제거

네일 기본관리

네일 렌

네일 팁

젠퐁 네일

아크릴릭 네일

평면 네일아트

융합 네일아트

네일숍 운영관리

학습모듈은

NCS 능력단위 1개당 1개의 학습모듈 개발을 원칙으로 합니다. 그러나 필요에 따라 고용 단위 및 교과 단위를 고려하여 능력단위 몇 개를 묶어서 1개의 학습모듈로 개발할 수 있으며, NCS 능력단위 1개를 여러 개의 학습모듈로 나누어 개발할 수도 있습니다. 또한 보완 개발 세분류의 경우 일부 보완된 능력단위에 한해 학습모듈을 보완 개발하는 경우가 있으며, 이는 학습모듈명 우측에 있는 *표기로 확인할 수 있습니다.

2. NCS 학습모듈의 개요



구성

- NCS 학습모듈 개요는 학습모듈이 포함하고 있는 내용을 개략적으로 설명한 것으로서 **학습모듈의 목표**, **선수 학습**, **학습모듈의 내용 체계**, **핵심 용어**로 구성되어 있습니다.

학습모듈의 목표

해당 NCS 능력단위의 정의를 토대로 학습목표를 작성한 것입니다.

선수 학습

해당 학습모듈에 대한 효과적인 교수·학습을 위하여 사전에 이수해야 하는 학습모듈, 학습 내용, 관련 교과목 등을 기술한 것입니다.

학습모듈의 내용 체계

해당 NCS 능력단위요소가 학습모듈에서 구조화된 방식을 제시한 것입니다.

핵심 용어

해당 학습모듈의 학습 내용, 수행 내용, 설비·기자재 등 가운데 핵심적인 용어를 제시한 것입니다.



활용 안내

예시 : 네일미용 세분류의 ‘네일 기본관리’ 학습모듈

네일 기본관리 학습모듈의 개요

학습모듈의 목표

고객의 네일 보호와 미적 요구 충족을 위하여 효과적인 네일 관리로 프리에지 형태 만들기, 큐티를 정리하기, 컬러링하기, 보습제 도포하기, 마무리를 할 수 있다.

선수학습

네일숍 위생서비스(LM1201010401_14V2)

학습모듈의 내용체계

학습	학습 내용	NCS 능력단위 요소	
		코드번호	요소 명칭
1. 프리에지 형태 만들기	1-1. 네일 파일에 대한 이해와 활용 1-2. 프리에지 형태 파일링	1201010403_12v2.1	프리에지 모양 만들기
2. 큐티를 정리하기	2-1. 네일 기본관리 매뉴얼 이해 2-2. 큐티를 관리	1201010403_14v2.2	큐티를 정리하기
3. 컬러링하기	3-1. 컬러링 매뉴얼 이해 3-2. 컬러링 방법 선정과 작업 3-3. 웰 컬러링 작업	1201010403_14v2.3	컬러링
4. 보습제 도포하기	4-1. 보습제 선정과 도포 4-2. 각질제거	1201010403_14v2.4	보습제 바르기
5. 네일 기본관리 마무리하기	5-1. 유분기 제거 5-2. 네일 기본관리 마무리와 정리	1201010403_14v2.5	마무리하기

학습모듈의 목표는

학습자가 해당 학습모듈을 통해 성취해야 할 목표를 제시한 것으로, 교수자는 학습자가 학습모듈의 전체적인 내용흐름을 파악할 수 있도록 지도하는 것이 필요합니다.

선수 학습은

교수자나 학습자가 해당 모듈을 교수 또는 학습하기 이전에 이수해야 할 학습내용, 교과목, 핵심 단어 등을 표기한 것입니다. 따라서 교수자는 학습자가 개별 학습, 자기 주도 학습, 방과 후 활동 등 다양한 방법을 통해 이수할 수 있도록 지도하는 것이 필요합니다.

핵심 용어는

학습모듈을 통해 학습되고 평가되어야 할 주요 용어입니다. 또한 당해 모듈 또는 타 모듈에서도 핵심 용어를 사용하여 학습내용을 구성할 수 있으며, 「NCS 국가 직무능력표준」사이트(www.ncs.go.kr)에서 색인(찾아보기) 중 하나로 이용할 수 있습니다.

핵심 용어

프리에지, 니퍼, 퓨셔, 폴리시, 네일 파일, 스웨어형, 스웨어 오프형, 라운드형, 오발형, 포인트형

3. NCS 학습모듈의 내용 체계



- NCS 학습모듈의 내용은 크게 **학습**, **학습 내용**, **교수·학습 방법**, **평가**로 구성되어 있습니다.

학습

해당 NCS 능력단위요소 명칭을 사용하여 제시한 것입니다.

학습은 크게 학습 내용, 교수·학습 방법, 평가로 구성되며 해당 NCS 능력단위의 능력단위 요소별 지식, 기술, 태도 등을 토대로 학습 내용을 제시한 것입니다.

학습 내용

학습 내용은 학습 목표, 필요 지식, 수행 내용으로 구성하였으며, 수행 내용은 재료·자료, 기기(장비·공구), 안전·유의 사항, 수행 순서, 수행 tip으로 구성한 것입니다. 학습모듈의 학습 내용은 업무의 표준화된 프로세스에 기반을 두고 실제 산업현장에서 이루어지는 업무활동을 다양한 방식으로 반영한 것입니다.

교수·학습 방법

학습 목표를 성취하기 위한 교수자와 학습자 간, 학습자와 학습자 간의 상호작용이 활발하게 일어날 수 있도록 교수자의 활동 및 교수 전략, 학습자의 활동을 제시한 것입니다.

평가

평가는 해당 학습모듈의 학습 정도를 확인할 수 있는 평가 준거, 평가 방법, 평가 결과의 피드백 방법을 제시한 것입니다.



예시 : 네일미용 세분류의 ‘네일 기본관리’ 학습모듈의 내용

학습 1	프리에지 형태 만들기(LM1201010403_14v2.1)
학습 2	큐티클 정리하기(LM1201010403_14v2.2)
학습 3 컬러링하기(LM1201010403_14v2.3)	
학습 4	보습제 도포하기(LM1201010403_14v2.4)
학습 5	네일 기본관리 미무리하기(LM1201010403_14v2.5)

학습은

해당 NCS 능력단위요소 명칭을 사용하여 제시하였습니다. 학습은 일반교과의 ‘대단원’에 해당되며, 모듈을 구성하는 가장 큰 단위가 됩니다. 또한 완성된 직무를 수행하기 위한 가장 기초적인 단위로 사용할 수 있습니다.

학습내용은

요소 별 수행준거를 기준으로 제시하였습니다. 일반교과의 ‘중단원’에 해당합니다.

학습목표는

모듈 내의 학습내용을 이수했을 때 학습자가 보여줄 수 있는 행동수준을 의미합니다. 따라서 일반 수업시간의 과목목표로 활용할 수 있습니다.

3-1. 컬러링 매뉴얼 이해

학습목표

- 고객의 요구에 따라 네일 폴리시 색상의 침착을 막기 위한 베이스코트를 아주 얇게 도포할 수 있다.
- 작업 매뉴얼에 따라 네일 폴리시를 일plex 없이 균일하게 도포할 수 있다.
- 작업 매뉴얼에 따라 네일 폴리시 도포 후 컬러 보호와 광택 부여를 위한 톱코트를 바를 수 있다.

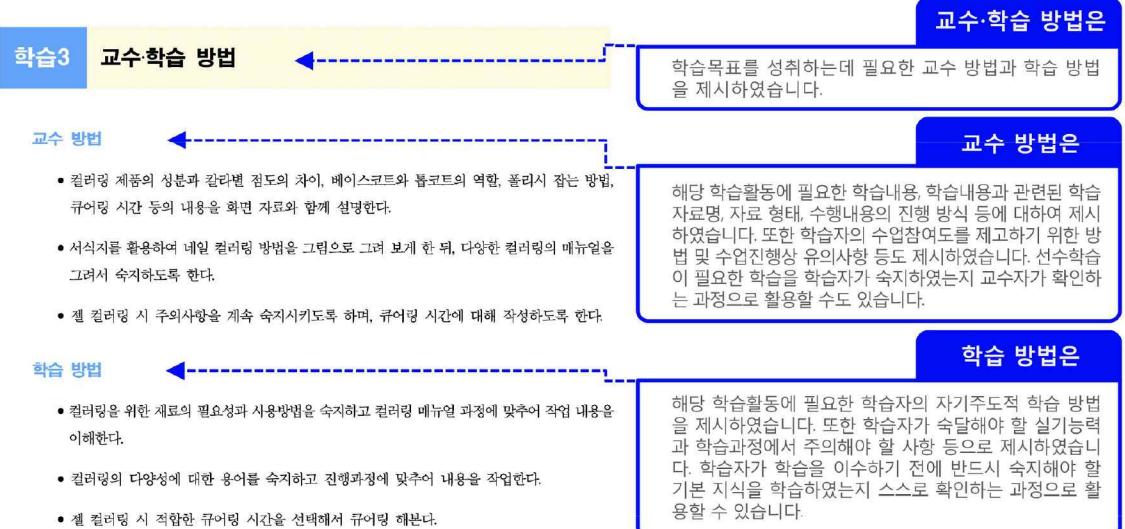
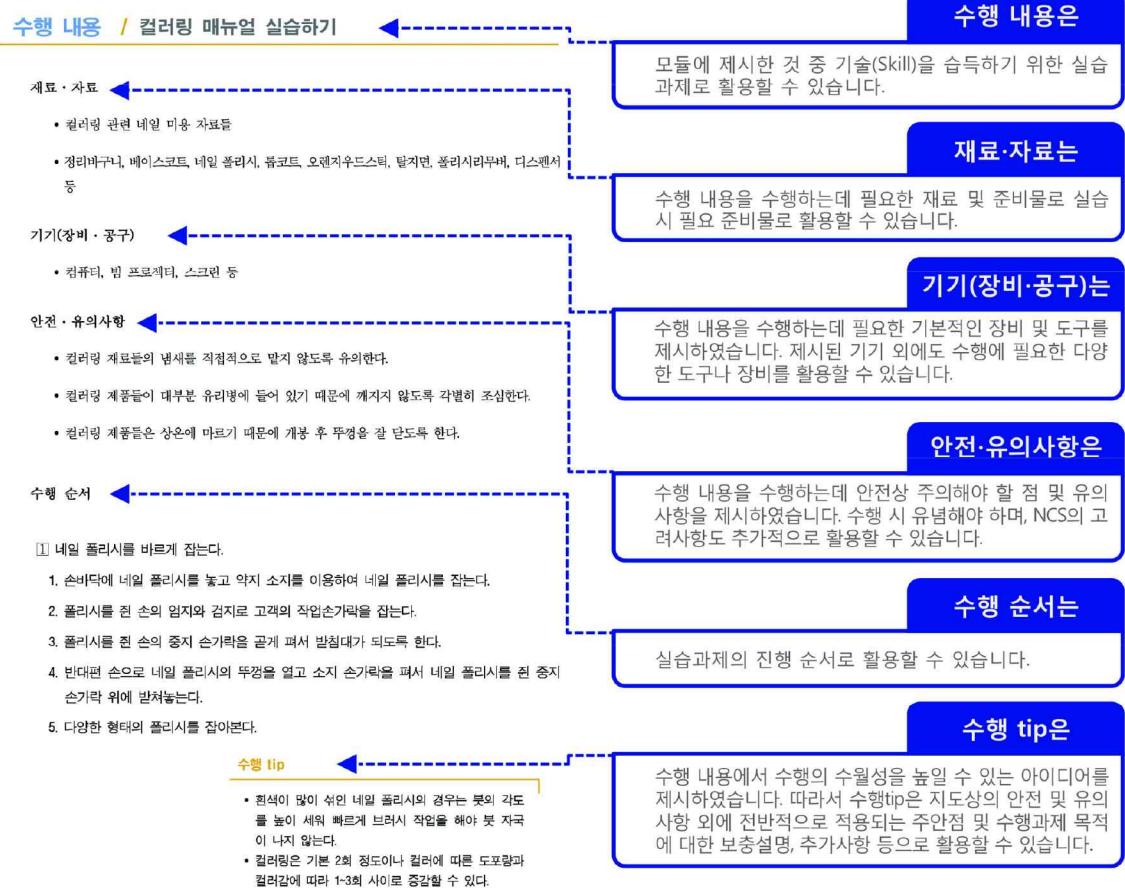
필요 지식 /

① 컬러링 매뉴얼

컬러링 작업 전, 아세톤 또는 네일 폴리시 리무버를 사용하여 손톱표면과 큐티클 주변, 손톱 일부 부분까지 깨끗하게 유분기를 제거해야 한다. 컬러링의 순서는 Base coating 1회 → Polishing 2회 → 컬러수정 → Top coating 1회 → 최종수정의 순서로 한다. 베이스코트는 착색을 방지하고 빌릴성 향상을 위해 가장 먼저 도포하며 컬러링의 마지막에 컬러의 유지와 광택을 위해 톱코트를 도포한다. 네일 보강제(Nail Strengthener)를 바를 시에는 베이스코트를 도포하기 전에 사용한다.

필요지식은

해당 NCS의 지식을 토대로 해당 학습에 대한 이해와 성과를 높이기 위해 알아야 할 주요 지식을 제시하였습니다. 필요지식은 수행에 꼭 필요한 핵심 내용을 위주로 제시하여 교수자의 역할이 매우 중요하며, 이후 수행순서 내용과 연계하여 교수·학습으로 진행할 수 있습니다.



학습3 평가

평가 준거

- 평가는 학습자가 학습 목표를 성공적으로 달성하였는지를 평가해야 한다.

- 평가는 다음 사항을 평가해야 한다.

학습내용	평가 항목	성취수준
		상 중 하
컬러링 매뉴얼 이해	- 고객의 요구에 따라 네일 폴리시 색상의 질적을 막기 위한 베이스코트를 아주 얇게 도포할 수 있다.	
	- 작업 매뉴얼에 따라 네일 폴리시를 얼룩 없이 균일하게 도포할 수 있다.	
	- 작업 매뉴얼에 따라 네일 폴리시 도포 후 컬러 보호 와 갈색 부여를 위한 들크트를 바를 수 있다.	

평가 방법

- 작업장 평가

학습내용	평가 항목	성취수준
		상 중 하
컬러링 매뉴얼 이해	- 고객의 요구에 따라 네일 폴리시 색상의 질적을 막기 위한 베이스코트를 아주 얇게 도포할 수 있다.	
	- 작업 매뉴얼에 따라 네일 폴리시를 얼룩 없이 균일하게 도포할 수 있다.	
	- 작업 매뉴얼에 따라 네일 폴리시 도포 후 컬러 보호 와 갈색 부여를 위한 들크트를 바를 수 있다.	

피드백

1. 작업장 평가

- 작업 결과물을 확인하여 수정사항을 제시하고 수정 부분을 인지하도록 한다.

4. 참고 자료

참고 자료

• 김미원(2011).『Nail Study』. 서울: 사)한국네일자격서비스협회.

• 민방정(2015).『비용사(네일)평가』. 서울: 예문사.

• 박온주(2014).『네일미용』. 서울: 정담미디어.

5. 활용 서식/부록

활용서식

프리에지 형태 실습지

1. 프리에지 형태의 이해

모양	이름	특징
	() Square nail	강한 느낌의 사각형태 네일의 양끝 모서리 부분이 90° 사각의 형태이다. () 발톱의 형태 활용 내인성 발톱의 보정시에 적용

평가는

해당 NCS 능력단위 평가방법과 평가 시 고려 사항을 준용하여 작성하였습니다. 교수자 및 학습자가 평가항목 별 성취수준을 확인하는데 활용할 수 있습니다.

평가 준거는

학습자가 해당 학습을 어느 정도 성취하였는지를 평가하기 위한 기준을 제시하고 있습니다. 학습목표와 연계하여 단위수업 시간에 평가항목 별 성취수준을 평가하는데 활용할 수 있습니다.

평가 방법은

NCS 능력단위의 평가방법을 준용하였으며, 평가 준거에 따른 평가방법을 2개 이상 제시하였습니다. 평가방법으로는 포트폴리오, 문제해결 시나리오, 서술형 시험, 논술형 시험, 사례연구, 평가자 체크리스트, 작업장 평가 등이 있으며, NCS의 능력단위 요소 별 수행 수준을 평가하는데 가장 적절한 방법을 선정하여 활용할 수 있습니다.

피드백은

평가 후에 학습자들에게 평가 결과를 피드백하여 부족한 부분을 알려주고, 학습 결과가 미진한 경우, 해당 부분을 다시 학습하여 학습목표를 달성하는 데 활용할 수 있습니다.

참고자료는

해당 학습모듈의 필요지식에 대한 출처와 인용한 참고자료 및 사이트를 제시하였습니다.

활용서식은

평가 서식, 실습시트 등 교수학습 시 활용 가능한 다양한 서식들로 구성하였습니다. 과제 진행에서 평가에 이르기까지 필요한 서식을 해당 학습모듈의 특성에 맞춰 개발하거나 기존의 양식을 활용하여 제시하였습니다.

부록

네일 기본관리 도구와 재료 목록

목록	비고	준비
위생가운	흰색	작업자 착용
위생 마스트	흰색	작업자 착용
보호안경	투명한 렌즈 (안경으로 대체 가능)	작업자 착용
세로정리합	제질, 색상 무관	작업대

부록은

활용서식 이외에 교수학습과정에서 참고할 수 있는 자료가 있는 경우 제시하였습니다.

[NCS-학습모듈의 위치]

대분류	정보 통신
중분류	정보 기술
소분류	정보 기술 개발

세분류	능력단위	학습모듈명
SW아키텍처	요구사항 확인	요구사항 확인
응용SW 엔지니어링	데이터 입출력 구현	데이터 입출력 구현
임베디드SW 엔지니어링	통합 구현	통합 구현
DB엔지니어링	정보시스템 이행	정보시스템 이행
NW엔지니어링	제품소프트웨어 패키징	제품소프트웨어 패키징
보안엔지니어링	서버프로그램 구현	서버프로그램 구현
UI/UX엔지니어링	인터페이스 구현	인터페이스 구현
시스템SW 엔지니어링	애플리케이션 배포	애플리케이션 배포
빅데이터 플랫폼구축	프로그래밍 언어 활용	프로그래밍 언어 활용
핀테크 엔지니어링	응용 SW 기초 기술 활용	응용 SW 기초 기술 활용
데이터아키텍트	애플리케이션 리팩토링	애플리케이션 리팩토링
	인터페이스 설계	인터페이스 설계
	애플리케이션 요구사항 분석	애플리케이션 요구사항 분석

기능 모델링	기능 모델링
애플리케이션 설계	애플리케이션 설계
정적모델 설계	정적모델 설계
동적모델 설계	동적모델 설계
화면 설계	화면 설계
화면 구현	화면 구현
애플리케이션 테스트 관리	애플리케이션 테스트 관리
애플리케이션 테스트 수행	애플리케이션 테스트 수행
소프트웨어공학 활용	소프트웨어공학 활용
<u>소프트웨어개발 방법론 활용</u>	<u>소프트웨어개발 방법론 활용</u>

차 례

학습모듈의 개요	1
학습 1. UI 요구사항 확인하기	
1-1. UI 요구사항 확인	3
1-2. UI 프로토타입 제작, 검토	27
• 교수 · 학습 방법	42
• 평가	43
학습 2. UI 설계하기	
2-1. UI 흐름설계	45
2-2. UI 상세설계	57
• 교수 · 학습 방법	73
• 평가	74
참고 자료	76
활용 서식	77

화면 설계 학습모듈의 개요

학습모듈의 목표

요구사항 분석 단계에서 파악된 화면에 대한 요구사항들을 확인하고, 소프트웨어 아키텍처 단계에서 정의된 구현 지침 및 UI/UX 엔지니어가 제시한 UI 표준과 지침에 따라 화면을 설계할 수 있다.

선수학습

UML(Unified Modeling Language), HTML, 요구사항 확인(2001020201_16v3)

학습모듈의 내용체계

학습	학습 내용	NCS 능력단위 요소	
		코드번호	요소 명칭
1. UI 요구사항 확인하기	1-1. UI 요구사항 확인 1-2. UI 프로토타입 제작, 검토	2001020224_16v4.1	UI 요구사항 확인하기
2. UI 설계하기	2-1. UI 흐름설계 2-2. UI 상세설계	2001020224_16v4.2	UI 설계하기

핵심 용어

UI 요구사항, 사용 사례, UI 프로토타입, HTML, UI 설계서, UI 설계

1-1. UI 요구사항 확인

학습 목표

- 응용 소프트웨어 개발을 위한 UI 표준 및 지침에 의거하여, 개발하고자 하는 응용 소프트웨어에 적용될 UI 요구사항을 확인할 수 있다.

필요 지식

/

① 소프트웨어 아키텍처 개념

1. 소프트웨어 아키텍처란

소프트웨어 아키텍처는 개발하고자 하는 소프트웨어의 사전 작업을 통하여 소프트웨어 개발을 쉽게 하도록 기본 틀을 만드는 것으로, 복잡한 개발을 체계적으로 접근하기 위한 밑그림이라 할 수 있다. 학술적인 정의로는 권도형(2004)에 따르면 소프트웨어를 구성하는 컴포넌트들의 상호 작용 및 관계, 각각의 특성을 기반으로 컴포넌트들이 상호 유기적으로 결합하는 소프트웨어의 진화를 위한 여러 가지 원칙들의 집합이라고 할 수 있다.

2. 소프트웨어 아키텍처의 활용

소프트웨어 아키텍처의 중요성과 활용 방법에 대해 살펴보면, 비교적 간단한 소프트웨어를 개발할 때에는 완성해야 하는 목적과 기능을 중점으로 설계하여도 품질에는 큰 문제가 없다. 그렇지만 소프트웨어의 기능이 복잡, 다양해짐에 따라, 그 기능을 목적에 알맞게 정의하여 분류하여야 하는 명제를 안게 되었다. 분류된 기능이 세분화되면 상호 간에 유기적으로 통합하는 과정이 매우 어려워진다. 그러므로 완전한 소프트웨어를 개발하기 위하여 각각의 기능적 특성을 사전에 파악하여 요구 분석 단계부터 설계 단계까지 분류된 기능과 함께 종합적인 시각으로 판단하는 것이 매우 필요해진다.

이런 이유로 개발하고자 하는 소프트웨어 시스템을 다양한 시각에서 모형화하고 문제의 특성과 본질을 파악하고 필요에 따라 활용할 방안이 요구되었다. 이에 대한 방안으로 필요한 것이 아키텍처이다.

② UI(User Interface)의 활용

UI 흐름 이해, 상세 내용 파악, 가이드라인 비교, 활용에 대해 알아본다.

1. UI 흐름 이해

UI 각각의 상세 내역을 확인하기 전에 먼저 큰 흐름을 살펴보아야 한다. 목적과 그에 맞는 용도, 개발 배경 등 가장 기본이 되는 사항을 확인하여야 하며, 서로 다른 부서 또는 조직 간의 관계와 역할에 대해 명확하게 이해하고 있어야 한다.

2 UI 상세

UI를 쉽게 풀이하면 사용자와 컴퓨터 상호 간의 소통을 원활히 하게 도와주는 연계 작업을 뜻한다. 1990년대부터 시작하였으며 초창기에는 사용자와 컴퓨터의 단순한 상호 작용에 국한된 연구에서 출발하였다. 이후 업무가 복잡해지고 다양해지면서 단순한 방법으로는 많은 문제점이 발생함에 따라 오류를 줄이기 위한 방법으로 변화되었다. 현재에는 작업 수행 내역을 구체적으로 작성하는 기능 위주에서 단순한 기능 전달이 아닌 정보의 내용과 그 안에 포함된 뜻을 전달하는 표현 방법으로 변화하였다.

(1) UI의 세 가지 분야

- (가) 정보 제공과 기능 전달을 위한 물리적 제어 분야
- (나) 콘텐츠의 상세적 표현과 전체적 구성에 관한 분야
- (다) 사용자의 편의성에 맞춰 쉽고 간편하게 사용 가능하게 하는 기능적 분야

(2) UI의 설계 원칙

- (가) 직관성 : 누구나 쉽게 이해하고 사용할 수 있어야 한다.
- (나) 유효성 : 사용자의 목적을 정확하게 달성하여야 한다.
- (다) 학습성 : 누구나 쉽게 배우고 익힐 수 있어야 한다.
- (라) 유연성 : 사용자의 요구사항을 최대한 수용하며, 오류를 최소화하여야 한다.

(3) UI의 설계 지침

- (가) 사용자 중심 : 사용자가 이해하기 편하고 쉽게 사용할 수 있는 환경을 제공하며 실사용자에 대한 이해가 바탕이 되어야 한다.
- (나) 일관성 : 버튼이나 조작 방법을 사용자가 기억하기 쉽고 빠른 습득이 가능하게 설계하여야 한다.
- (다) 단순성 : 조작 방법은 가장 간단하게 작동이 가능하도록 하여 인지적 부담을 감소시켜야 한다.
- (라) 결과 예측 가능 : 작동시킬 기능만 보고도 결과 예측이 가능하여야 한다.
- (마) 가시성 : 주요 기능을 메인 화면에 노출하여 조작이 쉽도록 하여야 한다.

- (바) 표준화 : 디자인을 표준화하여 기능 구조의 선행 학습 이후 쉽게 사용할 수 있어야 한다.
- (사) 접근성 : 사용자의 직무, 연령, 성별 등 다양한 계층을 수용하여야 한다.
- (아) 명확성 : 사용자가 개념적으로 쉽게 인지하여야 한다.
- (자) 오류 발생 해결 : 사용자가 오류에 대한 상황을 정확히 인지할 수 있어야 한다.
- (4) UI가 필요한 이유

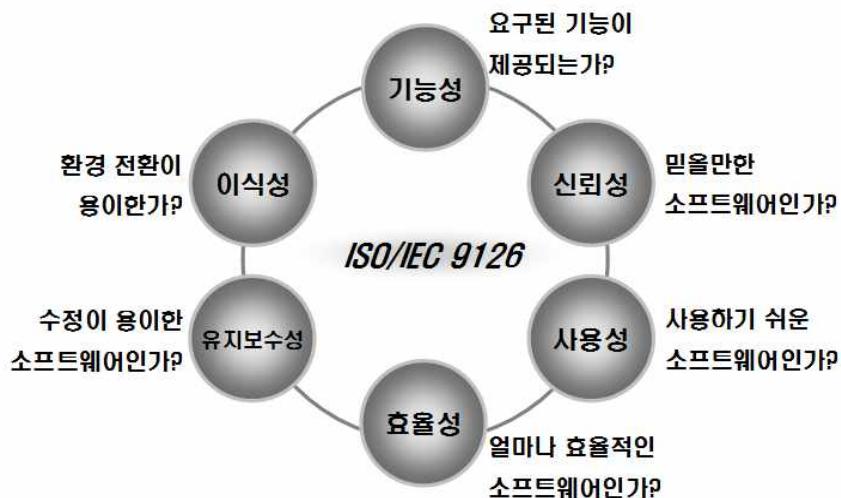
- (가) 구현하고자 하는 결과의 오류를 최소화하고 적은 노력으로 구현하는 결과를 얻을 수 있다.
- (나) 막연한 작업 기능에 대해 구체적인 방법을 제시하여 준다.
- (다) 사용자의 편의성을 높임으로써 작업 시간 단축과 업무에 대한 이해도를 높여 준다.
- (라) 정보 제공자와 공급자의 원활하고 쉬운 매개 역할을 수행한다.

③ UI 요구사항 정의

김영훈(2012)에 의하면 기술의 발전과 기업 환경의 변화에 맞춰 기업의 업무용 소프트웨어도 기존에는 단순하고 체계적이면서 구조화된 기능 중심 관점에서 그 기능을 중심으로 획일화된 UI를 공급하는 소프트웨어 방법을 취했다면, 이제는 시스템을 주로 이용하는 사용자 요구와 필요 기능을 중심으로 UI를 공급함으로써 실체적이고 효과적인 사용자 요구 기능에 알맞은 서비스를 제공하는 형태로 진화하고 있다.

1. 품질 요구사항

소프트웨어 아키텍처 품질 특성 도출은 아키텍처 방법론에 정의된 항목을 중심으로 작성한다.



[그림 1-1] 소프트웨어 아키텍처 품질 요구사항

2. 품질 요구사항 특성

<표 1-1> ISO/IEC 9126 품질 특성

품질 요구사항	상세 품질 요구사항
기능성(Functionality)	적절성, 정밀성, 상호 운용성, 보안성, 호환성
신뢰성(Reliability)	성숙성, 고장 허용성, 회복성
사용성(Usability)	이해성, 학습성, 운영성
효율성(Efficiency)	시간 효율성, 자원 활용성
유지 보수성(Maintainability)	분석성, 변경성, 안정성, 시험성
이식성(Portability)	적용성, 설치성, 대체성

(1) 기능성(Functionality)

실제 수행 결과와 품질 요구사항과의 차이를 분석하고, 실제 사용 시 정확하지 않은 결과가 발생할 확률 등과 관련하여 시스템의 동작을 관찰하기 위한 품질 기준이다.

<표 1-2> 품질 요구사항의 기능성

상세 품질 요구사항	설명
적절성(Suitability)	소프트웨어 제품이 주어진 작업과 사용자의 목표에 필요 적절한 기능들을 제공해 줄 수 있는 소프트웨어의 능력
정밀성(Accuracy)	소프트웨어 제품이 요구되는 정확도로 올바른 결과를 산출할 수 있는 능력
상호 운용성 (Interoperability)	소프트웨어 제품이 특정 시스템과 상호 작용하여 운영될 수 있는 능력
보안성(Security)	프로그램과 데이터에 대해, 비인가된 접근을 차단하고, 우연 또는 고의적인 접근을 인지하여 대처할 수 있는 능력
호환성(Compliance)	소프트웨어 제품이 비슷한 환경에서 연관된 표준, 관례 및 규정을 준수하는 능력

(2) 신뢰성(Reliability)

시스템이 일정한 시간 또는 작동되는 시간 동안 의도하는 기능을 수행함을 보증한다.

<표 1-3> 품질 요구사항의 신뢰성

상세 품질 요구사항	설명
성숙성(Maturity)	소프트웨어 결함으로 인한 고장을 회피할 수 있는 소프트웨어의 능력
고장 허용성 (Fault tolerance)	소프트웨어 결함이나 인터페이스 결여 시에도 특정 수준 이상의 성능을 유지할 수 있는 능력
회복성(Recoverability)	소프트웨어 고장과 그에 대한 시간과 노력이 요구되는 경우 영향 받은 데이터를 복구하고 성능의 수준을 다시 확보할 수 있는 능력

(3) 사용성(Usability)

사용자와 컴퓨터 사이에 발생하는 어떠한 행위를 정확하고 쉽게 인지 가능함을 의미 한다.

<표 1-4> 품질 요구사항의 사용성

상세 품질 요구사항	설명
이해성 (Understandability)	소프트웨어의 논리적인 개념과 적용 가능성(응용 가능성)을 분간하는 데 필요한 사용자의 노력 정도에 따른 소프트웨어 특성
학습성(Learnability)	소프트웨어 애플리케이션 학습에 필요한 사용자의 노력 정도에 따른 특성
운용성(Operability)	소프트웨어의 운용과 운용 통제에 필요한 사용자의 노력 정도에 따른 특성

(4) 효율성(Efficiency)

할당된 시간에 한정된 자원으로 얼마나 빨리 처리하는가를 의미한다.

<표 1-5> 품질 요구사항의 효율성

상세 품질 요구사항	설명
시간 효율성 (Time Behaviour)	소프트웨어의 기능을 수행하는 데 있어 반응 시간, 처리 시간 및 처리율에 따른 소프트웨어 특성
자원 효율성 (Resource Behaviour)	소프트웨어의 기능을 수행하는 데 있어 사용되는 자원의 양과 그 자체 시간에 따른 특성

(5) 유지 보수성(Maintainability)

요구사항을 개선하고 확장하는 데 있어 얼마나 용이한가를 의미한다.

<표 1-6> 품질 요구사항의 유지 보수성

상세 품질 요구사항	설명
분석성 (Analyzability)	소프트웨어 고장의 원인이나 결손 진단 또는 수정이 요구되는 부분의 확인에 필요한 노력 정도에 따른 소프트웨어 특성
변경성 (Changeability)	결함 제거 또는 환경 변화에 따른 수정에 필요한 노력 정도에 따른 특성
안정성(Stability)	소프트웨어의 변경으로 발생하는 예상치 못한 영향에 의한 위험 요소에 따른 특성
시험성(Testability)	소프트웨어가 변경되어 검증에 필요한 노력의 정도에 따른 특성

(6) 이식성(Portability)

다른 플랫폼(운영 체제)에서도 많은 추가 작업 없이 얼마나 쉽게 적용이 가능한가를 의미한다.

<표 1-7> 품질 요구사항의 이식성

상세 품질 요구사항	설명
적용성 (Adaptability)	고려된 소프트웨어의 목적을 위해 제공된 수단이나 다른 조치 없이 특정 환경으로 전환되는 능력에 따른 소프트웨어 특성
설치성 (Installability)	특정 환경에 소프트웨어를 설치하는 데 필요한 노력의 정도에 따른 특성
대체성 (Replaceability)	특정 운용 환경 하에서 동일한 목적 달성을 위해 다른 소프트웨어를 대신 사용할 수 있는 능력

④ UI 요구사항 확인

1. UI 요구사항 확인

프로젝트의 요구사항은 크게 시스템이 무엇을 하여야 하는지를 설명하는 기능적 요구사항 (Functional Requirements)과 개발 과정에서 지켜져야 할 제약조건들을 설명하는 비기능적 요구사항(Nonfunctional Requirements)으로 나눠진다.

(1) 기능적 요구사항

- (가) 시스템의 입력으로 무엇이 포함되어야 하나?
- (나) 시스템의 출력으로 무엇이 포함되어야 하나?
- (다) 시스템이 어떤 데이터를 저장해야 하나?
- (라) 시스템이 어떤 연산을 수행해야 하나?
- (마) 기타 요구사항(예: 동기화 등)

(2) 비기능적 요구사항

- (가) 사용성, 효율성, 신뢰성, 유지 보수성, 재사용성 등 품질에 관한 요구사항
- (나) 플랫폼, 사용 기술 등 시스템 환경에 관한 요구사항
- (다) 비용, 일정 등 프로젝트 계획에 관한 요구사항

수행 내용 / UI 요구사항 확인하기

재료 · 자료

- 기능 구조 설계 사례, 스타일 가이드 설계 사례

기기(장비 · 공구)

- 종이와 펜
- 컴퓨터
- 파워포인트(PowerPoint) 또는 한글(HWP)
- 메모장 등의 HTML 편집 도구

안전 · 유의 사항

- UI 아키텍처와 UI의 흐름에 대한 정확한 개념을 숙지한다.
- UI 요구사항들이 설계 원칙을 준수하여 모두 반영되도록 고려한다.
- UI에서 품질 요구사항의 중요성을 인식하고 설계에 앞서 충분한 사전 준비를 실시한다.
- UI 요구사항에서 기능적 요구사항과 비기능적 요구사항을 숙지한다.

수행 순서

① 응용 소프트웨어 개발을 위한 UI 표준 및 지침을 확인한다.

UI 표준에 맞춰 기본 화면 구조, 액션 인터페이스(Action Interface), 화면 배치(Layout), 일반 UI 네비게이션 표준 방식, 화면 구성 요소, 업무별 화면 샘플, 메시지 창, 버튼 목록, 배치(버튼, 필드), 데이터 포맷(Data Format)을 확인한다.

1. UI 스타일 가이드를 정의한다.

(1) 구동 환경을 정의한다.

(가) 컴퓨터 OS를 확인한다.

기업이 운영하고 있는 업무와 보안성이 높은 운영 체제(OS)를 확인한다.

(나) 웹 브라우저를 확인한다.

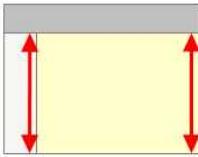
웹 브라우저는 익스플로러, 크롬, x-internet 등 기업 환경에 가장 적합한 것으로 확정한다.

(다) 모니터 해상도를 확인한다.

모니터 해상도는 1280px*1024px을 표준으로 한다. 컴퓨터 작업 표시줄 및 브라우저의 기본 환경(Settings)을 기준으로 브라우저 스크롤이 생기지 않는 안전 지역(Safety Area)은 1280px*2014px로 한다.

(라) 프레임 세트를 확인한다.

업무 처리가 주목적으로, 속도 및 업무 편의성을 고려하여 각 영역별(Top, Left, Contents 영역) 프레임을 구분해 적용한다.

구분	Case 1. Frame 구분	Case2. One frame	Web Application 경우
Contents 구성	Frame별 Contents 구성 각 Frame의 page에서 메뉴, 배너 구성 등 일괄 적용됨.	전체 페이지에서 각 area별 Contents를 자유롭게 구성할 수 있음. - 각 페이지 별로 메뉴, 배너 구성, Contents 변경 관리에 편리	Page 별 구성 Contents에 구성이 동일한 Pattern - page 별로 특화된 배너 적용하는 case 드럼.
Design	각 Frame 별 이미지 적용 - Frame 별 scroll이 생겨 Frame 내에 Background 이미지 적용	전체 페이지에 적용되는 이미지 가능. Background 이미지 사용시에도 전체 Page 내에서 scroll 생겨 적용 가능함.	업무 처리가 주목적으로 페이지 전체에 이미지 적용하는 경우는 드럼.
속도	변경되는 Frame 만 새로 Loading 됨. (브라우저 렌더링 하는 속도 향상.)	페이지 전체가 새로 Loading 됨. (이전에 봤던 Contents 가 메모리에 올라와 있더라도 렌더링 하는 속도가 걸림.)	메뉴 변경시 페이지 loading 이 빨라야 함.
스크롤			

[그림 1-2] 프레임 세트 적용 예

(2) 레이아웃을 정의한다.

(가) 화면 구조를 정의한다.

기본 배치(Layout)는 크게 Top, Left, Contents 영역의 3개 부분으로 설계하며, 하단 메뉴 구성(Footer Area)은 상황에 맞게 추가하거나 제외해도 된다.

(나) 상단 메뉴 구성(Top Area)을 정의한다.

필수적으로 적용하는 사항이며, 구성 요소로는 시스템 로고(System Logo), 로그인 사용자(Login User), 바로가기 메뉴(Quick Menu), 주 메뉴(Main Navigation)가 있다. 시스템 전체 페이지에 동일하게 적용되며 변경이나 추가가 필요한 경우 주 메뉴 크기만 변경 가능하게 한다.



[그림 1-3] 상단 메뉴 구성

1) ① Logo Area(로고 구역)

화면 왼쪽 상단에 위치하며 회사 로고(Logo) + 시스템 로고(System Logo)가 들어간다. 여백의 사이즈는 일정하게 하며, 페이지별로 크기를 고정하여 웹 사이트 전체에 일관성 있게 구현되도록 한다. 시스템별 특성에 맞는 로고를 제작해 적용한다. 시스템 로고에 링크(Link)를 걸어 메인 화면(Application Main Page)으로 이동 가능하도록 한다.

2) ② Login User(접속자) 정보

화면 우측 상단 첫 번째에 위치하며, 접속자에 대한 정보를 표시(Display)한다.

(예) [홍길동 0000000]님 로그인

3) ③ Quick Menu(바로가기 메뉴)

Logo 우측 상단 두 번째에 위치하며, 홈/Home), 매뉴얼(Manual), 사이트 맵(Site map), 관리자/Admin) 등 화면(Site) 전반에 걸친 메뉴를 우측 정렬로 배치한다.

4) ④ Main Navigation(주 메뉴)

로고 우측 상단 두 번째 줄에 위치하며, 시스템의 주 메뉴를 왼쪽 정렬로 배치 한다. 마우스 오버 시 해당 메뉴의 배경 화면색(Back Ground Color) 혹은 글자 색(Text Color)이 변경되도록 한다. 첫 번째(1 depth) 메뉴 항목의 폭(Width)은 일정하게 하도록 한다.



[그림 1-4] 기본형



[그림 1-5] 프로젝트 명을 표기할 경우

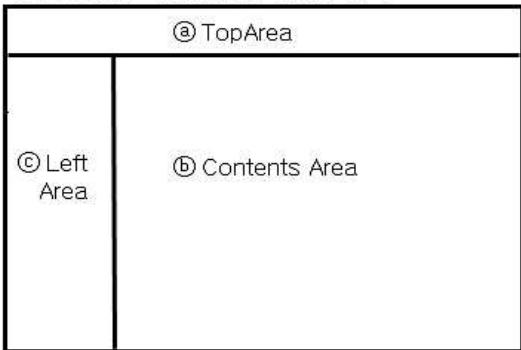


[그림 1-6] Open/close 메뉴와 프로젝트 명 표기할 경우(Project 명은 Left Frame에 존재)

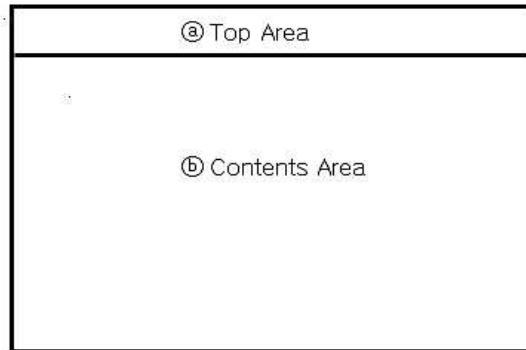
(다) 좌측 메뉴 구성(Left Area)을 정의한다.

선택적으로 적용하는 사항이며, 구성 요소로는 서브 메뉴(Sub Menu), 배너(Banner) 가 있다. 시스템별 서브 페이지에 선택 적용한다.

•Layout 1안: Left Area 적용할 경우



•Layout 2안 : Left Area 적용 안 할 경우



[그림 1-7] Left Area 적용 예시

(라) 내용 구성(Contents Area)을 정의한다.

필수적으로 적용하는 사항이며, 구성 요소로는 메인 이미지(Main Image), 시스템별 구성 콘텐츠가 있다. 시스템의 전체 콘셉트를 나타내는 메인 이미지와 시스템별로 필요한 콘텐츠를 적용하는 곳으로서 유동성 있게 구성이 가능하다.

(마) 하단 메뉴 구성(Footer Area)을 정의한다.

선택적으로 적용하는 사항이며, 구성 요소로는 회사 CI, Copyright가 있다. 회사 상황에 맞춰 적용 및 삭제 가능하다.

(바) 사용 환경에 맞춰 페이지 폭을 정의한다.

브라우저 사이즈에 따라 페이지 폭 크기(Width Size)를 유동적으로 적용하여 화면 활용도를 높이는 것을 기본으로 한다.

<표 1-8> 사용 환경에 따른 페이지 Width Size

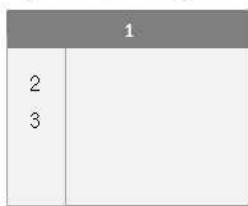
구분	폭 크기(Width Size) 유동적(표준)	폭 크기(Width Size) 고정(일부 예외)
예방	높은 해상도, wide computer에서 브라우저 가로 사이즈를 넓게 사용할 경우 Scroll 생기며 용함으로써 복잡한 프로그램일 경우 텍스트가 고정되어 보임. 우 화면 활용도 높아짐.	
탐지	텍스트 줄이 깨져 보이는 경우 발생	높은 해상도일 경우 화면 활용도가 떨어짐.
대응	Office Net, 토크 EPMS, 건축 영업 정보 시스템	Proqnet, UBMS, TOMS 등(UIGuide 제정 이전에 구축한 애플리케이션임.)

(3) 네비게이션을 정의한다.

(가) 메뉴 네비게이션(Menu navigation)을 정의한다.

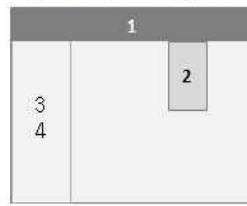
메뉴 네비게이션은 4가지 타입의 응용 프로그램(Application)의 메뉴 구조에 따라 적당한 타입을 선택하여 적용한다.

Type 1. 기본 Navigation



- Top : 1depth
- Left : 2~4 depth

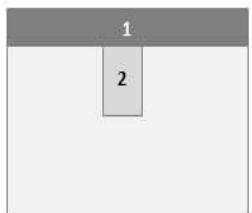
Type 2. 기본 Navigation 의 변형



- Top : 1, 2 depth
- Left : 3, 4 depth

[그림 1-8] 메뉴 네비게이션의 기본 적용 타입

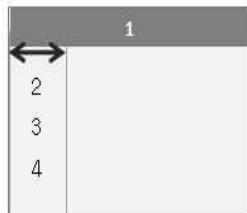
Type 3. 메뉴 구조가 2depth 인 경우 적용



- Top : 1, 2 depth

(세로 Drop down)

Type 4. 메뉴 구조가 복잡할 경우 적용.



- Top : 1depth
- Left : 2~4 depth

(Open, Close)

[그림 1-9] 화면을 넓게 사용할 경우 적용 타입

1) Type 1을 적용한다.

Top, Left 구성을 기본 타입으로 한다.

2) Type 2를 적용한다.

메뉴 구조가 복잡할 경우 상단 메뉴 구성에 2depth를 Drop down으로 적용한다.

3) Type 3, 4를 적용한다.

프로그램 구성 특성상 화면의 폭(Width)을 넓게 쓰고 싶을 경우 좌측 메뉴에 대해서는 유동적 적용을 한다.

(나) 모듈 뷰(Module View)를 정의한다.

효과적인 모듈의 분리를 위해 응집력(Cohesion)은 최대화하고 결합도(Coupling)는 최소화하여야 하며, 모듈 뷰 타입은 구현 단위 모듈 요소이며, 각 모듈은 기능적 책임을 할당한다.

응집력과 결합도의 정의

응집력이란 모듈과 모듈 요소 사이에 발생하는 끌림의 정도이다.

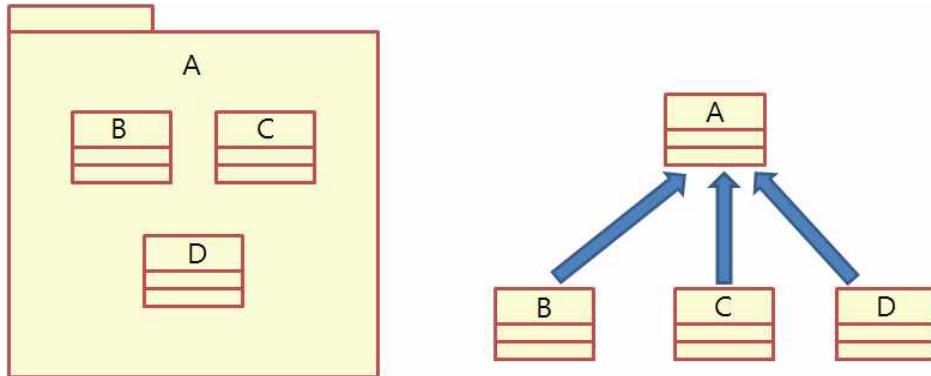
결합도는 모듈이 다른 모듈과 종속성을 형성하는 긴밀함의 정도이다.

분할 스타일, 사용 스타일, 일반화 스타일, 레이어 스타일의 4가지 스타일을 포함하여 모듈 뷰 타입을 정의한다.

1) 분할 타입(Decomposition Type)을 확인한다.

- 분할 모듈 뷰 타입은 아키텍처에 대한 청사진을 그린 후 여러 사람들과 의사 소통에 많이 사용되는 방법이다.
- 변경되거나 추가되는 기능을 아키텍처의 특정 위치에 할당하여 변경이 가능하도록 하여 품질속성의 변경을 허용하는 데 사용될 수 있다.

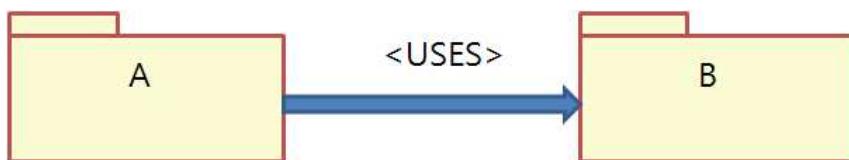
- 모듈을 작게 세분화한 서브 모듈로 쪼개는 기준은 사용하는 목적에 따라 달라진다. 일정한 품질속성을 구현하기 위해 분할을 선택할 수 있다.
- 데이터 숨김(Data-hiding) 원리를 이용하여 시스템에서 변경 가능한 부분을 별도의 특정 모듈로 캡슐화함으로써 변경된 기능의 영향을 최소하여 일정 부분에서만 영향이 가도록 할 수 있다.
- 품질속성을 높이기 위해 분할을 원한다면 높은 성능을 필요로 하는 모듈을 상대적으로 낮은 성능이 필요한 모듈에서 분리하여 각각 다른 전략을 적용한다.



[그림 1-10] 분할 타입

2) 사용 타입(Uses Type)을 확인한다.

- 사용 모듈 뷰 타입은 종속적인(depends on) 관계가 형성된 관계를 기반으로 한다. 사용 관계는 두 개의 모듈 사이의 관계에서 특정 모듈이 정확한 기능을 발휘하기 위해서는 다른 하나의 모듈이 정확하게 구현되어 있어야 함을 전제로 한다.
- A 모듈과 B 모듈은 A가 B를 사용하는 관계이다. 사용 관계는 종속적인 관계의 특이한 형태이며, 어떤 모듈 A의 정확한 정도가 모듈 B의 정확한 구현 여부에 종속된다면 A모듈은 B모듈을 사용한다고 할 수 있다. 사용 관계에서 A 모듈이 B모듈의 호출을 항상 하지는 않는다.



[그림 1-11] 사용 타입

3) 일반화 타입(Generalization Type)을 확인한다.

- 일반화 모듈 뷰 타입은 모듈이 일반적인 관계를 형성하는 경우 가변성(Variation)과 공통성(Commonality)을 표현할 수 있다.
- 공통성을 가지는 부모 모듈(Parent Module)은 가변성을 이루는 자식 모듈(Child Module)을 일반화한 것으로, 부모 모듈이 갖는 특성은 공통성을 포함하

며 자식 모듈은 가변성을 표현하게 된다. 일반화 관계를 사용하여 아키텍처의 확장(Extension) 가능성과 진화(Evolution) 가능성을 이룰 수 있다.

- 자식 모듈의 추가나 변경 또는 삭제를 통하여 아키텍처를 확장할 수 있으며, 부모 모듈이 변경하면 모든 자식 모듈도 다 같이 변경되어 진화를 이룰 수 있게 한다.

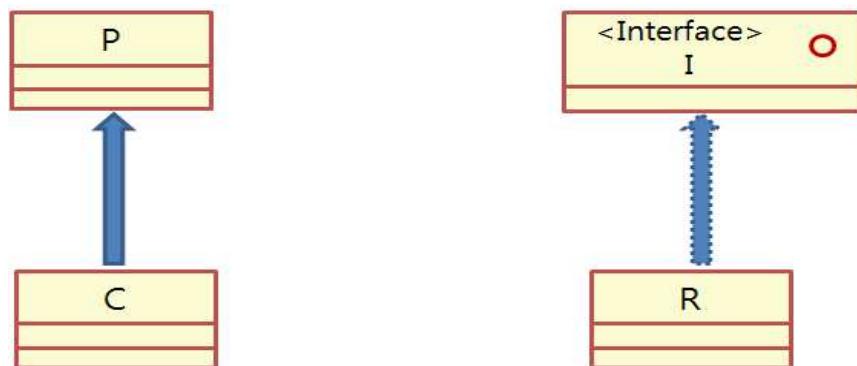
일반화 타입(Generalization Type)의 구현 상속과 인터페이스 상속 차이

-구현 상속(Implementation Inheritance) :

부모 모듈로부터 처리하는 일을 상속받고 그 일을 수행함으로써 빠르고 쉽게 일을 처리한다. UML에서 일반화(Generalization) 관계로 표현한다.

-인터페이스 상속(Interface Inheritance) :

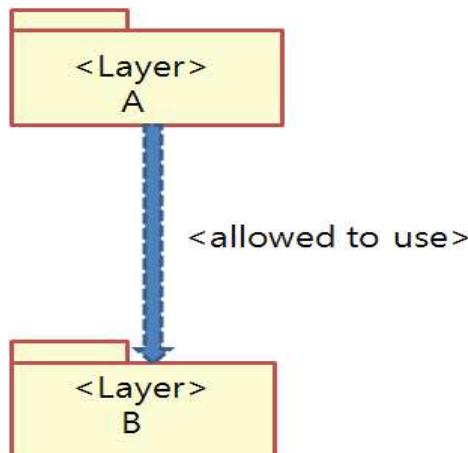
자식 모듈이 부모 모듈이 지정한 일(인터페이스가 정의된 규칙)을 실현하는 것을 의미한다. UML에서 실현(Realization) 관계로 표현한다.



[그림 1-12] 일반화 타입

4) 레이어 타입(Layered Type)을 확인한다.

레이어 타입은 개발 팀에게 작업 할당의 기준을 제시한다. 레이어는 또한 분석 도구로서 활용 가치가 있으며, 시스템의 변경이 필요한 경우 변경 영역을 결정하고 변경된 사항을 개발 팀에 넘겨 설계에 반영하게 한다.



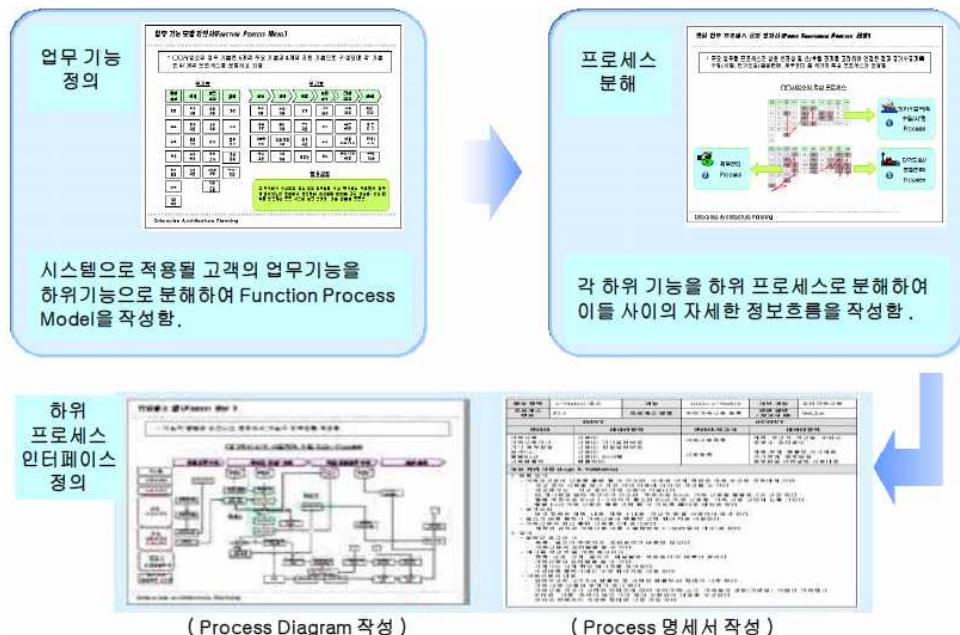
[그림 1-13] 레이어 타입

(4) 기능을 정의한다.

시스템 요구사항에 대한 개념 모델을 논리적 모델(프로세스 모델, UI설계, 논리 데이터 모델, 아키텍처 정의, 인터페이스 설계 측면)로 상세화한다.

(가) 프로세스 모델링(Process Modeling)을 정의한다.

신규 시스템으로 적용할 해당 업무 과정에서 일어나는 모든 활동들을 알기 쉽게 파악하고 정확한 영역 등의 파악을 위해 업무 기능 모델(Function Process Model)을 수립한다.

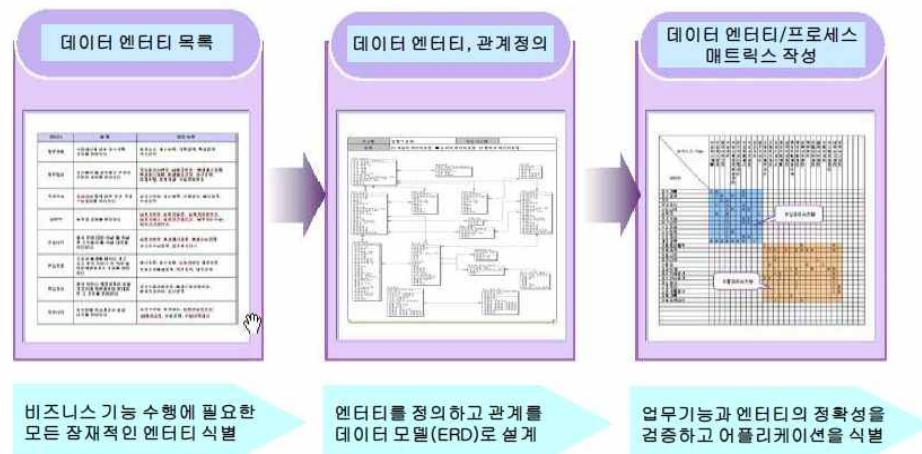


[그림 1-14] 프로세스 모델링 예제

(나) 데이터 모델(Data Model)을 정의한다.

적용할 해당 업무 영역에 필요한 데이터 엔티티를 식별하고 정의하며, 이를 기반으로 데이터 엔티티 및 엔티티 간의 관계를 논리적 데이터 모델로 정의한다.

Data Model 정의



[그림 1-15] Data 모델

(5) 구성 요소를 정의한다.

(가) 그리드를 정의한다.

UI를 구성하는 방법 중 테이블(Table) 형태로 구성한다.

Type A : Fetch 적용이 필요한 Grid화면

A1화면 : LocalFetch, 총 건수 정보/이동버튼

조회된 Data는 첫 row에 Display

Local Fetch 건수 / 총건수

NextFetch Request Icon

Scroll Down & Next Fetch Request

선택	입력학목타이틀	입력학목타이틀	입력학목타이틀	입력학목타이틀	입력학목타이틀
<input type="checkbox"/>	가나다라마바사	가나다	000,000	가나다라마	00
<input checked="" type="checkbox"/>	가나다라마바사	가나다	000,000	가나다라마	00
<input type="checkbox"/>	마자차카타파하	가나다	000,000	가나다라마	00
<input type="checkbox"/>	마자차카타파하	가나다	000,000	가나다라마	00
<input type="checkbox"/>	가나다라마바사	가나다	000,000	가나다라마	00

[그림 1-16] 그리드 적용 예제 1

- 1) Type A는 그리드 데이터 처리(Grid Data Fetch) 방식으로 기존에 조회된 데이터에 새로 조회된 데이터가 추가되는(Append) 방식으로 한다.
- 2) 한 번 처리(1 Fetch)에 대한 양은 업무(Business) 특성에 맞게 20, 30, 50, 100, 150건 등에서 적절한 값을 선택한다(한 번의 처리로 업무의 80% 정도를 수행 할 수 있는 양을 정한다).
- 3) 스크롤 내림(Scroll Down) 시 마지막 조회 데이터가 화면에 표시(Display)될 시 다음 페이지를 자동 조회하여 추가(Append)한다.
- 4) 데이터 디스플레이에는 선택 항목이 없을 시에는 조회된 데이터가 추가(Append) 되면 해당 페이지의 첫 행(row)이 화면의 첫 부분에 보임(Display), 선택 항목이 있을 시에는 현재 페이지가 보이고 데이터만 추가(Append)한다.

Type B : 총 건수만 필요한 Grid 화면

B화면 : 총 건수만 Display

선택	입력할목타이틀	입력할목타이틀	입력할목타이틀	입력할목타이틀	입력할목타이틀
<input checked="" type="checkbox"/>	가나다라마바사	가나다	000,000	가나다라마	00
<input type="checkbox"/>	가나다라마바사	가나다	000,000	가나다라마	00
<input type="checkbox"/>	가나다라마바사	가나다	000,000	가나다라마	00
<input type="checkbox"/>	가나다라마바사	가나다	000,000	가나다라마	00
<input type="checkbox"/>	가나다라마바사	가나다	000,000	가나다라마	00

[그림 1-17] 그리드 적용 예제 2

- 5) Type B는 대상 데이터 건수가 적은 화면에 사용한다. 데이터가 많아도 관리 업무나 해당 화면의 사용 빈도수가 적은 화면, 데이터의 총 건수가 중요한 화면은 이 타입을 적용한다.
- 6) 이 타입은 총 건수 처리에 대한 시스템 부하가 많으므로 가급적 사용하지 않는다.

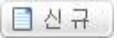
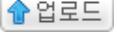
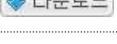
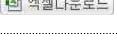
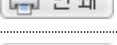
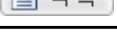
(나) 버튼을 정의한다.

기능 버튼, 검색 버튼, 그리드 관련 버튼, 기타 버튼 4가지로 구분해 제작한다.

1) 기능 버튼을 정의한다.

조회, 저장과 같이 화면마다 공통으로 사용되면서 화면 전체에 영향을 미치는 버튼으로, 화면 상단에 위치한다. 좌측에 아이콘 이미지가, 우측에 버튼 명이 위치하도록 한다.

<표 1-9> 기능 버튼

국문 명	영문 명	설명	버튼	정렬
신규	New	신규 등록할 수 있는 입력 화면으로 이동한다.	 신규	좌측
초기화	Clear	입력, 수정 중이던 항목들의 값을 원 상태로 되돌린다.	 초기화	
수정	Modify	조회된 정보를 수정할 수 있는 상태로 변환한다.	 수정	
저장	Save	입력, 수정한 내용을 DB에 저장한다. 다중 데이터(Multi Data)를 다룰 경우 그리드의 추가, 수정, 삭제된 데이터를 실제로 DB에 반영한다.	 저장	
삭제	Delete	하나(Single)의 데이터를 다룰 경우 조회된 데이터를 삭제하고자 할 경우 해당 버튼을 누르면 실제 DB에서 삭제한다.	 삭제	
업로드	Upload	자료를 업로드한다.	 업로드	
다운로드	Download	자료를 다운로드한다.	 다운로드	
엑셀	Excel	엑셀 파일로 자료를 다운로드한다.	 엑셀다운로드	
인쇄	Print	자료를 인쇄한다.	 인쇄	
목록	List	목록 화면으로 이동한다.	 목록	우측

2) 검색 버튼을 정의한다.

검색 영역 안에 위치하는 버튼으로 색(Color)과 스타일(Style)은 애플리케이션별로 자유롭게 디자인하도록 한다.

3) 그리드(Grid) 관련 버튼을 정의한다.

그리드의 데이터를 처리(Handling)하는 버튼이다.

<표 1-10> 그리드 관련 버튼

국문 명	영문 명	설명	버튼
행추가	Add row	신규 등록할 수 있는 입력 화면으로 이동한다.	
행삭제	Delete row	입력, 수정 중이던 항목들의 값을 원 상태로 되돌린다.	
행복사	Copy row	조회된 정보를 수정할 수 있는 상태로 변환한다.	
저장	Save	해당 그리드의 변경된 데이터를 DB에 저장한다.	

4) 기타 버튼을 정의한다.

업무 관련 버튼, 그리드 버튼을 제외한 모든 버튼을 말하며 도움말 버튼을 제외하고는 모두가 한 화면의 특정 영역 내에서만 사용된다.

<표 1-11> 기타 버튼

종류	설명	버튼
찾기	특정 입력 필드에 대해서 사용자가 입력할 값을 명확히 알지 못할 때 입력 값을 검색할 수 있는 팝업 창을 띄우는 버튼이다.	
달력	날짜 입력 필드에 대해 사용자가 달력을 보고 선택할 수 있도록 팝업 창을 띄우는 버튼이다.	
우편번호	우편번호 찾기 팝업 창을 띄운다.	

2. UI 패턴 모델을 정의한다.

CRUD 방식을 기반으로 하여 데이터의 입력과 출력을 처리하는 화면 플로우(Flow)를 포함하여 오퍼레이션 방식에 대한 표준 절차를 표시하고, UI 패턴 모델(Pattern Model)을 개발 표준 프레임워크(Framework)로 개발하고, 유스케이스(Use Case)를 이용해 패턴별 표준 개발 방법 총 7가지 영역을 정의한다.

(1) 업무 화면 클라이언트(Client)를 정의한다.

어느 형태의 클라이언트를 사용할 것인가는 제안 단계에서 결정된다. 설계자(Architect)는 결정된 클라이언트를 가지고 개발 시에 필요한 공통 요소 식별, 디렉토리 구성, 개발 환경 구축에 관여해야 한다. 클라이언트에 출력되는 UI는 X-Internet으로 대변되는 리치 클라이언트(Rich Client) 도구와 일반 JSP, Html 기반의 신 클라이언트(Thin Client) 방식이 대표적으로 사용된다.

<표 1-12> 클라이언트 화면 비교 예시

구분	Rich Client	Thin Client	비고
브라우저	<p>전용 브라우저를 사용할 수도 있고, 기존 웹 브라우저(IE)에 ActiveX 방식으로 임베디드하는 방식을 사용할 수 있으나, 대부분 전용 브라우저를 사용한다.</p> <p>전체 화면을 JSP 등 Thin Client 방식으로 가져가면서 업무 화면(Body)만을 X-Internet으로 가져가는 경우에는 임베디드 방식을 사용해야 한다.</p> <p>전용 브라우저 혹은 ActiveX는 초기 접속 시 웹 서버로부터 클라이언트 PC에 자동 설치된다.</p>	<p>기존의 웹 브라우저만으로 충분하다.</p>	
프로그래밍	<p>자체 스크립트 엔진을 가지고 있으며, 따라서 자체적인 Script Language를 제공한다. 솔루션에 따라 표준 ECMA Script를 준수하기도 한다.</p> <p>Dataset이라는 Table 형태의 반복적인 구조체를 지원한다.</p>	<p>웹 브라우저의 종류 및 버전에 따라 지원하는 Javascript가 차이가 나므로 이 점을 사전에 확인해야 한다. 구조체를 지원하지 않으므로 입력은 파라미터 방식으로 처리하고, 출력 시는 JSTL 혹은 scriptlet 방식으로 Rendering한다.</p>	<p>Dataset의 존재로 인해 프로그래밍 시 데이터 표현은 Rich Client가 훨씬 풍부하게 할 수 있다.</p>
서버 통신	<p>XML 기반의 http 통신을 사용한다.</p> <p>http body에 자체 정의한 XML 데이터를 Request/Response에 사용한다.</p>	<p>전형적인 form submit 방식 혹은 GET 방식을 사용한다.</p> <p>부기적으로 AJAX를 사용하거나 이를 더욱 확장한 JSON-RPC를 사용할 수 있다.</p>	<p>필요한 데이터만 주고받을 수 있다.</p>
UI Control	<p>그리드, 트리 등 매우 다양하고 강력한 기능의 컨트롤을 제공하며, 자체 엔진에 의해 렌더링 및 작동한다. 제품에 따라 CSS 등 웹 표준 스타일을 지원하기도 한다. (TrustForm) HTML 컨트롤과는 호환되지 않는다.</p>	<p>HTML에서 제공하는 표준 UI 태그를 사용한다. 이 또한 브라우저의 종류 및 버전에 따라 차이가 날 수 있다.</p> <p>CSS/DOM 적용이 가능하다.</p>	

(2) 서버 컨트롤러(Controller)를 정의한다.

프레임워크를 도입한다면 해당 프레임워크가 제공하는 방식을 채택한다. 두 가지 방식을 다 지원하면 기준을 정하여 상황에 맞게 적용한다. 또한 별도의 클라이언트 제품을 도입하는 경우 서버 컨트롤러와의 연동 방식을 결정한다.

(3) 서버 메시지 및 예외(Exception) 처리를 정의한다.

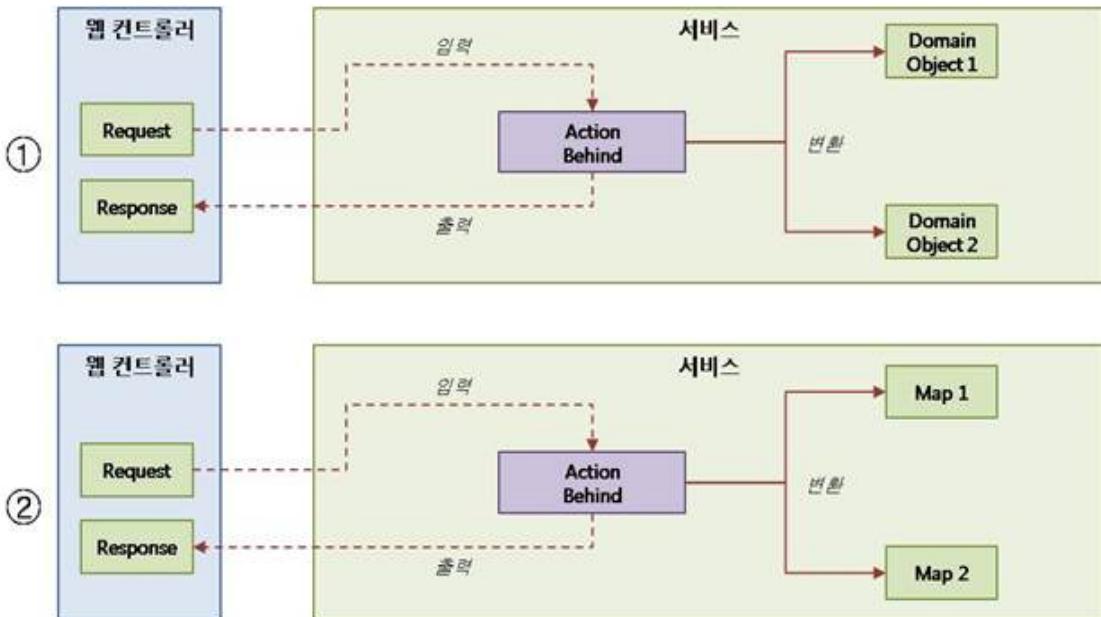
서버의 메시지 및 예외 처리를 클라이언트 UI에 전달하는 방식을 결정한다.

<표 1-13> 예외 처리

유형	메시지 유형 설명
S(System)	시스템 오류로 인해 발생하는 메시지이다. 런타임 예외를 throw할 때 사용되며, 모든 트랜잭션은 자동으로 복원(rollback)한다.
E(Error)	업무 처리 로직의 일환으로 애플리케이션 예외(Application Exception)를 throw할 때 사용한다. 이때 모든 트랜잭션은 자동으로 복원한다. (ex: E0001: 금액이 마이너스 값입니다.)
I(Information)	정상적인 업무 처리 결과나 관련 정보에 대한 확인 메시지를 사용자에게 알려주고자 할 때 사용한다. 이때 모든 트랜잭션은 실행(commit)한다. (ex : 고객 정보가 성공적으로 조회되었습니다.)

(4) 클라이언트(Client) – 서버 간 데이터 변환을 정의한다.

어떤 방식의 오브젝트(Object)를 사용할 것인지를 먼저 결정하고, 이에 따라 클라이언트와 서버 간의 데이터 형태 변환을 어떻게 처리할 것인지 방안을 마련한다.



[그림 1-18] Client-서버 간 데이터 변환

유형(Type) 1은 별도의 명백한(Plain) 자바 빙(Java Bean, Domain Object)을 DTO로 구현하여 Request 객체 내에 있는 해당 데이터를 Plain VO로 전환하는 방식이다. 예를

들면, 고객(Customer), 주문(Order), 상품(Product) 등을 표현하는 클래스들을 일일이 개발하여 활용한다.

유형(Type) 2는 별도의 구현 없이 자바 맵(Java Map) 기반의 공통 데이터 클래스를 활용하는 방식이다.

(5) EP 연계를 정의한다.

EP - SSO - Client 간 연계 방안을 URL 연계 시와 포틀릿(Portlet) 연계 시를 고려하여 마련한다.

(6) 보고서를 정의한다.

클라이언트와 리포트(Report) 솔루션 간의 연계 방식을 결정한다.

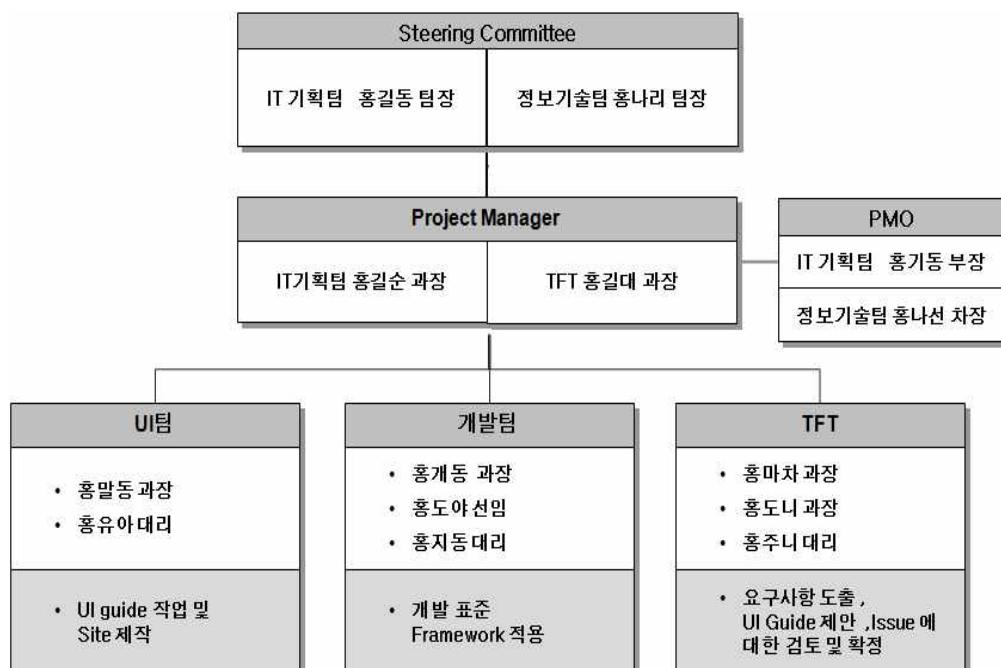
(7) 신 클라이언트(Thin Client)에 외부 컴포넌트 연계를 정의한다.

외부 UI 컴포넌트를 도입할 시 서버와의 연계 방식을 결정한다.

3. UI표준 수립을 위한 조직을 구성한다.

(1) 조직 구성 및 역할을 정의한다.

효과적인 프로젝트 추진을 위하여 UI 팀 및 표준 개발 팀을 주축으로 한 추진 조직 구성을 확정한다.



[그림 1-19] UI 표준 수립을 위한 조직 구성의 예

(2) 커뮤니케이션 방안을 수립한다.

프로젝트가 원활히 수행되도록 정식 보고 및 정기적인 회의뿐 아니라 이슈 회의 등 다양한 방식의 커뮤니케이션을 수행한다.

<표 1-14> 커뮤니케이션 방안

커뮤니케이션 형태	내 용	시 기
착수 보고	프로젝트 범위, 일정, 조직 등	KICK OFF 시
정식 보고	1차 보고	UI Guide 기획 보고 UI Guide 기획 완료 시
	2차 보고	UI Guide 제작 및 Web Site 기획 보고 UI Guide Style Guide, Pattern Model 제작 완료 시
완료 보고	프로젝트 결과물 및 향후 계획 보고	프로젝트 종료 시
정기 회의	주간/월간 회의	프로젝트 진행 사항 파악 및 공유 이슈 사항 검토 및 해결 매주 마지막 주간 회의는 월간 회의로 대체
비정기 검토	사용자 인터뷰	요구사항 도출 및 협의 수시/필요 시점
	이슈 회의	이슈 파악에 대한 해결책 마련 및 시행 수시/주요 현안 발생 시

4. UI표준을 위한 환경을 분석한다.

(1) 사용자 트렌드를 분석한다.

- (가) 기존 UI와 현재 UI 트렌드를 숙지한다.
- (나) 현재 UI의 단점을 상세히 나열한다.
- (다) 사용자가 필요로 하는 핵심 요구사항을 파악한다.
- (라) 사용자가 쉽게 이해 가능한 기능을 위주로 기술 영역을 정의한다.

(2) 기능 및 설계를 분석한다.

(가) 기능 조작성에 대해 분석한다.

- 1) 사용자 편의를 위한 조작에 대해 확인한다.
- 2) 스크롤바는 지원 가능한지 확인한다.
- 3) 마우스 조작 및 업무 처리 시 동선에 대해 확인한다.

(나) 오류 방지에 대해 분석한다.

- 1) 사용자 조작 시 오류에 대해 예상 가능한지 확인한다.
- 2) 사용자 의도와 관계 없는 페이지 이동이 있는지 확인한다.
- 3) 기능 버튼은 사용자가 명확한 구분이 가능한지 확인한다.
- 4) 기능 버튼 명이 사용자 조작과 일치하는지 확인한다.

(다) 최소한의 조작으로 업무 처리 가능한 형태를 확인한다.

- 1) 작업 흐름에 가장 적합한 레이아웃을 확인한다.

- 2) 기능 특성에 맞는 UI를 확인한다.
 - 3) 조작 단계를 최소화하고 동선은 단순한지 확인한다.
- (라) UI의 정보 전달력을 확인한다.
- 1) 중요 정보에 대해 사용자가 인지하기 쉽도록 전달 가능한지 확인한다.
 - 2) 정보 제공 방식이 일관적이며 사용자가 쉽게 이해 가능한지 확인한다.
 - 3) 상호 연관성이 높은 정보가 쉽게 인지 가능한지 확인한다.
 - 4) 오류 발생 시 조치를 위한 접근이 쉬운지 확인한다.
 - 5) 사용자 정보 제공이 간결하고 명확한지 확인한다.

② 개발하고자 하는 응용 소프트웨어에 적용될 UI 요구사항을 확인한다.

다음은 정보통신산업진흥원 부설 SW공학센터의 “소프트웨어 개발 UI/UX 참조모델 가이드”를 적용한 처리 절차이다.

1. 비즈니스 요구사항을 확인한다.

(1) 목표를 정의한다.

- (가) 사용자를 대상으로 심층 인터뷰를 통해 의견을 수렴하여 비즈니스 요구사항을 정의한다.
- (나) 인터뷰를 통해 사업적, 기술적인 요소를 깊게 이해하여 목표를 명확히 한다.
- (다) 사업적, 기술적 목표가 확정되면 UI/UX 디자인 프로세스를 정의한다.
- (라) 인터뷰 시 주의할 사항은 사용자 리서치 시작 전에 선행되어야 한다는 점이다
(인터뷰 후 취득한 결과를 바탕으로 보다 효율적인 리서치를 준비할 수 있다).

(2) 활동 사항을 정의한다.

- (가) 초기 비전과 기대 : 사용자, 고객, 회사의 비전을 일치시키는 작업을 진행한다.
- (나) 비용과 일정 확인 : UX 리서치와 UI/GUI 디자인에 필요한 예산과 일정을 결정한다(리서치의 규모, 디자인 목표 등이 결정된다).
- (다) 기술적 제약과 가능성 : 현재 기술의 발전 가능성 파악, UI/UX 디자인 미래와 나아갈 방향을 제시한다.
- (라) 사업 전략과 사업 목표, 프로세스의 책임자 선정, 회의 일정 및 계획 작성, 우선순위의 설정, 개별적인 단위 업무를 구분한다.
- (마) 경영진의 프로젝트의 이해도 : 인터뷰한 내용을 기반으로 경영진 내의 서로 다른 UI/UX 개발 프로젝트의 이해를 돋고 협의하도록 한다.

(3) 인터뷰를 진행한다.

- (가) 인터뷰는 가능하면 한꺼번에 여러 명을 하지 않고 개별 진행한다.
- (나) 다수의 목소리에 집중하여 개인의 중요한 목소리를 놓칠 위험을 염두해야 한다.
- (다) 각 인터뷰는 한 시간을 넘어서는 안 된다.

2. 요구사항을 작성한다.

여러 경로를 통해 수집, 작성된 요구사항을 검토하고, 목적을 기준으로 데이터 요구, 기능 요구, 제품 품질속성, 제약사항으로 요구사항을 작성한다. 다음과 같이 UI 요구사항들을 바탕으로 UI의 전체적인 구조를 파악 및 검토하고, 내부에 구성될 여러 가지 요소들의 종류와 각각의 표현 방식 등을 검토한다.

(1) 요구사항 요소를 확인한다.

- (가) 데이터 요구 : 사용자가 요구하는 모델과 객체들의 주요한 특성에 기반한 데이터 객체들을 정리한다. 인터페이스에 영향을 주므로 초기에 확인해야 한다.
ex) 이메일의 메시지 속성에는 제목, 발신일, 발신인, 참조인, 답변
- (나) 기능 요구 : 사용자의 목적 달성을 위해 무엇을 실행해야 하는지를 동사형으로 설명한다. 기능 요구 리스트로 정리하며, 최대한 철저하게 해야 한다.
ex) 페르소나(Persona)는 이메일의 메시지 내용을 읽거나 삭제하며, 일정한 양식으로 다른 메시지와 함께 보관한다.
- (다) 제품, 서비스의 품질 : 데이터 및 기능 요구 외에 중요하게 고려할 몇 개의 속성은 제품 품질이 있으며, 여기에는 감성적인 품질도 고려한다.
ex) 시스템이 얼마나 빨리 파일을 처리하는지 여부와 같이 실용적이며 정량화 가능한 요구사항들을 확인한다.
- (라) 제약사항 : 제품 출시의 데드라인, 개발 및 제작에 드는 비용, 시스템 준수에 필요한 규제가 포함되며, 사전에 제약사항의 변경 여부(변경 가능, 변경 불가)를 확인한다.

(2) 정황 시나리오를 작성한다.

- (가) 요구사항 정의의 가장 기초적인 시나리오를 뜻하는 것으로, 높은 수준과 낙관적인 상황에서의 이상적인 시스템 작동에 초점을 맞춘다. 개발하는 서비스의 모습을 상상하는 단계이며, 사용자 관점에서 시나리오를 작성한다.
- (나) 사용자가 주로 사용하는 기능 위주로 작성하여야 하며, 같이 동작하는 기능들은 하나의 시나리오에 통합한다.
- (다) 정리된 리스트를 기반으로 사용자의 관점에서 서술한다. 육하원칙에 따르고 간결하고 명확하게 작성하여 정확하게 전달한다.
- (라) 작성된 시나리오는 외부 전문가 또는 경험이 풍부한 사람에게 검토를 의뢰한다.

수행 tip

- 웹 사이트는 4C(Contents, Commerce, Community, Communication)의 다양한 목적을 가진다.
- 웹 사이트의 경우는 목적에 맞는 사용자의 접근성 및 VI(Visual Identity)를 반영한 디자인 요소를 중요시한다.
- 웹 응용은 업무 처리를 주목적으로 가진다. 또한 효율성을 중시하며, 기능 및 Task 중심이다.
- UI설계를 위해 소프트웨어 아키텍처를 반드시 숙지하고 있어야 한다.
- UI 요구사항 작성 시에는 여러 사람의 인터뷰를 통해 다양한 의견을 수렴하여 작성하는 것이 중요하다.
- UI 요구사항 작성 관점은 반드시 실사용자 중심이어야 하며, 설계 원칙을 최대한 반영하되 사용자 의견과 충돌 발생 시 사용자 설득이 필요하다.
- 사용자가 설계 원칙을 무시하고 요구하는 경우가 종종 발생하므로 설계자는 사용자 설득을 위해 실제 개발에 따른 위험성을 바탕으로 사용자를 설득해야 한다.

1-2. UI 프로토타입 제작, 검토

학습 목표

- 응용 소프트웨어 개발을 위한 UI 표준 및 지침에 의거하여, UI 요구사항을 반영한 프로토타입을 제작할 수 있다.
- 작성한 프로토타입을 활용하여 UI/UX 엔지니어와 향후 적용할 UI의 적정성에 대해 검토할 수 있다.

필요 지식 /

① UI 프로토타입 이해

1. 프로토타입(Prototype)의 뜻

프로토타입은 원래의 온전한 형태, 전형적인 예, 기초적인 표준이다. 시제품 전의 제품 원형으로 개발 검증과 양산 검증의 과정을 거쳐 시제품이 완성된다.

프로토타입은 “새로운 컴퓨터 시스템이나 소프트웨어의 설계 또는 성능, 구현 가능성, 운용 가능성을 평가하거나 요구 사항을 좀 더 잘 이해하고 결정하기 위하여 전체적인 기능을 간략한 형태로 구현한 시제품” (한국어사전)이다. 프로토타입은 사용자의 요구사항이 모두 정확하게 반영될 때까지 계속하여 개선, 보완된다. 실제 수많은 애플리케이션들이 프로토타입의 지속적인 확장, 보강을 통해 최종 설계가 완성된다.

2. 어원에 따른 접근

“프로토타입(prototype)의 사전적 의미는 대량 생산에 앞서 미리 제작해보는 원형 또는 시제품으로, 제작물의 모형이라 할 수 있다. 소프트웨어 개발에서는 정식 절차에 따라 완전한 소프트웨어를 만들기 전에 사용자의 요구를 받아 일단 모형을 만들고 이 모형을 사용자와 의사소통하는 도구로 활용한다.” (김치수, 2015). 이는 원초적이라는 뜻의 ‘프로토타이포스’의 중간 음에서 온 것으로, 더 들어가면 “최초의”라는 뜻의 ‘프로토스’와 “인상”이라는 뜻의 ‘타이포스’에서 비롯된 것이다. UI에 있어 프로토타이핑이란 추후 구현될 시스템의 골격으로서, 사전에 시스템의 일부분 또는 시스템의 기초 모형이 될 것을 수행하는 과정이다.

3. 프로토타입의 의의

프로토타이핑은 설계의 결과물이 아닌 과정으로 요구사항을 반영하고, 1차 검토 후 수정하고, 2차 요구사항을 반영하여 최종적인 시제품을 완성하기 위해 반복적으로 수행되어야 하는 단계이다. 사전에 프로토타입을 먼저 제작하고 이를 바탕으로 향후 설계될 UI의 적정성을 평가해 수정 보완함으로써 추후 발생 가능한 오류들을 사전에 방지하여 시스템 설계 및 개발에 소요되는 총 비용 및 노력을 절감할 수 있다.

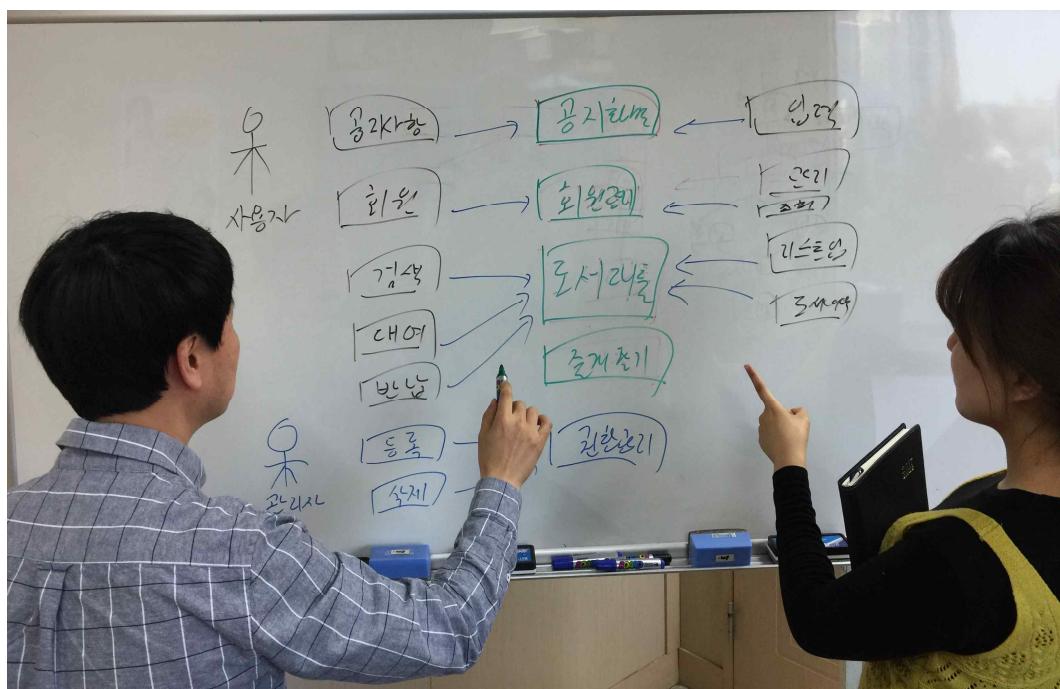
4. 프로토타입의 활용

소프트웨어 개발에 있어 프로토타입은 설명에 대한 시간을 줄여 주며, 특히 고객과 개발 스펙을 논의 시 설득과 이해를 돋기 위해 UI 프로토타입을 만든다. 또한 추가적으로 기술적인 검증을 위해서 프로토타입을 만드는 경우도 있다. 개발 스펙을 작성하다 보면 구현이 가능한지 불가능한지를 미리 예측할 필요가 생기는데 이때 먼저 제작해서 사전 검증을 하는 것이다. 프로토타입 작성 후 확실하지 않으면 수정 보완 후 다시 검증한다. 이렇게 함으로써 실제 개발에 들어가기 전 최대한의 오류를 줄일 수 있다. 프로토타입은 검증을 위한 것이므로 최대한 간단하게 만든다.

② UI 프로토타입 상세

1. UI 프로토타입 전략

확정된 요구사항을 기반으로 UI전략을 실체화하는 과정이며, UI 디자인 작성 이전에 미리 화면을 설계하는 단계이다. 아날로그적인 방법으로 스케치, 그림, 글 등을 손으로 직접 작성하는 페이퍼 프로토타입(Paper Prototype)과 컴퓨터 등 도구를 사용하여 작성하는 디지털 프로토타입(Digital Prototype)이 있다. 환경 및 상황에 따라 적절하게 선택하여 사용하면 된다.



[그림 1-20] 저수준 프로토타입 설계 예시

2. UI 프로토타입의 장점과 단점

(1) 장점

- (가) 사용자 설득과 이해가 쉽다.
- (나) 개발 시간이 감소한다.
- (다) 오류를 사전에 발견할 수 있다.

(2) 단점

- (가) 너무 많은 수정 과정을 거친다면 오히려 작업 시간이 늘어날 수 있다. 사용자의 요구사항은 가능한 들어주되 적절한 태협이 필요하다.
- (나) 자원 효율성 관점에서 보면 필요 이상으로 자원을 많이 소모한다.
- (다) 정확한 문서 작업이 생략될 수 있다.

3 UI 프로토타입 작성 도구 및 방법

<표 1-15> 프로토타입 제작 방법 구분

구분	방법	비고
아날로그	화이트보드, 펜, 종이를 이용 포스트잇 사용	손으로 직접 작성
디지털	파워포인트, 아크로뱃, 비지오, Invision, Marvel, Adobe XD, Flinto, Principle, Keynote, UX pin, HTML	툴을 사용하여 작성

4 UI 프로토타입 작성 시 고려할 사항

1. 프로토타입 계획 작성

프로젝트의 상황에 따라 다르지만 일반적으로 프로토타입 작성은 계획을 수립하는 과정과 실행 후 결과를 보고하는 프로세스로 진행된다. 프로토타입 계획을 세울 때 고려할 부분과 결과서를 작성할 때 고려할 부분을 고려하여야 한다.

2. 프로토타입 범위 확인

프로토타입의 범위는 프로젝트의 범위나 리스크 상황 등의 주변 여건을 감안해서 정해야 한다. 우선 목적을 명확히 하고 그 목적을 수행할 수 있는 환경이 마련되었는지 확인해야 한다. 특히 프로토타입 팀을 별도로 구성할 수 있는지 반드시 확인해야 한다.

3. 프로토타입 목표 확인

프로토타입을 통해서 얻고자 하는 목표를 미리 명확하게 준비해야 한다. 기능과 관련된 것인지, 성능과 관련된 것인지, 개발 환경에 관련된 것인지에 대한 부분을 고객과 협의하여 명확하게 준비하고 진행해야 한다.

4. 프로토타입 기간 및 비용 확인

가급적 프로토타입에 투입되는 기간 및 비용은 최소화하여 목적을 달성할 수 있도록 계획하는 것이 좋다. 검증할 범위를 너무 넓게 잡거나 기간을 많이 잡으면 고객이 원하는 목표가 너무 커져서 오히려 문제가 될 수도 있으므로 주의해야 한다.

5. 프로토타입 산출물 확인

프로토타입에서 나오는 산출물은 실제 개발에 그대로 참조될 수 있는 수준이 되어야 한다. 하지만 프로토타입을 통해 개발된 UI를 실제 개발 범위에 넣는 것은 좋은 방법이 아니다. 실제 기능 요구사항을 가지고 개발되었다 하더라도 아키텍처 요소 검증을 위한 것 이므로 실제 개발에서는 참조만 하는 수준으로 활용해야 한다.

6. 프로토타입 유의 사항 확인

프로토타입은 작은 범위와 적은 인원을 가지고 최소 기간 내에 위험 요소를 식별하고 해결하는 것이 중요하다. 가급적 프로토타입에 투입되는 기간 및 비용을 최소화하여 목적을 달성할 수 있도록 계획하는 것이 좋다. 검증할 범위를 너무 넓게 잡거나 기간을 많이 잡으면 원하는 목표가 너무 커져서 오히려 문제가 될 수도 있으므로 주의해야 한다.

⑤ UI 프로토타입 계획 시 고려할 사항

프로젝트의 상황에 따라 다르지만 일반적으로 프로토타입은 계획을 수립하고 실행한 후에 그 결과를 보고하는 프로세스로 진행된다. 다음과 같이 프로토타입 계획을 세울 때 고려할 부분과 권고 사항을 확인한다.

1. 프로토타입 목표 확인

가장 큰 목표는 아키텍처 검증(성능, 안정성, 개발 생산성 측면)이다. 이외에 각종 가이드 확정, 개발 환경 세팅 완료, 공통 모듈 확보, 인력 양성 등을 들 수 있다. 분석, 설계 기법이 프로젝트 팀원들에게 익숙하지 않은 경우에는 그 개선을 프로토타입의 목적으로 잡는 것을 권고한다.

2. 프로토타입 환경 확인

프로토타입을 위한 솔루션(소프트웨어 확보), 인프라 환경(하드웨어 확보)을 마련해야 한다. 분석과 설계 및 개발, 테스트 가이드 베타 버전을 확인한다. 프로토타입 개발에 필요한 환경(개발 툴, 테스트 툴, 빌드 및 배포 툴, 형상 관리 등)을 마련해야 한다. 인프라 아키텍트(개발자)와 협의하여 가급적 실제와 가까운 프로토타입 인프라 환경을 구축하는 것이 좋다. 대형 프로젝트의 경우 개발용 서버를 미리 도입하여 진행하는 것도 좋은 방법이다.

3. 프로토타입 일정 확인

일반적으로 아키텍처가 확정된 이후 프로젝트의 실제 분석 작업이 완료되기 이전에 진행 한다. 프로토타입의 목표를 아키텍처 검증만으로 한다면 프로젝트 개발 이전에 완료해도 무방하다. 대형 프로젝트를 기준으로 프로토타입은 대략 1개월 정도로 잡는 것이 좋다. 분석, 설계 가이드에 대한 검증을 목적으로 기간을 잡을 경우 2개월을 추가할 수도 있다.

4. 프로토타입 범위 확인

아키텍처의 핵심이 되는 요소(UI)를 프로토타입의 범위로 잡는다. 아키텍처 요소 중에 위험이 많은 요소(검증되지 않은 요소와의 연동 등)를 범위로 잡을 수 있다. 핵심이 되는 요소를 판단할 때에는 많은 개발자들이 참여하여 개발하는 부분인가 하는 점을 기준으로 삼는다.

5. 프로토타입 인원 확인

프로토타입 역할은 리더, 솔루션 담당자, 인프라 담당자, 개발 환경 리더, 공통 모듈 개발자, 프로토타입 개발자가 있다. 대형 프로젝트를 기준으로 리더 1인, 솔루션 담당자 파트타임 2인 이상, 인프라 담당자 파트타임 1인, 개발 환경 리더 겸 공통 모듈 개발자 1인, 프로토타입 개발자 3~4인으로 구성된다.

6. 프로토타입 아키텍처 검증 확인

기 수립된 아키텍처로 주어진 비즈니스 요구사항을 모두 만족할 수 있는지 검증한다. 인프라 환경이 가능하다면 아키텍처 요소 간의 성능을 측정하여 결과 보고한다. 아키텍처에 대한 검증 요소는 품질속성별로 존재할 수 있으나 프로토타입을 통해서 모든 품질속성을 보여줄 수는 없다. 일반적으로 기능 요구사항을 만족시킬 수 있는지 여부와 일부 구간의 성능 측정을 하는 것을 권고한다.

7. 프로토타입 이슈 및 해결

프로토타입을 통해서 발생하는 이슈를 모두 취합하여 보고한다. 프로토타입에서 나오는 이슈의 대부분은 아키텍처 요소 검증 중에 발생하며 분석, 설계 이슈와 개발 환경 등의 이슈가 추가될 수 있다. 프로토타입은 이슈가 많이 발생할수록 좋은 것이다. 따라서 프로토타입을 통해서 발생한 이슈와 해결한 이슈의 종류별 개수를 취합하여 결과 보고하는 것이 좋다. 프로토타입 리더가 날마다 이슈를 취합하고 해결 방법을 제시한다. 이것을 모두 정리하여 결과 보고에 반영한다.

8. 프로토타입 가이드 확정

프로토타입에서 검증하려고 한 표준 가이드(분석, 설계, 개발, 테스트 가이드)를 프로토타입을 하면서 수정하여 최종 확정한다. 프로토타입은 실제 개발과 유사하여 각종 가이드를 실전에 가장 가깝게 만들 수 있으므로 가능한 모든 가이드를 적용하는 것이 좋다.

9. 프로토타입 개발 생산성 확인

프로토타입을 진행하면서 가장 많은 시간이 소요되는 구간(분석, 설계, 개발, 테스트 등)을 찾아 그 원인을 찾고 시정할 수 있는 방법을 찾아 실제 프로젝트에 적용하면 많은 시간을 절약할 수 있다. 프로토타입을 진행하면서 프로토타입 개발자들이 분석, 설계, 개발, 테스트하는 시간을 분 단위로 적게 하여 매일 취합한다. 가장 많은 시간이 소요되는 부분을 찾고 그 원인을 분석하여 해결 방법을 제시한다.

10. 프로토타입 결과 시연

프로토타입의 결과(화면 위주)를 고객, PM, PL 개발자에게 시연한다. 프로토타입의 목적을 구체적으로 설명한다. 개발이 완료된 화면 위주로만 시연하지 말고 분석, 설계, 개발, 테스트 과정을 모두 설명하면서 시연하는 것이 중요하다. 확정된 가이드 및 개발 환경 구조, 그리고 재활용이 가능한 공통 모듈 등을 같이 소개하는 것이 좋다.

수행 내용 / UI 프로토타입 제작, 검토하기

재료 · 자료

- 프로토타입 작성 사례, 사용자 행동 흐름 사례

기기(장비 · 공구)

- 종이와 펜, 컴퓨터, 파워포인트(PowerPoint), 메모장 등의 HTML 편집 도구

안전 · 유의 사항

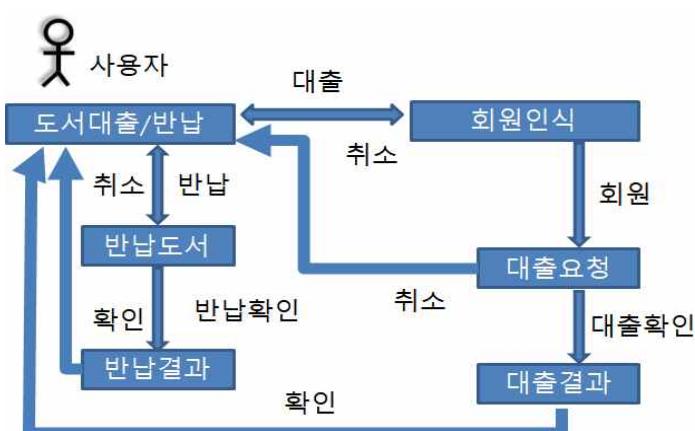
- 페이지 프로토타입 작성 시 바른 글과 그림에 유의한다.
- 디지털 프로토타입 작성 시 가장 숙달된 도구를 사용한다.
- UI 요구사항과 설계 원칙을 준수하도록 고려한다.
- UI 요구사항과 설계 과정에서 발생 가능한 위험에 대해 고려한다.

수행 순서

① 응용 소프트웨어 개발을 위한 UI 표준 및 지침에 의거하여, 소프트웨어 아키텍처의 설계 원리를 확인한다.

- 구축할 시스템의 인터페이스 설계 시 소프트웨어 아키텍처의 설계 원리가 어떻게 사용될 수 있는지 확인한다.
여기에서는 도서대출 예약시스템을 예시로 삼는다.

(1) UI 흐름을 확인한다.



[그림 1-21] 사용자 인터페이스 흐름 예시

(2) 사용자의 행동 흐름을 확인한다.

사용자의 행동 확인 (도서대출 예약시스템의 예)

- (가) 사용자가 도서대출 시의 행동을 확인한다.
- (나) 사용자가 도서반납 시의 행동을 확인한다.
- (다) 관리자의 도서대출 시의 행동을 확인한다.
- (라) 관리자의 도서반납 시의 행동을 확인한다.

(3) 행동에 따른 설계 원리를 생각하고 처리 흐름을 확인한다.

사용자의 행동에 따른 흐름 (도서대출 예약시스템의 예)

- (가) 사용자가 도서대출 행동 후 결과를 확인한다.
- (나) 사용자가 도서반납 행동 후 결과를 확인한다.
- (다) 관리자의 도서대출 행동 후 결과를 확인한다.
- (라) 관리자의 도서반납 행동 후 결과를 확인한다.

2. 개발 요청자가 가장 이해하기 편한 방법이 무엇인지 확인한다.

(1) 페이퍼 프로토타입이 적합한지 확인한다.

페이퍼 프로토타입을 적용하면 좋은 경우

- 제작 기간이 짧은 경우 적용한다.
- 제작 비용이 적을 경우 적용한다.
- 업무 협의가 빠른 상황일 경우 적용한다.

(2) 디지털 프로토타입이 적합한지 확인한다.

디지털 프로토타입을 적용하면 좋은 경우

- 재사용이 필요한 경우 적용한다.
- 산출물과 비슷한 효과를 필요로 할 경우 적용한다.
- 숙련된 전문가가 있을 경우 적용한다.

② UI 설계 원리를 바탕으로 프로토타입 유스케이스(USE CASE)를 작성한다.

1. 유스케이스를 작성한다.

(1) 유스케이스의 개요를 작성한다.

(2) 액터를 정의한다.

(3) 이벤트 흐름을 작성한다.

(가) 기본 사항을 작성한다.

(나) 추가 사항을 작성한다.

- (4) 처리 내용을 작성한다.
- (5) 도서 대출 예약 시스템의 도서 대출 유스케이스 사례와 같이 작성한다.

도서 대출 예약 시스템의 도서 대출 유스케이스(USE CASE)

1. 개요

사용자는 대출반납기기를 통하여 원하는 도서를 대출한다.

2. 액터

사용자

3. 이벤트 흐름

(1) 기본 사항

- (가) 사용자는 ‘도서대출/반납메인화면’에서 대출 버튼을 누른다.
- (나) 시스템은 사용자에게 회원카드의 인식을 요청하는 ‘회원카드인식요청화면’을 출력한다.
- (다) 사용자는 회원카드를 대출반납기기에 인식시킨다.
- (라) 시스템은 대출할 도서의 인식을 요청하는 ‘대출도서인식요청화면’을 출력한다.
- (마) 사용자는 대출하고자 하는 도서를 대출반납기기에 인식시킨다.
- (바) 시스템은 데이터베이스에 대출정보를 저장한다.
- (사) 시스템은 대출 결과를 보여주는 ‘대출결과화면’을 출력한다.
- (아) 사용자는 확인 버튼을 누른다.

(2) 추가 사항

- (가) 사용자가 ‘회원카드인식요청화면’에서 취소를 누를 경우
- (나) 시스템은 ‘도서대출/반납메인화면’을 출력한다.
- (다) 사용자가 ‘대출도서인식요청화면’에서 취소를 누를 경우
- (라) 시스템은 ‘도서대출/반납메인화면’을 출력한다.
- (마) 사용자가 ‘대출결과화면’에서 영수증 출력을 누를 경우
- (바) 시스템은 ‘도서대출/반납메인화면’을 출력한다.

4. 처리 내용

데이터베이스에 대출 정보가 저장된다.

(6) 도서 대출 예약 시스템의 도서 반납 유스케이스 사례와 같이 작성한다.

도서 대출 예약 시스템의 도서 반납 유스케이스(USE CASE)

1. 개요

사용자는 대출반납기기를 통하여 원하는 도서를 반납한다.

2. 액터

사용자

3. 이벤트 흐름

(1) 기본 사항

- (가) 사용자는 ‘도서대출/반납메인화면’에서 반납 버튼을 누른다.
- (나) 시스템은 반납할 도서의 인식을 요청하는 ‘반납도서인식요청화면’을 출력한다.
- (다) 사용자는 반납하고자 하는 도서를 대출반납 기기에 인식시킨다.
- (라) 시스템은 데이터베이스에서 대출 정보를 수정한다.
- (마) 시스템은 반납 결과를 보여주는 ‘반납결과화면’을 출력한다.
- (바) 사용자는 확인 버튼을 누른다.

(2) 추가 사항

- (가) 사용자가 ‘반납도서인식요청화면’에서 취소를 누를 경우
- (나) 시스템은 ‘도서대출/반납메인화면’을 출력한다.

4. 처리 내용

데이터베이스에 대출 정보가 수정된다.

③ UI 요구사항을 반영한 프로토타입을 제작한다.

1. 실질적인 제작에 앞서 UI 프로토타입 제작 단계를 숙지한다.

(1) 1단계

사용자 요구사항을 분석하는 단계이다. 작업 담당자는 기본적인 요구사항이 도출되어 확정되기 전까지 사용자 관점에서 사용자처럼 수행한다.

(2) 2단계

작업 담당자가 위의 1단계에서 도출된 요구사항을 충족하는 프로토타입을 종이 등에 손으로 직접 그리거나 프로그래밍 언어나 편집 도구 등의 디지털 방식을 이용하여 개발한다. 프로토타입은 개발이 필요한 시스템의 핵심적인 기능을 중심으로 개발한다.

(3) 3단계

작성 또는 개발된 프로토타입을 사용자가 실제적으로 사용하여 요구사항이 잘 수행되고 있는지를 확인하는 과정이다. 프로토타입의 추가 사항이나 보완을 위해 다양한 제

안 과정이 있는 단계이다.

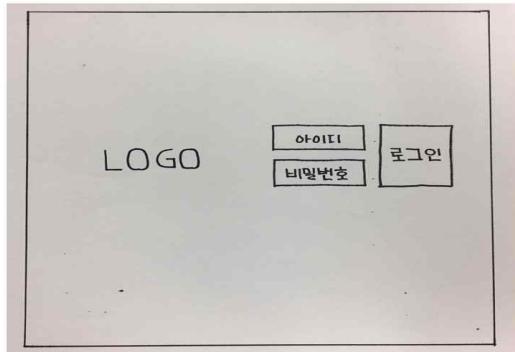
(4) 4단계

프로토타입의 결과를 토대로 수정과 합의가 이루어지는 단계이다. 작업 담당자는 사용자가 요청한 다양한 제안 사항을 포함하고 수용하여 보완 작업을 한다. 프로토타입 결과물이 완성된 후에는 3단계로 다시 되돌아간다. 사용자의 최종 승인 완료까지 3단계와 4단계는 반복된다.

2. 프로토타이핑(Prototyping)을 초기 작성한다.

(1) 페이퍼 프로토타이핑을 초기 작성한다.

종이와 펜을 활용해 화면의 구조를 스케치하는 방법으로 가장 쉽게 만들 수 있는 형태이다. [그림 1-22]와 같이 UI 페이퍼 프로토타이핑을 초기 작성한다.



[그림 1-22] UI 페이퍼 프로토타이핑: 로그인 화면

(2) 디지털 프로토타이핑을 초기 작성한다.

디지털 편집기(자신이 가장 잘 사용하는 편집 프로그램), HTML 등의 프로토타이핑 도구를 사용해 화면의 구조를 만들어 내는 방법으로 가장 폭넓게 사용된다.

(가) 디지털 편집기 기반 프로토타이핑을 초기 작성한다.

[그림 1-23]과 같이 화면의 구조를 디지털 편집 도구를 활용해 프로토타이핑을 초기 작성한다.



[그림 1-23] UI 디지털 프로토타이핑: 로그인 화면

(나) HTML 기반 디지털 프로토타이핑을 초기 작성한다.

HTML을 이용한 프로토타이핑은 기본적으로 HTML, CSS, JavaScript 등과 같은 프로그래밍 언어들을 이해하고 활용한다. HTML을 이용한 프로토타이핑의 경우 다음의 절차를 통해 진행한다.

- 1) 프로토타이핑 대상이 되는 화면의 구조와 구성 요소들을 파악한다.
- 2) HTML의 다양한 태그들을 통해 화면의 레이아웃을 구상한다.
- 3) 적절한 HTML 화면 구성 요소들을 이용해 각 UI 표현 요소들을 표현한다.

```
<html>
<head>
    <title> 로그인 </title>
</head>

<body>
    <h1>Login</h1>
    <p><input type = "text" name = "login" value = "" placeholder = "아이디"></p>
    <p><input type = "password" name = "password" value = "" placeholder = "비밀번호"></p>
    <p>
        <label>
            <input type = "checkbox" name = "auto_login" id = "auto_login">
            로그인 상태 유지
        </label>
    </p>
    <p><input type = "submit" name = "commit" value = "로그인"></p>
    <div class = "new_account">
        <p> 아이디가 없습니까? <a href = ""> 회원가입 시작 </a></p>
    </div>
    <div class = "find_password">
        <p> 비밀번호를 잊으셨습니까? <a href = ""> 여기</a>를 눌러 확인 </p>
    </div>
</body>
</html>
```

[그림 1-24] HTML 기반 UI 디지털 프로토타이핑: 로그인 화면용 소스코드

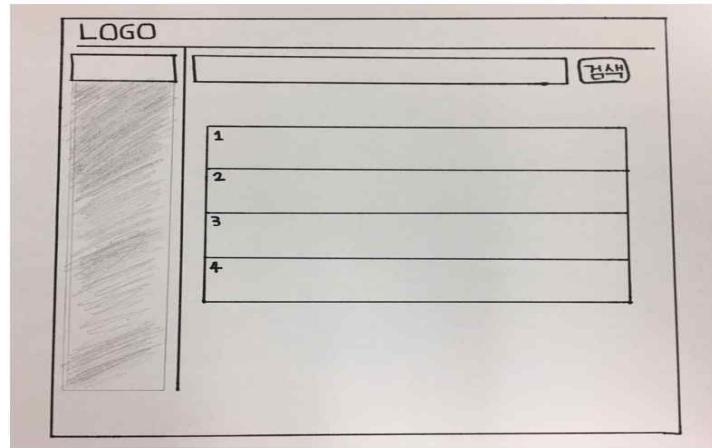
[그림 1-24]는 HTML을 활용해 화면의 구조를 프로토타이핑한 예시이다. 메모장 등의 문서 편집기를 통해 [그림 1-24]와 같은 HTML 코드를 작성한 뒤 .html 확장자를 붙이고 인터넷 브라우저에서 실행한다.

3. 프로토타이핑을 상세 수행한다.

(1) 페이퍼 프로토타이핑을 상세 수행한다.

(가) 앞서 초기 작성한 페이퍼 프로토타입을 바탕으로 UI 페이퍼 프로토타이핑을 상세 수행한다.

- 1) UI 요구사항을 바탕으로 각 화면 구조를 구상한다.
 - 어떤 형태의 배치를 통해 각 기능을 표현할 것인지 각 화면별로 구상한다.
- 2) 화면에 표현되는 각 요소들을 설계한다.
 - 각 입력 요소들의 형태와 입력 방법에 대해서 구상한다.
 - 각 화면 요소들의 배치에 대해서 구상한다.
 - 각 요소들을 설계한다.



[그림 1-25] UI 페이퍼 프로토타이핑: 도서 검색 화면

(2) 디지털 프로토타이핑을 상세 수행한다.

- (가) 앞서 초기 작성한 디지털 프로토타입을 바탕으로 파워포인트 기반 디지털 프로토타이핑을 상세 수행한다.
구체적인 방법은 앞의 “(1) 페이퍼 프로토타이핑”의 경우와 같다.



[그림 1-26] UI 디지털 프로토타이핑: 도서 검색 화면

- (나) 앞서 초기 작성한 디지털 프로토타입을 바탕으로 HTML 기반 디지털 프로토타이핑을 상세 수행한다.
구체적인 방법은 앞의 “(1) 페이퍼 프로토타이핑”의 경우와 같다.



[그림 1-27] HTML 기반 UI 디지털 프로토타이핑: 로그인 화면

④ 작성한 프로토타입을 활용하여 UI/UX 엔지니어와 향후 적용할 UI의 적정성에 대해 검토한다.

1. 실행 차를 줄이기 위한 UI 적정성을 확인한다.

(1) 사용 의도를 파악한다.

구축할 시스템의 설계 시 필요한 기능들을 추론해 본다. 또한 발견된 기능들 가운데 불필요할 수 있는 부가 기능들을 구분한다.

(2) 행위 순서를 규정한다.

구축할 시스템의 설계 과정에서 사용자가 원하는 도서를 찾을 수 있는 최대한 다양한 방법을 제공하고자 한다. 어떤 단어가 검색어로 사용될 수 있는지 가능한 한 많은 검색 방법을 확인한다.

(3) 행위의 순서대로 실행한다.

구축할 시스템의 로그인 화면을 설계하는 경우 올바른 커서의 시작 위치를 확인한다.

2. 평가 차를 줄이기 위한 UI 적정성을 확인한다.

(1) 수행한 키 조작의 결과를 사용자가 빠르게 지각 가능한지 확인한다.

도서 대출 예약 시스템의 키 조작 후 결과 확인

구축할 시스템에서 사용자의 요청으로 도서 검색이 시작된 후부터 도서 검색이 완료되기 전까지 시스템이 어떤 반응을 보이는 것이 적절할지 확인한다.

(2) 키 조작으로 변화된 시스템의 상태를 사용자가 쉽게 인지 가능한지 확인한다.

도서 대출 예약 시스템의 화면 키 조작 후 상태 인지 능력 확인

구축할 시스템에서 다수의 사용자가 동시 접속함에 따라 사용자 A가 원하는 도서를 검색한 후 예약을 진행하는 동안 해당 도서를 뒤늦게 검색한 다른 사용자 B가 A보다 먼저 예약을 시도할 경우, 시스템은 사용자 B에게 다른 사용자가 대출 예약 중이라는 메시지를 출력하여 사용자 B에게 알려 주어야 하며, 사용자 A에게는 대출 처리중이라는 사실을 알 수 있도록 확인한다.

- (3) 사용자가 가진 원래 의도와 시스템 결과 간의 유사 정도를 사용자가 쉽게 파악 가능하지 확인한다.

도서 대출 예약 시스템의 유사 정도 확인

구축할 시스템에서 사용자가 이미 대출 중인 도서를 예약하고자 검색한 경우, 시스템이 결과를 어떻게 알려주어야 사용자가 가진 원래 의도에 맞출 수 있는지에 대해 토론한다.

수행 tip

- UI 전체 구조에 대해 먼저 고려한 뒤 세분화된 페이지에 대해 고려하는 것이 효율적이다.
- UI 요구사항을 만족시키기 위해서는 유스케이스를 통해 사용자와 시스템의 행위를 먼저 이해하고 나서 주요 맥락을 검토하는 것이 중요하다.
- HTML을 이용한 프로토타이핑을 위해서는 HTML 및 CSS와 같은 프로그래밍 언어에 대한 학습이 선행되어야 한다.
- 프로토타이핑을 잘 하기 위해서는 다양한 형태의 화면을 토대로 연습을 진행해 보는 것이 중요하다.

학습1 교수 · 학습 방법

교수 방법

- 유스케이스, HTML 언어에 대한 이해도를 파악한 후 수업을 진행한다.
- 유용성에 관한 설계 원리 관점에서 기존 UI의 좋은 예시와 나쁜 예시에 대한 PPT 자료와 함께 설명한다.
- 다양한 사용 사례 예시 및 UI 프로토타이핑 예시에 대한 PPT 자료와 함께 설명한다.
- UI 프로토타이핑의 필요성에 대해 이해할 수 있도록 예시를 들어 설명한다.
- 다양한 UI 프로토타이핑 도구와 활용 방법에 대해 지도한다.
- UI 프로토타이핑 과정에 대해 단계적 실습이 이루어질 수 있도록 지도한다.

학습 방법

- 유용성에 관한 설계 원리 관점에서 기존 UI들을 평가한다.
- 유스케이스를 통해 사용자와 시스템 간의 상호 작용을 확인한다.
- 다양한 유스케이스를 숙지한다.
- 유스케이스로부터 UI 요구사항들을 도출한다.
- UI 프로토타이핑의 목적과 활용에 대해 숙지한다.
- UI 요구사항들을 반영한 UI 페이퍼 프로토타이핑 과정에 대해 실습한다.
- UI 요구사항들을 반영한 UI 디지털 프로토타이핑 과정에 대해 실습한다.
- 여러 형태의 UI 프로토타이핑을 연습한다.

학습1 평 가

평가 준거

- 평가자는 학습자가 학습 목표를 성공적으로 달성하였는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.

학습 내용	학습 목표	성취수준		
		상	중	하
UI 요구사항 확인	- 응용 소프트웨어 개발을 위한 UI 표준 및 지침에 의거하여, 개발하고자 하는 응용 소프트웨어에 적용될 UI 요구사항을 확인할 수 있다.			
UI 프로토타입 제작 검토	- 응용 소프트웨어 개발을 위한 UI 표준 및 지침에 의거하여, UI 요구사항을 반영한 프로토타입을 제작할 수 있다. - 작성한 프로토타입을 활용하여 향후 적용할 UI의 유용성에 대해 검토할 수 있다.			

평가 방법

- 포트폴리오

학습 내용	평가 항목	성취수준		
		상	중	하
UI 요구사항 확인	- 사용 사례에 대한 이해 능력			
UI 프로토타입 제작 검토	- 사용 사례에 따른 UI 요구사항 도출 능력 - UI 요구사항을 반영한 페이퍼 프로토타입 제작 능력 - UI 요구사항을 반영한 디지털 프로토타입 제작 능력			

- 평가자 체크리스트

학습 내용	평가 항목	성취수준		
		상	중	하
UI 요구사항 확인	- 사용 사례에 대한 이해 능력			
	- 사용 사례에 따른 UI 요구사항 도출 능력			
UI 프로토타입 제작 검토	- UI 요구사항을 반영한 페이퍼 프로토타입 제작 능력			
	- UI 요구사항을 반영한 디지털 프로토타입 제작 능력			

피드백

1. 포트폴리오

- 실습한 사용 사례 결과에 대해 평가한 후에 주요 사항에 대하여 표시하여 돌려준다.
- 실습한 사용 사례에 따른 UI 요구사항 도출 능력 결과에 대해 평가한 후에 주요 사항에 대하여 표시하여 돌려준다.
- 실습한 UI 페이퍼 프로토타이핑 결과에 대해 평가한 후에 주요 사항에 대하여 표시하여 돌려준다.
- 실습한 UI 디지털 프로토타이핑 결과에 대해 평가한 후에 주요 사항에 대하여 표시하여 돌려준다.

2. 평가자 체크리스트

- 결과물을 제출받아 사용 사례에 대한 이해 정도에 대해 평가한 후에 필요한 사항 등을 정리하여 돌려준다.
- 사용 사례에 따른 UI 요구사항 도출 수준에 대해 평가한 후에 필요한 사항 등을 정리하여 돌려준다.
- 결과물을 제출받아 페이퍼 프로토타입 제작 능력 이해 정도에 대해 평가한 후에 필요한 사항 등을 정리하여 돌려준다.
- 결과물을 제출받아 디지털 프로토타입 제작 능력 이해 정도에 대해 평가한 후에 필요한 사항 등을 정리하여 돌려준다.

2-1. UI 흐름 설계

학습 목표

- UI 요구사항과 UI 표준 및 지침에 따라, 화면과 품의 흐름을 설계하고, 제약사항을 화면과 품 흐름 설계에 반영할 수 있다

필요 지식 /

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 해당 되는 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.

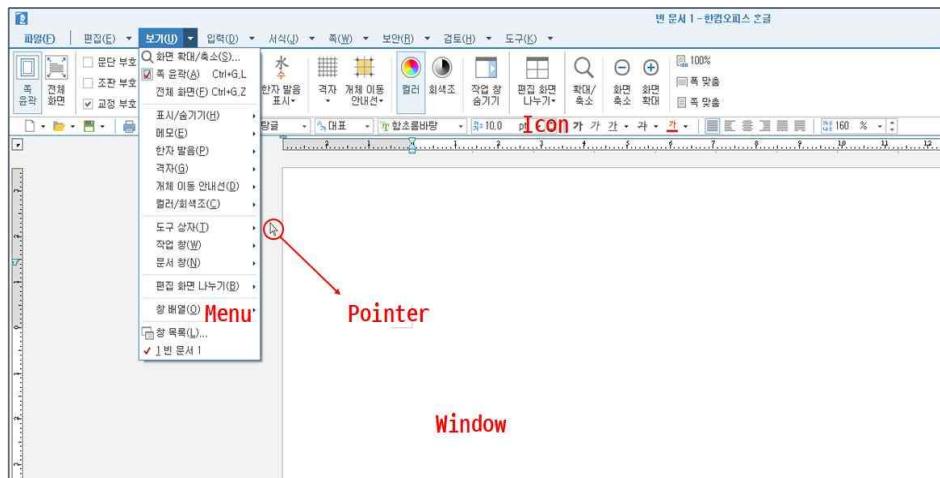
② UI 화면 설계의 기본 구성 요소

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.

- 평가 차, 즉 시스템의 실행 결과와 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 내용 / UI 흐름설계하기

재료 · 자료

- Use Case 사례, 기능 및 폼 설계 사례

기기(장비 · 공구)

- 컴퓨터, 파워포인트(PowerPoint), 메모장 등의 HTML 편집 도구

안전 · 유의 사항

- UI 화면 설계 시 기능과 비기능적 요소에 대해 고려한다.

- 기능 버튼은 최대한 간결하고 명확하게 인지 가능한지 고려한다.
- 기능 버튼의 규칙은 사용자의 요구가 최대한 반영되도록 고려한다.
- 작성된 스토리보드를 중심으로 UI 설계를 고려한다.

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.

(3) 행위의 순서대로 실행한다.

(가) 각 기능을 사용하기 위해 필요한 사전 행위들을 바탕으로 UI 설계서 화면 설계 결과에서 편집 창(Editor)이 활용되는 경우 커서와 포인터 중 무엇이 먼저 활성화되어야 하는지 결정한다.

(나) 각 기능을 사용하기 위해 필요한 사전 행위들을 바탕으로 UI 설계서 화면 설계 결과에서 여러 개의 편집 창(Editor)이 활용되는 경우 올바른 커서의 시작 위치를 결정한다.

(다) 각 기능을 사용하기 위해 필요한 사전 행위들을 바탕으로 UI 설계서 화면설계 결과에서 메뉴(Menu)가 활용되는 경우 초기 값으로 무엇이 먼저 활성화되어야 하는지 결정한다.

(라) 그 외 초기 값 설정이 필요한 다른 요소들이 없는지 확인한다.

2. 평가 차를 줄이기 위한 UI 설계 원리를 확인한다.

(1) 수행한 키 조작의 결과를 사용자가 빠르게 지각 가능하지 확인한다.

(가) UI 설계서의 각 화면 설계 결과에서 각 메뉴/아이콘/원도우가 활성화될 경우 적절한 포인터의 형태를 결정한다.

(나) UI 설계서의 각 화면 설계 결과에서 각 기능이 수행되는 동안 적절한 포인터의 형태를 결정한다.

(2) 키 조작으로 변화된 시스템의 상태를 사용자가 쉽게 인지 가능하지 확인한다.

시스템의 상태 정보가 가능한 한 단순하게, 그리고 이해하기 쉽게 제시되어 있는지 확인한다.

(3) 사용자가 가진 원래 의도와 시스템 결과 간의 유사 정도를 사용자가 쉽게 파악 가능하지 확인한다.

사용자가 가진 원래 의도가 시스템을 통해 충족되었는지 또는 충족될 수 있는지를 사용자가 쉽게 파악할 수 있도록 설계되어 있는지 확인한다.

② UI 요구사항과 UI 표준 및 지침에 따라, 화면과 품의 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.

(2) 비기능적 요구사항을 검토한다.

- (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 조사 한다.

(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)는 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 책 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 책에 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

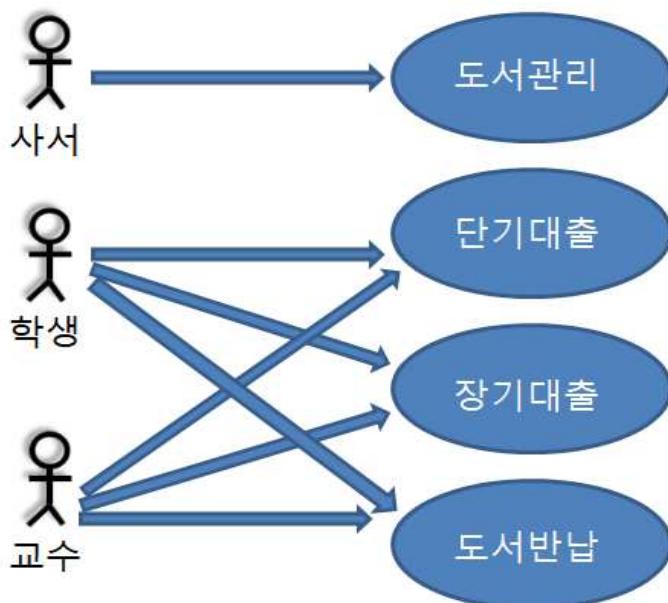
앞서 작성한 도서 대출 예약 시스템 중 도서 반납에 관한 사용 사례를 바탕으로 각 화면 간 이동과 흐름들을 확인한다.

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.

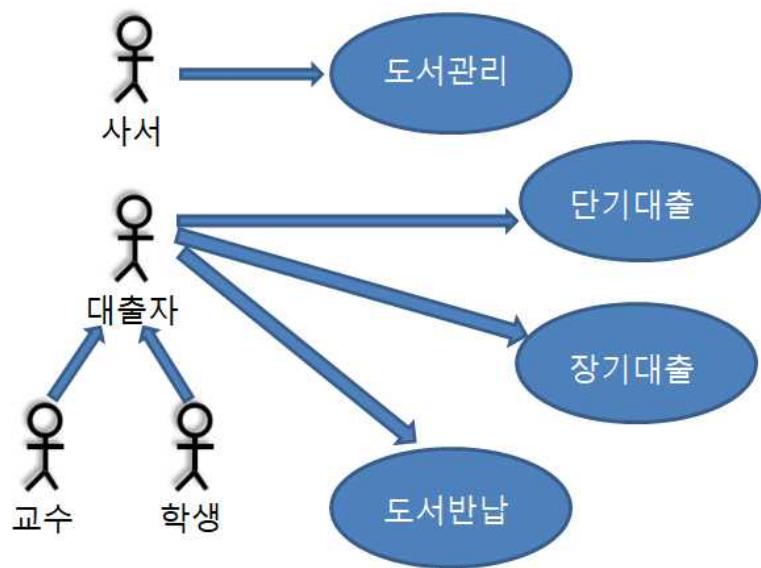
각각의 액터가 어떤 행위를 하는지 작성해 본다.



[그림 2-2] 액터의 행위 사례(도서 관리 시스템)

(2) UI 요구사항을 바탕으로 액터를 세분화한다.

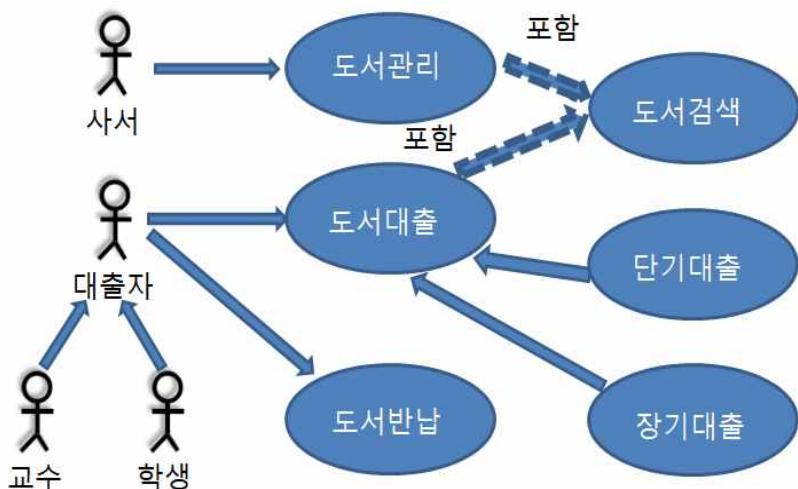
액터의 상호 작용을 생각하고 비슷한 행위를 하는 액터를 그룹으로 지정하여 부모 액터를 정의한다. 아래의 [그림 2-3]은 도서 관리 시스템에서 대출자 액터가 학생과 교수의 부모 액터임을 나타내는 사례이다.



[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.



[그림 2-5] Input Box

(가) Input Box는 Edit 가능 여부를 구분하여 CSS를 적용한다.

- Edit 가능: 입력이 가능하다는 표시이며, 경계선과 배경색으로 흰색을 사용한다.
(CSS명: FormText)
- readonly: 보기 전용 입력란과 일시적 비활성화 입력란에 배경색을 사용한다.
(CSS명: ReadOnlyText)

(나) 필드(Field)의 길이는 자료의 길이와 일치하게 만든다.

(다) 텍스트(Text) 정렬 방식은 가운데 정렬한다.

(라) 라벨(Label)은 좌측 정렬한다.

(마) 필수 입력 항목을 구분해 Label에 다음과 같이 CSS 적용한다.

- 일반 입력 Label HTML <th>/th>
- 필수 입력 Label : Label Text 뒤에 * 표시 넣음.(CSS명: MandatoryIcon)

(2) Combo Box를 확인하고, 규칙을 정의한다.



[그림 2-6] Combo Box

(가) 선택 항목일 경우 Default 값이 없을 경우는 “전체” 혹은 “선택” 의 기본 문구로 통일한다.

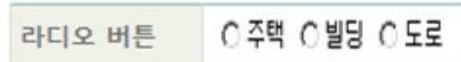
“전체” 는 주로 검색 조건에 많이 쓰이고, “선택” 은 주로 입력 항목에 많이 쓰인다.

(나) 콤보 박스 버튼의 초기 설정 값은 자주 사용하는 값을 우선으로 한다.

상황에 따라서는 버튼 값은 비워 두고 사용자가 선택하도록 한다.

(다) 콤보 박스 버튼을 클릭하면 별도의 저장 버튼을 눌러서 저장이 되도록 한다.

(3) Radio Box를 확인하고, 규칙을 정의한다.



[그림 2-7] Radio Box

라디오 버튼은 많은 메뉴들 중에서 한 개만 선택이 가능하도록 강제하는 버튼이다.

(가) 라디오 버튼을 사용할 경우 텍스트와의 정렬이 맞도록 CSS를 적용한다. (css명: formradio)

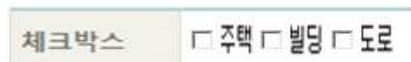
(나) 라디오 버튼의 초기 설정 값은 자주 사용하는 값을 우선으로 한다.

상황에 따라서는 버튼 값은 비워 두고 사용자가 선택하도록 한다.

(다) 라디오 버튼을 클릭하면 자동으로 선택된 값이 저장되도록 한다.

별도의 저장 버튼을 누름으로써 작동이 되는 방식도 무방하다.

(4) Check Box를 확인하고, 규칙을 정의한다.



[그림 2-8] Check Box

체크 박스 버튼은 많은 메뉴들 가운데 한 개 또는 여러 개를 동시에 선택이 가능하게끔 하는 버튼이다.

(가) 체크 박스 버튼을 사용할 경우 텍스트와의 정렬이 맞도록 CSS를 적용한다. (CSS명: formcheck)

(나) 체크 박스 버튼의 초기 설정 값은 자주 사용하는 값을 우선으로 한다.

상황에 따라서는 버튼 값은 비워 두고 사용자가 선택하도록 한다.

(다) 체크 박스 버튼을 클릭하면 별도의 저장 버튼을 눌러서 저장이 되도록 한다.

수행 tip

- UI 전체 구조에 대해 먼저 고려한 뒤 세분화된 페이지에 대해 고려하는 것이 효율적이다.
- UI 요구사항을 만족시키기 위해서는 사용 사례를 통해 주요 맥락을 검토하는 것이 중요하다.
- UI 설계 과정에서 발생 가능한 기술적 문제에 대해 사전에 충분히 고민하는 것이 효율적이다.
- UI 화면 설계를 잘 하기 위해서는 다양한 형태의 화면을 토대로 연습을 진행해 보는 것이 중요하다.

2-2. UI 상세설계

학습 목표

- UI 요구사항과 UI 표준 및 지침에 따라, 사용자의 편의성을 고려한 메뉴 구조를 설계할 수 있다.
- UI 요구사항과 UI 표준 및 지침에 따라, 하위 시스템 단위의 내·외부 화면과 품을 설계할 수 있다.

필요 지식 /

① UI 시나리오 작성 원칙

UI 상세설계에 있어 시나리오 작성은 반드시 필요한 사항이다. 정보통신산업진흥원 부설 SW공학센터의 “소프트웨어 개발 UI/UX 참조모델 가이드”(2014)에 따르면 시나리오 작성의 원칙은 다음과 같이 설명한다.

1. UI의 전체적인 기능과 작동 방식을 개발자가 한눈에 쉽게 이해 가능하도록 구체적으로 작성하여야 한다.
2. 모든 기능은 공통 적용이 가능한 UI 요소와 인터랙션을 일반적인 규칙으로 정의한다.
3. “대표 화면의 레이아웃과 그 화면들 속의 기능”을 정의한다.
이때의 대표 화면은 시나리오에 포함되는 서로 다른 형태를 가진 독립적인 화면들을 가리킨다.
4. 인터랙션의 흐름을 정의하며, 화면 내와 화면 간 인터랙션의 순서(Sequence), 분기(Branch), 조건(Condition), 루프(Loop) 등을 명시한다.
이때의 인터랙션은 페이퍼 프로토타입에서 발견된 문제점을 모두 개선하여 적용한 최종 인터랙션이어야 한다.
5. 예외 상황에 대비한 케이스를 정의한다.
대부분의 소프트웨어 개발자와 품질 관리자들이 UI 시나리오 문서에서 가장 많은 불만을 드러내는 부분이 예외 케이스의 정리가 부실하다는 것이다.
6. UI 일반 규칙을 지키면서 기능별 상세 기능 시나리오를 정의한다.
7. UI 시나리오 규칙을 지정한다.

<표 2-1> 프로토타입 제작 구분

구분	설명
주요 키의 위치와 기능	화면상에 공통적으로 배치되는 주요 키의 위치와 기능을 설명한 것으로 여러 화면 간의 일관성을 보장하기 위한 것이다.
공통 UI 요소	체크 박스, 라디오 버튼, 스크롤바, 텍스트 입력 필드, 상하/좌우 훨, 모드 설정, 탭, 팝업 등의 각 UI 요소를 언제 사용하며 어떤 형태인지 정의하고 사용자의 조작에 어떻게 반응하는지 그 흐름을 상세하게 설명한 것이다.
기본 스크린 레이아웃 (Basic Screen Layouts)	여러 화면 내에 공통적으로 나타나는 Indicators, Titles, Ok/Back, Soft Key, Option, Functional Buttons 등의 위치와 속성을 정의한 것으로서 여러 기능들 간에 화면 레이아웃의 일관성을 보장하기 위한 것이다.
기본 인터랙션 규칙 (Basic Interaction Rules)	터치 제스처 등의 공통적으로 사용되는 조작의 방법, 홈 키의 동작 방식과 같은 운항 규칙, 실행, 이전, 다음, 삭제, 이동 등의 화면 전환 효과 등에 대해 기술한 것이다.
공통 단위 태스크 흐름 (Task Flows)	많은 기능들에 공통적으로 자주 나타나는 삭제, 검색, 매너 모드 상태에서의 소리 재생 등의 인터랙션 흐름을 설명한 것이다.
케이스 문서	다양한 상황에서의 공통적인 시스템 동작에 대해 정의한 문서이다. (ex. 사운드, 조명, 이벤트 케이스 등)

출처 : 정보통신산업진흥원 부설 SW공학센터. 소프트웨어 개발 UI/UX 참조 모델 가이드 참조

② UI 시나리오 문서 작성의 요건

정보통신산업진흥원 부설 SW공학센터의 “소프트웨어 개발 UI/UX 참조모델 가이드”(2014)에 따르면 시나리오 작성의 요건은 다음과 같이 설명한다.

1. 완전성(Complete)

- (1) (누락 없이) 완전해야 한다.
- (2) 최대한 빠짐없이 가능한 한 상세하게 기술한다.
- (3) 시스템 기능보다 사용자의 태스크에 초점을 맞춰 기술한다.

2. 일관성(Consistent)

- (1) 일관성이 있어야 한다(서비스에 대한 목표, 시스템 및 사용자의 요구사항).
- (2) 모든 문서의 UI 스타일(Flow 또는 Layout)을 일관적으로 구성한다.

3. 이해성(Understandable)

- (1) 처음 접하는 사람도 이해하기 쉽도록 구성하고 설명한다.
- (2) 이해하지 못하는 추상적인 표현이나 이해하기 어려운 용어는 사용하지 않는다.

4. 가독성(Readable)

- (1) 문서를 쉽게 읽을 수 있어야 한다(문서 템플릿과 타이포그래피).

- (2) 표준화된 템플릿을 작성하여 적용한다(회사의 고유한 문서 양식).
- (3) 버전의 넘버링은 v1.0, v2.0 등과 같이 일관성 있게 한다. 문서의 인덱스에 대한 규칙 적용, 목차 제공이 중요하다.
- (4) 줄의 간격은 충분하게 유지하며, 단락에 대한 구분과 들여쓰기의 기준을 마련하여 읽기에 쉽고 편해야 한다.
- (5) 여백과 빈 페이지는 적절하게 활용하여 여백의 미를 살리도록 한다.
- (6) 시각적인 효과를 위한 하이라이팅은 일관성 있게 활용하도록 한다.
- (7) 편집기의 상호 참조(Cross-referencing) 기능을 활용한다(하이퍼링크 등).

5. 수정 용이성(Modifiable)

- (1) 쉽게 변경이 가능해야 한다.
- (2) 수정 또는 개선 사항을 시나리오에 반영함에 있어 쉽게 적용할 수 있어야 한다.
- (3) 동일한 수정 사항을 위해 여러 문서를 편집하지 않도록 한다.

6. 추적 용이성(Traceable)

- (1) 쉽게 추적이 가능해야 한다.
- (2) 변경 사항들이 언제, 어디서, 어떤 부분들이, 왜 발생하였는지 추적이 쉬워야 한다.

7. 모범적인 UI 시나리오 문서의 효과

- (1) 요구사항 오류가 감소한다.
- (2) 의사소통 오류가 감소한다.
- (3) 개발 과정에서의 재작업이 감소하고, 혼선이 최소화된다.
- (4) 불필요한 기능을 최소화한다.
- (5) 시나리오 작성과 소프트웨어 개발 비용을 절감한다.
- (6) 개발 속도를 향상시킨다.
- (7) 유관 부서 만족도를 제고한다.

수행 내용 / UI 상세설계하기

재료 · 자료

- UI 설계서 작성 사례, 메뉴 구조 설계 사례, 내·외부 폼 설계 사례

기기(장비 · 공구)

- 컴퓨터
- 파워포인트(PowerPoint)
- 메모장 등의 HTML 편집 도구

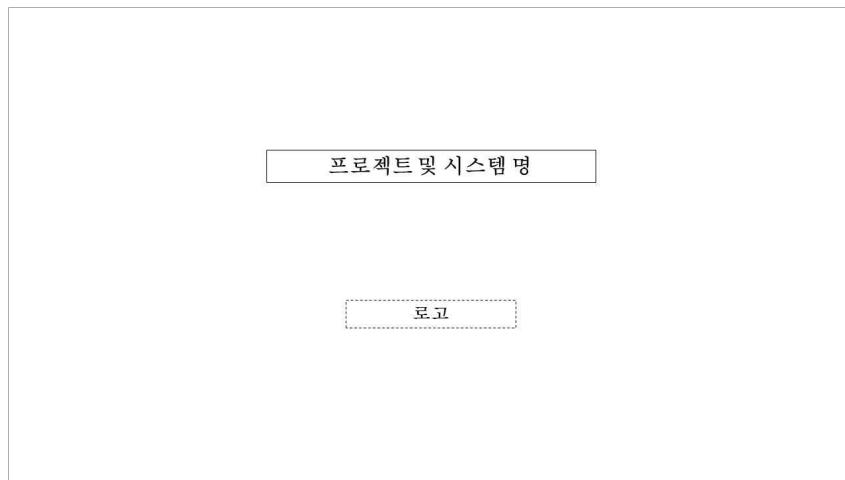
안전 · 유의 사항

- UI 전체 설계서 작성 시 사전에 시나리오에 대해 충분히 숙지한다.
- UI 시나리오는 누구나 읽기 편하고 이해가 쉬워야 한다.
- 메뉴 구조를 설계할 때 UI 설계 원칙이 모두 반영되도록 고려한다.
- UI 사용성 평가와 함께 개선 사항에 대한 반영을 고려한다.

수행 순서

① UI 요구사항과 UI 표준 및 지침에 따라, 사용자의 편의성을 고려한 메뉴 구조를 설계한다.

1. UI 상세설계를 위한 요구사항을 최종 확인한다.
 - (1) UI 설계는 다수의 페이지를 대상으로 다양한 구조 및 디자인에 대한 고민이 필요하기에 요구사항을 다시 한번 살펴보고 검증 후 진행한다.
 - (가) 수집된 요구사항을 바탕으로 기능 및 제약조건을 확인한다.
 - (나) 구조 및 디자인은 사용자의 목적에 맞게 동선의 편리함과 기능을 위주로 철저히 준비한다.
2. UI 설계서 표지 및 개정 이력을 작성한다.
 - (1) UI 설계서 표지를 작성한다.
 - (가) UI 설계서에 포함될 프로젝트 명 또는 시스템 명을 재확인한다.
 - 1) 타 문서와 혼동되지 않도록 프로젝트 명 또는 시스템 명은 동일하게 사용한다.



[그림 2-9] UI 설계서: 표지 예시

(2) UI 설계서 개정 이력을 작성한다.

- (가) UI 설계서를 처음 작성 시에는 첫 번째 항목으로 ‘초안 작성’을 포함시키고, 그에 해당되는 초기 버전(Version)을 1.0으로 설정한다.
- (나) 이후 여러 회의를 거치며 회의 내용을 반영하기 위해 변경 및 보완을 할 경우 각 항목을 추가하고 버전을 0.1씩 더한다.
- (다) 변경 또는 보완이 충분히 이루어져 완성이 되었다고 판단할 경우 버전을 x.0으로 바꾸어 설정한다.

<표 2-2> UI 설계서 : 개정이력사례

No	내용	Version	수정일	작성자
1	초안 작성	V1.0	2007-09-03	홍길동
2	보완	V1.1	2007-09-09	김유신
3	2007-09-10 회의 내용 반영	V1.2	2007-09-11	홍길동
4	2007-09-15 회의 내용 반영	V1.3	2007-09-17	홍길동
5	2007-09-21 회의 내용 반영	V1.4	2007-09-22	김유신
6	2007-09-29 회의 내용 반영	V1.5	2007-09-30	홍길동

3. UI 구조를 설계한다.

(1) UI 요구사항들과 UI 프로토타입에 기초하여 UI 구조를 설계한다.

- (가) UI 요구사항들을 재확인한다.

1) 필요 시 UI 요구사항들을 다시 정리하고 확정/미정 여부를 표시한다.

<표 2-3> 요구사항 정의(RFP) 사례

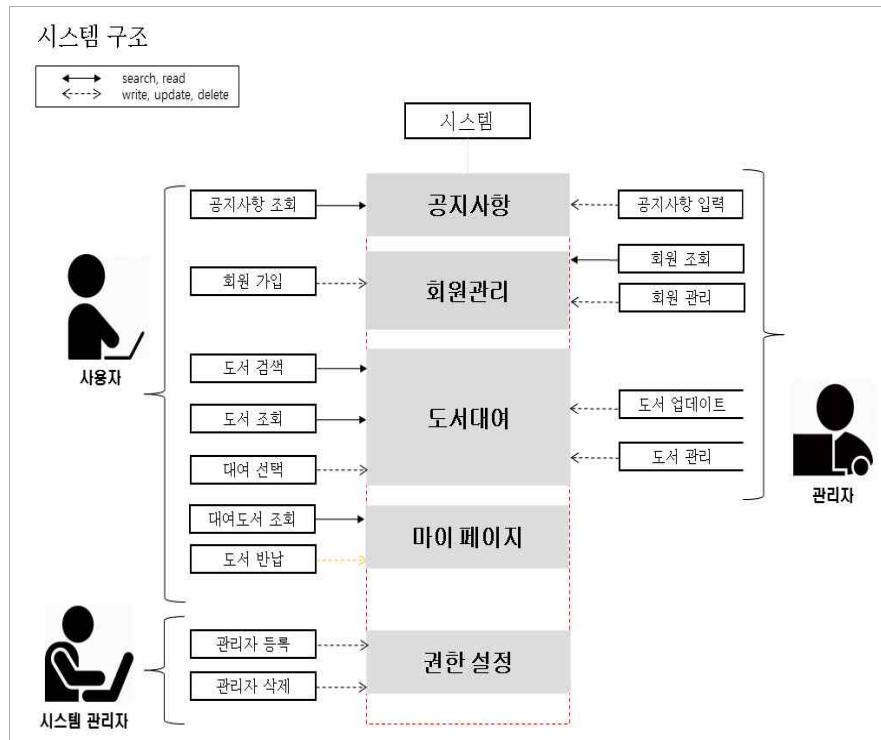
No	RFP	확정 여부	비고
1	요구사항1 - 화면에 표현되어야 할 기능	확정	화면 설계 적용
2	요구사항2 - 화면 입력 요소	확정	화면 설계 적용
3	요구사항3 - 추가적으로 필요 한 화면 요소	확정	화면 설계 적용(1안)
4	요구사항4 - 가능을 표현하기 위한 페이지	확정	화면 설계 적용
5	요구사항5 - 각 화면 간 이동과 흐름	확정	2017-09-15 회의 결과 반영
6	요구사항6 - 공고 및 이벤트 협업 창	확정	2017-09-30 협의 후 결정

(나) UI 프로토타입을 재확인한다.



[그림 2-10] UI 디지털 프로토타이핑: 도서 검색 화면 예시

(다) UI 요구사항들과 UI 프로토타입에 기초해 UI 시스템 구조를 설계한다.



[그림 2-11] UI 설계서: UI 시스템 구조 예시

4. 사용자 기반 메뉴 구조를 설계한다.

(1) 사이트 맵(Site Map) 구조를 통해 사용자 기반 메뉴 구조를 설계한다.

(가) UI 시스템 구조를 바탕으로 사이트 맵 구조를 설계한다.

1) UI 시스템 구조의 내용을 사이트 맵의 형태로 작성한다.



[그림 2-12] UI 설계서: 사이트 맵 구조 예시

2) 사이트 맵 상세 내용을 표 형태로 작성한다.

<표 2-4> 사이트 맵 상세(Site Map Detail) 사례

메뉴 1	메뉴 2	Description
공지사항		공지
도서	도서검색	제목/저자/출판사/내용
	도서조회(항목별)	유아도서
		어린이도서
		소설
		에세이
		경제
		외국어
		사전
마이 페이지		대여 도서 조회
		도서 반납

(나) 사용자 관점에서 요구되는 프로세스들을 진행되는 순서에 맞추어 정리한다.



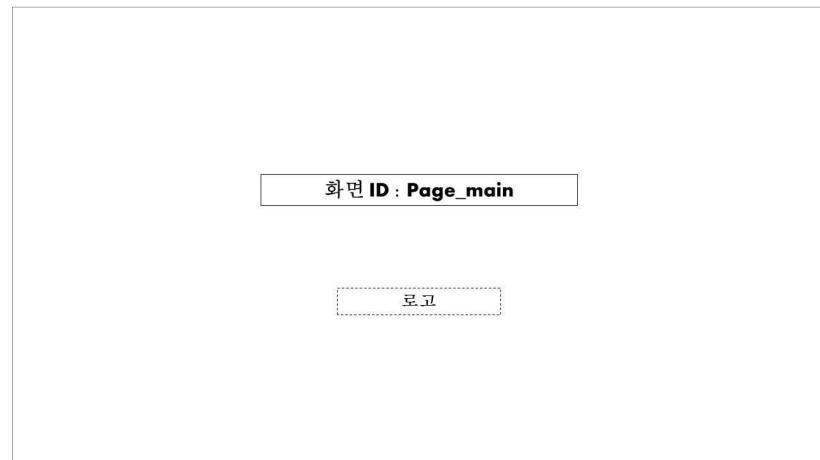
[그림 2-13] UI 설계서: 프로세스 정의 예시

5. 화면을 설계한다.

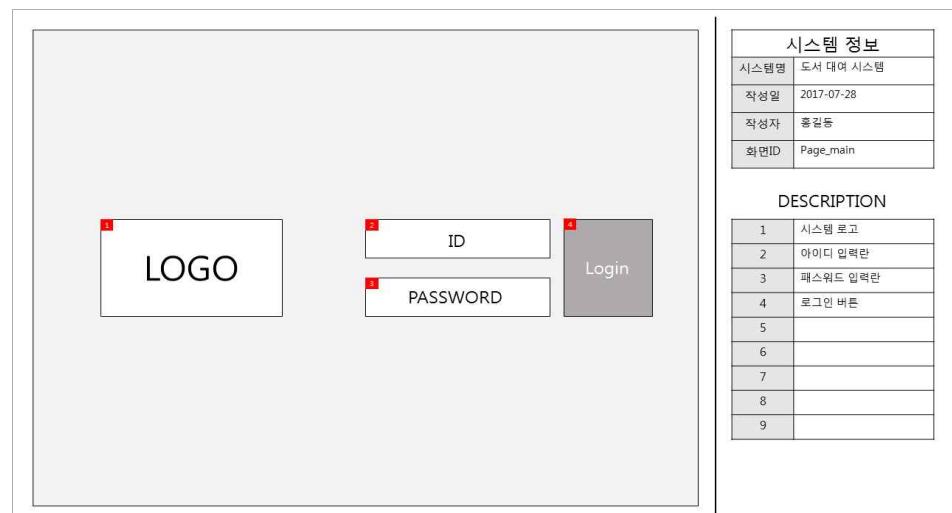
(1) UI 프로토타입과 UI 프로세스 정의를 참고해 화면을 설계한다.

(가) UI 프로토타입과 UI 프로세스 정의를 참고해 각 페이지별로 필요한 화면을 설계 한다.

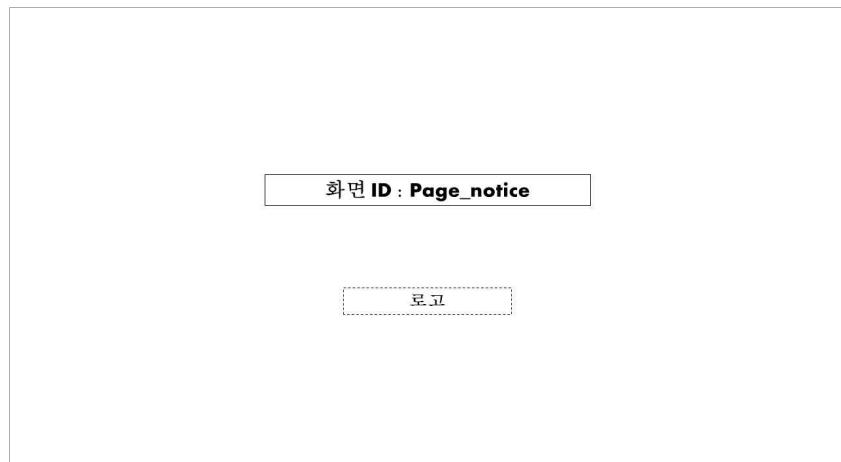
- 1) 필요한 화면 수를 산출한다.
- 2) 각 화면별로 구분되도록 각 화면별 고유 ID를 부여하고 별도의 표지 페이지를 작성한다.
- 3) 각 화면별로 필요한 화면 내용을 설계한다.



[그림 2-14] UI 설계서: 로그인 페이지 타이틀 예시



[그림 2-15] UI 설계서: 로그인 페이지 화면 설계 예시

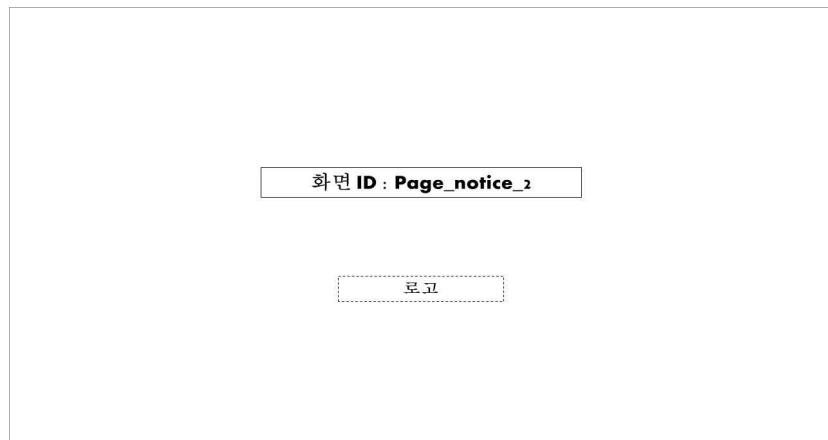


[그림 2-16] UI 설계서: 공지사항 페이지 타이틀 예시

시스템 정보	
시스템명	도서 대여 시스템
작성일	2017-07-28
작성자	홍길동
화면ID	Page_notice

DESCRIPTION	
1	시스템 로고
2	로그아웃 버튼 Page Id = Page_main
3	검색어 입력란
4	검색 버튼
5	Page Id = Page_book
6	공지목록 Page Id = Page_notice_2
7	다음페이지 이동버튼
8	
9	

[그림 2-17] UI 설계서: 공지사항 페이지 화면 설계 예시



[그림 2-18] UI 설계서: 공지사항 조회 페이지 타이틀 예시

LOGO

Logout

검색

제목	도서 대여 관련 공지		
날짜	2017-07-28	작성자	조자룡
내용			

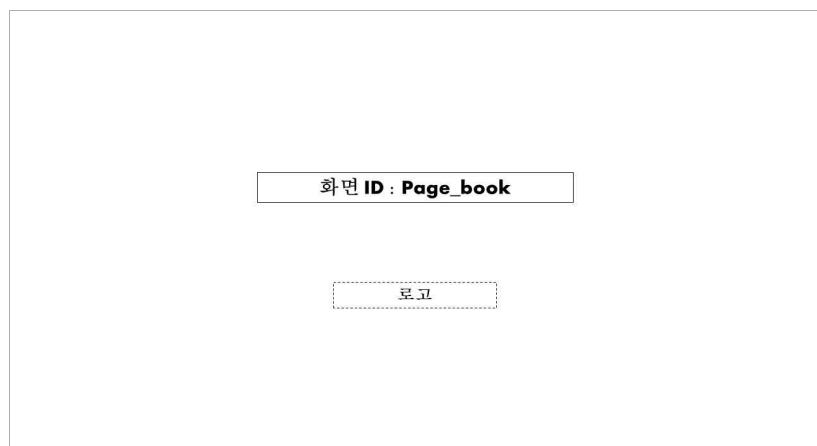
시스템 정보

시스템명	도서 대여 시스템
작성일	2017-07-28
작성자	홍길동
화면ID	Page_notice_2

DESCRIPTION

1	공지사항 세부내용
2	
3	
4	
5	
6	
7	
8	
9	

[그림 2-19] UI 설계서: 공지사항 조회 페이지 화면 설계 예시



[그림 2-20] UI 설계서: 도서목록 페이지 태이틀 예시

LOGO

Logout

검색

제목	도서명 / 색인정보	대출상태
1	도서명 / 색인정보	대출중
2	도서명 / 색인정보	대출가능
3	도서명 / 색인정보	대출가능
4	도서명 / 색인정보	대출중
5	도서명 / 색인정보	대출가능
6	도서명 / 색인정보	대출가능
7	도서명 / 색인정보	대출가능
8	도서명 / 색인정보	대출가능
9	도서명 / 색인정보	대출가능
10	도서명 / 색인정보	대출가능

1 2 3 4

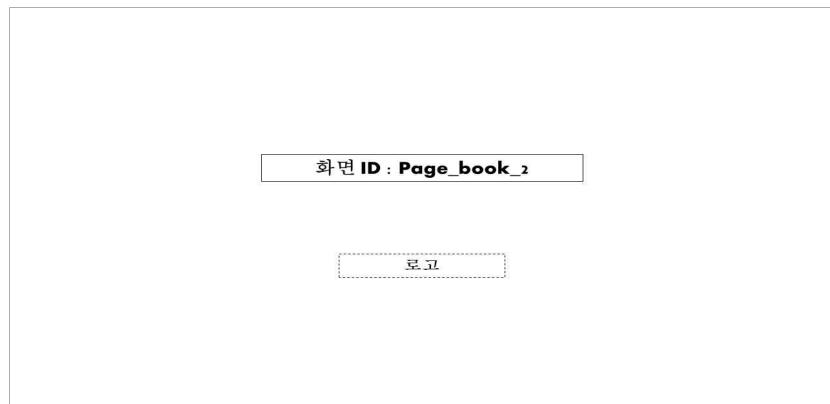
시스템 정보

시스템명	도서 대여 시스템
작성일	2017-07-28
작성자	홍길동
화면ID	Page_book

DESCRIPTION

1	도서 카테고리 목록
2	도서 정보 Page id = Page_book_2
3	대출상태
4	
5	
6	
7	
8	
9	

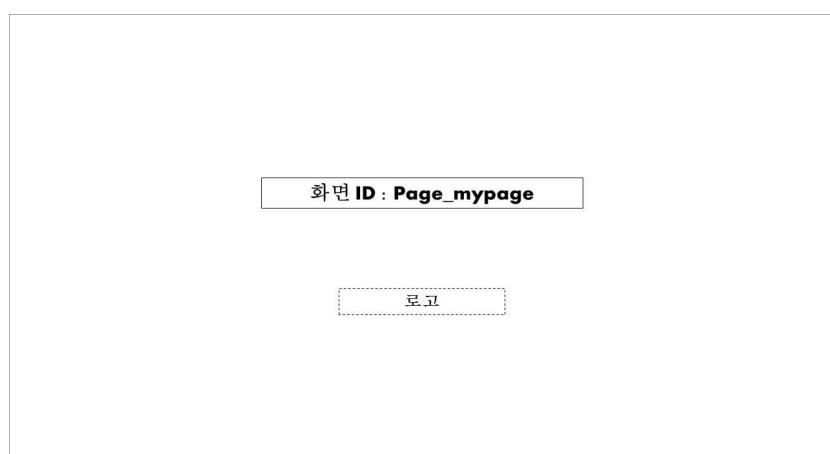
[그림 2-21] UI 설계서: 도서목록 페이지 화면 설계 예시



[그림 2-22] UI 설계서: 도서대여 페이지 타이틀 예시

This screenshot displays a detailed UI design for a book borrowing page. On the left is a sidebar with a "LOGO" icon at the top, followed by three buttons: "공지사항" (Notice), "도서검색" (Book Search) which is highlighted with a red border, and "마이페이지" (My Page). The main content area features a search bar with a magnifying glass icon and a "검색" (Search) button. Below the search bar is a table with two rows. The first row contains the column header "제목" (Title) and the value "뽀롱뽀롱 뽀로로 남극탐험". The second row contains the columns "출판사" (Publisher), "저자" (Author), and "김문학". To the right of the table are two large rectangular boxes: "도서이미지" (Book Image) and "설명" (Description). At the bottom of the main content area is a "대여" (Borrow) button. To the right of the main content area are two vertical panels: "시스템 정보" (System Information) and "DESCRIPTION". The "시스템 정보" panel lists system details: System Name (도서 대여 시스템), Creation Date (2017-07-28), Creator (홍길동), and Screen ID (Page_book_2). The "DESCRIPTION" panel lists numbered items from 1 to 9, each corresponding to a specific element on the page.

[그림 2-23] UI 설계서: 도서대여 페이지 화면 설계 예시



[그림 2-24] UI 설계서: 대여도서 조회 페이지 타이틀 예시

The screenshot shows a user interface for a library system. On the left, there's a sidebar with 'LOGO' at the top, followed by three buttons: '공지사항', '도서검색', and '마이페이지' (My Page). The main area has a search bar with a magnifying glass icon and a '검색' button. Below the search bar is a table with the following data:

1	이름	김첨지		
대여 권 수	4	가입일	2017-06-15	
1	도서명 / 색인정보 / 반납일자	3	반납	
2	도서명 / 색인정보 / 반납일자	3	반납	
3	도서명 / 색인정보 / 반납일자	3	반납	
4	도서명 / 색인정보 / 반납일자	3	반납	

To the right of the main content, there are two tables:

시스템 정보	
시스템명	도서 대여 시스템
작성일	2017-07-28
작성자	홍길동
화면ID	Page_mypage

DESCRIPTION	
1	회원 정보
2	대여 도서 정보
3	반납 버튼
4	
5	
6	
7	
8	
9	

[그림 2-25] UI 설계서: 대여도서 조회 페이지 화면 설계 예시

② UI 요구사항과 UI 표준 및 지침에 따라, 하위 시스템 단위의 내·외부 화면과 폼을 설계한다.

1. 실행 차를 줄이기 위한 UI 설계 원리를 검토한다.

(1) 사용 의도를 파악한다.

(가) 구축할 시스템의 UI 설계서를 통해 각 화면 설계 결과에서 불필요한 부가 기능이 있는지 조사한다.

(나) 구축할 시스템의 UI 설계서를 통해 각 화면 설계 결과에서 중복되는 기능이 있는지 조사한다.

(2) 행위 순서를 검토한다.

(가) 구축할 시스템의 UI 설계서를 통해 사용자가 시스템의 각 기능을 사용하기 위해 어떤 사전 행위들이 수행되어야 하는지 나열하고 조사한다.

(나) 구축할 시스템의 UI 설계서를 통해 각 기능들의 상대적 중요도를 상, 중, 하로 나눈 후 적절성을 비교한다.

<표 2-5> 프로토타입 제작 구분 예시 표

번호	요구기능	상대적 중요도
1	사용자가 계정 아이디와 패스워드를 입력할 수 있어야 한다.	상
2	시스템이 도서 탐색 화면을 제공해 주어야 한다.	상
3	시스템이 책의 정보를 제공해 주어야 한다.	중
4	사용자가 책 이름을 입력할 수 있어야 한다.	상
5	시스템은 일치하는 책을 검색하여 사용자가 선택이 가능한 리스트를 제공해 주어야 한다.	상
6	사용자가 책을 선택할 수 있어야 한다.	상
7	시스템이 예약 여부를 제공해 줄 수 있어야 한다.	중
8	예약이 이루어진 경우 예약내역 또는 대출내역이 책에 반영되어야 한다.	중
9	관리자가 사용자들에게 공지사항을 전달할 수 있어야 한다.	하

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

(라) 전반적으로 기능들을 사용하기 위해 필요한 단계 수가 저마다 적절한지 분석한다.

(3) 행위의 순서대로 실행을 검토한다.

(가) 각 기능을 사용하기 위해 필요한 사전 행위들을 바탕으로 구축할 시스템의 UI 설계서 화면 설계 결과에서 편집 창(Editor)이 활용되는 경우, 커서와 포인터 중 무엇이 먼저 활성화되어야 하는지 파악한다.

(나) 각 기능을 사용하기 위해 필요한 사전 행위들을 바탕으로 구축할 시스템의 UI 설계서 화면 설계 결과에서 여러 개의 편집 창(Editor)이 활용되는 경우 커서의 올바른 시작 위치를 파악한다.

(다) 각 기능을 사용하기 위해 필요한 사전 행위들을 바탕으로 구축할 시스템의 UI 설계서 화면 설계 결과에서 메뉴(Menu)가 활용되는 경우 초기 값으로 무엇이 먼저 활성화되어야 하는지 조사한다.

(라) 그 외 초기 값 설정이 필요한 다른 요소들이 없는지 조사한다.

2. 평가 차를 줄이기 위한 UI 설계 원리를 검토한다.

(1) 수행한 키 조작의 결과를 사용자가 빠르게 지각이 가능한지 측정한다.

(가) 구축할 시스템의 UI 설계서 각 화면 설계 결과에서 각 메뉴/아이콘/윈도우가 활성화될 경우 적절한 포인터의 형태를 해야려 결정한다.

- (나) 구축할 시스템의 UI 설계서 각 화면 설계 결과에서 각 기능이 수행되는 동안 적절한 포인터의 형태를 해아려 결정한다.
- (2) 키 조작으로 변화된 시스템의 상태를 사용자가 쉽게 인지 가능한지 검토한다.
- (가) 구축할 시스템에서 다수의 사용자가 동시 접속함에 따라 하나의 처리 업무를 여러 사용자가 동시 접근할 경우 처리 방법을 정리한다.
- (3) 사용자가 가진 원래 의도와 시스템 결과 간의 유사 정도를 사용자가 쉽게 파악 가능한지 검토한다.
- (가) 구축할 시스템에서 이미 처리 중인 업무를 사용자가 다시 처리하고자 할 경우 시스템은 결과를 어떻게 알려주어야 하는지 파악한다.

③ UI 검토(Iteration)를 수행하고 보완한다.

정보통신산업진흥원 부설 SW공학센터의 “소프트웨어 개발 UI/UX 참조모델 가이드”(2014)에 따라 다음과 같이 수행한다.

1. UI 검토를 수행한다.

- (1) UI 상세설계를 위한 단계에서 사용성의 반복적인 검토를 통합으로써 완성도가 높은 UI 상세설계 수행이 가능하므로 UI 검토를 2-3번 확인한다.
- (2) UI 스토리보드 활용을 통해 페이퍼 프로토타입의 평가는 짧은 단위로 개발 및 평가를 반복하여 확인한다.

2. UI 검토를 보완한다.

- (1) 대표 화면이나 인터페이스 위젯(아이콘, 팝업, 체크 박스 등), 물리적인 버튼을 사용하여 인터랙션에 대한 흐름이 눈에 보이도록 스토리보드를 제작한다.
- (2) 사용자가 페이퍼 프로토타입을 실제 상황처럼 시뮬레이션하여 사용하면, 컴퓨터의 역할을 하는 사람은 인터랙션을 시뮬레이션하여 시연한다.
- (가) 컴퓨터 역할을 하기 위해 페이퍼를 조작하는 사람 1~2명 : 매우 중요한 역할을 수행하므로 가능한 한 인터페이스에 대한 이해도가 가장 높은 사람으로 추천한다.
- (나) 전체적인 평가를 위한 평가 진행자(Facilitator) 1명 : 평가 진행자가 사용자에게 수행할 과업(Task)을 제시하고 질문 사항에 응답을 받는 실험을 진행한다.
- (다) 관찰자(Observer) 1명 이상 : 사용자가 수행하는 말과 행동을 관찰하고, 사용자가 실제 체험하는 어려움이나 설계의 문제점을 관찰 후 기록한다.
- (3) 위의 사용자 평가 결과를 토대로 설계를 보완한다.

3. UI 시연을 통한 사용성에 대한 검토 및 검증을 수행한다.

소프트웨어의 개발이 시작되면, 각 스크린의 레이아웃과 인터랙션의 대부분이 적용된 고 수준(High-Fidelity)의 프로토타입을 활용하여 내부 개발자 및 전문가의 평가를 거침으로써 개선 사항을 지속적으로 반영하려는 목적으로 UI 사용성 평가를 작성한다.

수행 tip

- UI 설계는 2-3인을 한 개의 팀으로 구성해 실습을 진행한다.
- 전체 구성원의 수를 고려하여 팀을 구성한다.
- 구성원 개인별 능력을 고려하여 팀을 구성한다.
- 검토 대상자 선정은 전문가를 활용한다.
- 사용성 검토를 통해 여러 의견을 수집 및 분석하고 개발에 반영한다.
- 설계는 언제든지 변경 가능 하다는 것을 숙지한다.
- 미리 몇 가지 검토 안을 만들어 두는 것이 편리하다.

학습2 교수 · 학습 방법

교수 방법

- 사전에 UI 요구사항 및 UI 프로토타입이 적절히 준비되어 있는지 파악한 후 수업을 진행 한다.
- 유용성에 관한 설계 원리 관점에서 기준 UI의 좋은 예시와 나쁜 예시에 대해 충분히 설명 한 후 수업을 진행한다.
- 다양한 UI 설계서 예시에 대한 PPT 자료와 함께 설명한다.
- UI 설계서 작성 과정에 대해 단계적 실습이 이루어질 수 있도록 지도한다.

학습 방법

- 다양한 UI 요구사항과 프로토타입을 사전에 먼저 숙지한다.
- UI 설계서의 사용 예시를 통해 목적과 활용에 대해 숙지한다.
- UI 요구사항들을 반영한 UI 설계 작성 과정에 대해 실습한다.
- 여러 예제에 대해 UI 설계서 작성을 연습해 본다.

학습2 평 가

평가 준거

- 평가자는 학습자가 학습 목표를 성공적으로 달성하였는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.

학습 내용	학습 목표	성취수준		
		상	중	하
UI 흐름설계	- UI 요구사항과 UI 표준 및 지침에 따라, 화면과 품의 흐름을 설계하고, 제약사항을 화면과 품 흐름설계에 반영할 수 있다.			
UI 상세설계	- UI 요구사항과 UI 표준 및 지침에 따라, 사용자의 편의성을 고려한 메뉴 구조를 설계할 수 있다. - UI 요구사항과 UI 표준 및 지침에 따라, 하위 시스템 단위의 내·외부 화면과 품을 설계할 수 있다.			

평가 방법

- 평가자 체크리스트

학습 내용	학습 목표	성취수준		
		상	중	하
UI 흐름설계	- 체계적인 UI 설계서 작성 능력			
UI 상세설계	- 사용자의 편의성을 고려한 메뉴 구조 설계 능력 - 하위 시스템 단위의 내·외부 화면과 품 설계 능력			

- 포트폴리오

학습 내용	학습 목표	성취수준		
		상	중	하
UI 흐름설계	- 체계적인 UI 설계서 작성 능력			
UI 상세설계	- 사용자의 편의성을 고려한 메뉴 구조 설계 능력 - 하위 시스템 단위의 내·외부 화면과 폼 설계 능력			

피드백

1. 평가자 체크리스트

- 결과물을 제출받아 UI 설계서 작성에 대한 이해 정도에 대해 평가한 후에 필요한 사항 등을 정리하여 돌려준다.
- UI 설계서 내 각 화면별 사용자의 편의성을 고려한 상세설계 수준에 대해 평가한 후에 필요한 사항 등을 정리하여 돌려준다.
- UI 설계안 적정성의 하위 시스템 단위의 내·외부 화면과 폼 설계 능력 검토 결과에 대해 평가한 후에 필요한 사항 등을 정리하여 돌려준다.

2. 포트폴리오

- 실습한 UI 설계서 작성 결과에 대해 평가한 후에 주요 사항에 대하여 표시하여 돌려준다.
- 실습한 사용자의 편의성을 고려한 메뉴 구조 설계 능력 결과에 대해 평가한 후에 주요 사항에 대하여 표시하여 돌려준다.
- 실습한 하위 시스템 단위의 내·외부 화면과 폼 설계 능력 결과에 대해 평가한 후에 주요 사항에 대하여 표시하여 돌려준다.

참고자료



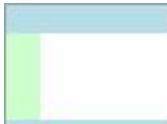
- 권도형(2004). 「실시간 소프트웨어 아키텍처의 성능 중심 평가 방법」. 석사학위논문, 숭실대학교 대학원 컴퓨터학과.
- 김영훈(2012). 「비즈니스 개발 환경에서의 UX(User Experience) 추진 전략」. 석사학위논문, 단국대학교 정보미디어대학원 IT학과.
- 김치수(2015). 「쉽게 배우는 소프트웨어 공학」. 한빛아카데미.
- 미래창조과학부, 정보통신산업진흥원 부설 SW공학센터, 한국디자인진흥원(2014). 소프트웨어 개발 UI/UX 참조모델 가이드.

활용서식

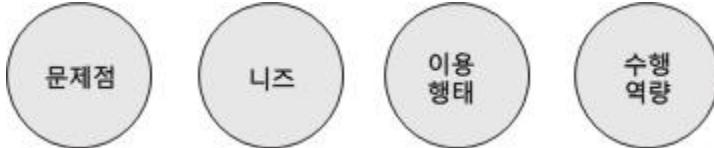


작업 포트폴리오

UI 시나리오 스토리보드 (UI Scenario StoryBoard)

화면코드		페이지명	
화면경로			
작성자		작성일	
VER		P.NO	
 - 영역			
Description			

인터뷰 리포트 (Interview Report)

인터뷰 리포트		YY/MM/DD 수행시간 장소
이름/부서/직급/역할/근무처 (인구통계학적인 데이터) (근속년수/해당업무담당기간)	업무/사용 메뉴/업무 프로세스 이용동기 1 이용동기 2 이용동기 3	
시스템에 대한 선호도 / 태도 / 이용환경 / 수행역량		
질의내용/답변		참고자료/인터뷰시 기록자료
		
주요 이슈	진행자 의견 / 보완사항	

요구사항 명세서 양식

요구사항번호	요구사항 명
	①
	②
	③
	④
	⑤
요구사항 내용	⑥
관련요구사항	
필요조건	우선순위

유스케이스 설명 양식

유스케이스 명	
필요조건	
시나리오 상세	
[주요 이벤트 흐름]	
[예외 흐름]	
처리후조건	

NCS학습모듈 개발이력

발행일	2015년 12월 31일	
세분류명	응용SW엔지니어링(20010202)	
개발기관	한국소프트웨어기술진흥협회, 한국직업능력개발원	
집필진	강석진(이비스톰)* 김보운(이화여자대학교) 김홍진(LG CNS) 유은희 장현섭((주)커리텍) 주선태(T3Q) 진권기(이비스톰) 최재준	검토진 김승현(경희대학교) 엄기영(우리에프아이에스) 장온순(한국IT컨설팅) 조상욱(세종대학교) 조성호(삼성카드)
		* 표시는 대표집필자임
발행일	2017년 12월 31일	
세분류명	응용SW엔지니어링(20010202)	
개발기관	(사)한국정보통신기술사협회, 한국직업능력개발원	
집필진	박미화(동국대학교)* 김승환((주)캐롯아이) 김원기(LG CNS) 김종명(SM신용정보) 박현기(프리랜서) 유현주((사)한국정보통신기술사협회) 이구성(한국아이씨티(주)) 이숙희(서초문화예술정보학교) 최창선(한빛디엔에스(주)) 홍민표(한화S&C)	검토진 권순명(주씨에이에스) 김태형((사)한국정보통신기술사협회) 양승화(라이나생명보험) 이성화(시스원) 황극인(주)코스콤
		* 표시는 대표집필자임
발행일	2018년 12월 31일	
학습모듈명	화면 설계(LM2001020224_16v4)	
개발기관	한국직업능력개발원	

화면 설계(LM2001020224_16v4)

저작권자	교육부
연구기관	한국직업능력개발원
발행일	2018. 12. 31.

* 이 학습모듈은 자격기본법 시행령(제8조 국가직무능력표준의 활용)에 의거하여 개발하였으며, NCS통합포털사이트(<http://www.ncs.go.kr>)에서 다운로드 할 수 있습니다.



www.ncs.go.kr