

2. 화면 설계

화면 설계는 생소한 설계 이론을 이해하는 데 많은 시간이 필요할 수 있습니다. 따라서 모든 부분을 학습하기보다는 핵심 개념에 중점을 두고 UI의 설계원칙, 스타일 가이드, UML, 유스케이스 다이어그램, 프로토타입, UI 상세 설계 등을 학습해봅시다.



UI 유형

- CLI (Command Line Interface): 명령어를 텍스트로
- GUI(Graphic User Interface) : 마우스, 전자펜
- NUI :터치, 음성
- OUI(Organic User Interface) : 현실에 존재하는 모든 사물

UI 설계 원칙 / 직유학유

- 직관성(Intuitiveness) : 누구나 쉽게 이해하고, 쉽게 사용할 수 있어야 함
- 유효성(Efficiency) : 정확하고 완벽하게 사용자의 목표가 달성 될 수 있도록 제작
- 학습성(Learnability) : 초보와 숙련자 모두가 쉽게 배우고 사용할 수 있게 제작
- 유연성(Flexibility) : 사용자의 요구사항을 최대한 수용하고, 실수를 방지할 수 있도록 제작

UI 품질 요구사항 / 기신사효유이

- 기능성(Functionality) : 실제 사용시 정확하지 않은 결과가 발생할 확률과 시스템의 동작 관찰
 - 적절성, 정밀성, 상호 운용성, 보안성, 호환성
- 신뢰성(Reliability): 일정한 시간, 작동되는 시간동안 의도하는 기능을 수행함을 보증
 - 성숙성, 고장 허용성, 회복성
- 사용성(Usability) : 어떠한 행위를 정확하고 쉽게 인지할 수 있는
 - 이해성, 학습성, 운용성
- 효율성(Efficiency) : 할당된 시간에 한정된 자원으로 얼마나 빨리 처리할 수 있는가
 - 시간 효율성, 자원 효율성
- 유지보수성(Maintainability) : 요구사항 개선, 확장에 있어 얼마나 용이한가
 - 분석성, 변경성, 안정성, 시험성
- 이식성(Portability) : 다른 운영체제에서도 얼마나 쉽게 적용이 가능한가
 - 적용성, 설치성, 대체성

UI 개발을 위한 주요 기법

- 3C 분석 : 고객(Customer), 자사(Company), 경쟁사(Competitor) 비교하고 분석
- SWOT 분석 : 기업 내부 환경과 외부환경을 분석해 Strength, Weakness, Opportunity, Threat 요인을 규정하고 이를 토대로 경영 전략 수립
- 시나리오 플래닝 : 상황 변화를 사전에 예측하고 다양한 시나리오를 설계하는 방법
- 사용성 테스트 : 사용자가 직접 제품을 사용하면서 미리 작성된 시나리오에 맞추어 과제를 수행 한 후, 질문에 답하도록 하는 테스트
- 워크숍 : 소집단 인원으로 특정 문제나 과제에 대한 새로운 지식, 기술, 아이디어들을 서로 교환하고 검토하는 세미나

UI 화면 설계

- 스토리보드 : 정책, 프로세스, 와이어 프레임, 기능 정의, 데이터베이스 연동 등 서비스 구축을 위한 정보가 수록된 문서, 디자이너와 개발자가 최종적으로 참고하는 산출 문서
- 와이어 프레임 : 화면 단위의 레이아웃을 설계하는 작업
- 프로토타입 : 정적인 화면(와이어 프레임, 스토리보드)에 동적 효과를 적용해 실제 구현된 것처럼 시뮬레이션 할 수 있는 모형

UML(Unified Modeling Language)

객체지향 소프트웨어 개발 과정에서 산출물을 명세화, 시각화, 문서화 할 때 사용되는 모델링 기술과 방법론을 통합해서 만든 표준화된 범용 모델링 언어

UML구성요소

사물, 관계, 다이어그램

UML 다이어그램

- 구조적(Structural) 다이어그램 / 정적(Static) 다이어그램
 - 클래스(Class) : 클래스 간 정적인 관계를 표현
 - 객체(Object): 클래스에 속한 사물, 인스턴스
 - 컴포넌트(Component) : 컴포넌트와 그들 사이 의존 관계
 - 배치(Deployment) : 컴포넌트 사이의 종속성, 물리적 요소들의 위치
 - 복합체 구조(Composite Structure) : 클래스나 컴포넌트가 복합 구조를 갖는 경우 그 내부 구조를 표현
 - 패키지(Package) : 유스케이스나 클래스 등의 모델 요소들을 그룹화한 패키지들의 관계를 표현
- 행위적(Behavioral) 다이어그램 / 동적(Dynamic) 다이어그램
 - 유스케이스 : 시스템 외부 요소를 사용자의 관점에서 표현
 - 시퀀스 : 시간적 개념을 중심으로 메시지 흐름으로 표현
 - 커뮤니케이션 : 객체들이 주고받는 메시지를 표현하고 객체 간의 연관까지 표현
 - 상태 : 상태가 어떻게 변화하는지 표현
 - 활동 : 어떤 기능을 수행하는지, 처리 로직이나 처리 흐름
 - 타이밍 : 객체 상태 변화와 시간 제약을 명시적으로 표현

UML 확장 모델의 스테레오 타입

'<< >>'(길러멧; Guillemet) 기호를 사용하여 표현

클래스 다이어그램

- 접근제어자
 - public + : 클래스 외부 접근 허용
 - private - : 클래스 내부 접근 허용
 - protected # : 동일 패키지/파생 클래스에서 접근
 - default ~ : 동일 패키지 클래스에서 접근
- 클래스 간의 관계
 - 연관 : 실선, 2개 이상의 사물이 서로 관련되어 있는 상태
 - 집합 : 속이 빈 마름모 (차/엔진, 바퀴, 운전대), 하나의 객체에 여러 개의 독립적인 객체들이 구성
 - 복합(=포함) : 속이 채워진 마름모, 집합 보다 더 강한 관계
 - 일반화 : 부모-자식, 속이 빈 화살표 (차/버스, 택시, 자가용), 상속 관계
 - 의존 : 점선 화살표, 서로 연관은 있으나 필요에 따라 짧은 시간동안만 연관을 유지

UI 시나리오 문서의 작성 요건(완일이가 추수)

완전성, 일관성, 이해성, 가독성, 추적 용이성, 수정 용이성