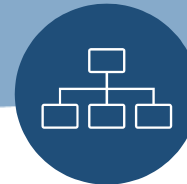
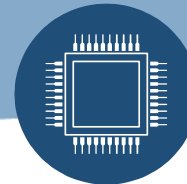


[일머리사관학교 디지털헬스케어]



디지털헬스케어와 블록코딩 앱개발

경남대학교 창의융합대학 교수 유현주
comjoo@kyungnam.ac.kr



강의 진행 안내



- 수업 자료 공유 URL
 - <https://github.com/yoohjoo/>
 - 수업에 필요한 강의 자료 다운로드 저장
- 소통을 위한 메일 주소: comjoo@kyungnam.ac.kr
yoohjoo94@gmail.com





데이터리터러시

- SW개발 프로젝트와 앱 서비스 설계 과정
- 데이터 기반 문제 해결
- 인공지능 기반 앱 설계



SW개발 프로젝트와 앱 서비스 설계 과정



앱 서비스 설계



- 앱 서비스 설계의 이해

- 앱 서비스 설계에서의 디자인은 어떤 목표나 비전을 실현하기 위해 계획을 세우는 과정이며
앱 서비스 설계의 궁극적인 목표는 사용자의 만족을 추구하는 것임
- 단순히 아이디어를 설계하는 것을 넘어서 실제로 실행 가능한 형태로 구체화해야 함
- 앱 서비스 설계에는 다양한 방법이 있음

- 앱 서비스 설계 프로세스

- 설계를 하기에 앞서 무엇부터 시작해야 하고, 어떤 준비가 필요한지 전체적인 흐름을 파악
- 사례 프로세스는 시장 분석, 타겟 설정, 목표 수립, 현장 조사와 아이디어 구체화,
테스트까지의 단계별 과정을 정리한 문제 해결 접근 방식을 말함

앱 서비스 설계



● 앱 서비스 설계 프로세스

- 디자인 씽킹 : 사용자의 행동과 이해를 바탕으로 문제를 재정의하고, 프로토타입 배포와 테스트를 거쳐 혁신적인 해결책을 도출하는 비선형적이고 반복적인 프로세스를 말하며, '창의적 문제 해결 방법론'으로도 불림
 - ✓ 공감 : 사용자 조사를 진행하여 해결하려는 문제에 공감적 이해를 얻는 과정
 - ✓ 정의 : 수집한 정보와 축적된 관찰 내용을 분석하고 종합하여 사용자의 요구사항에서 찾은 핵심 문제를 정의하는 단계
 - ✓ 아이디어 발굴 : 고정관념에서 벗어나 문제 해결 방안을 고안하려고 다양한 가설을 제시하는 단계
 - ✓ 프로토타입 : 고안한 해결 방안을 시각화하고, 구체적인 서비스를 개발하는 단계로 제안된 솔루션의 장점과 단점을 빠르게 식별하는 것을 목표로 함
 - ✓ 테스트 : 프로토타입을 기반으로 테스트 계획을 수립하고 결과를 학습하는 단계

앱 서비스 설계



● 앱 서비스 설계 프로세스

● 더블 다이아몬드 방법론

✓ 문제를 정의하고 해결책을 도출하는 두 가지 과정이 한 점에서 시작해 확산과 수렴이라는 과정으로 구조화한 방법

① 첫 번째 다이아몬드 영역

사용자가 겪는 문제를 찾고, 디자인으로 해결할 문제를 정의하는 단계

② 두 번째 다이아몬드 영역

앞에서 찾은 문제를 해결할 수 있는 다양한 방안을 탐색하고 솔루션을 만들어나가는 단계

✓ 더블 다이아몬드 방법론을 한 번 실행해서 최종 해결책이 도출되는 것은 아님

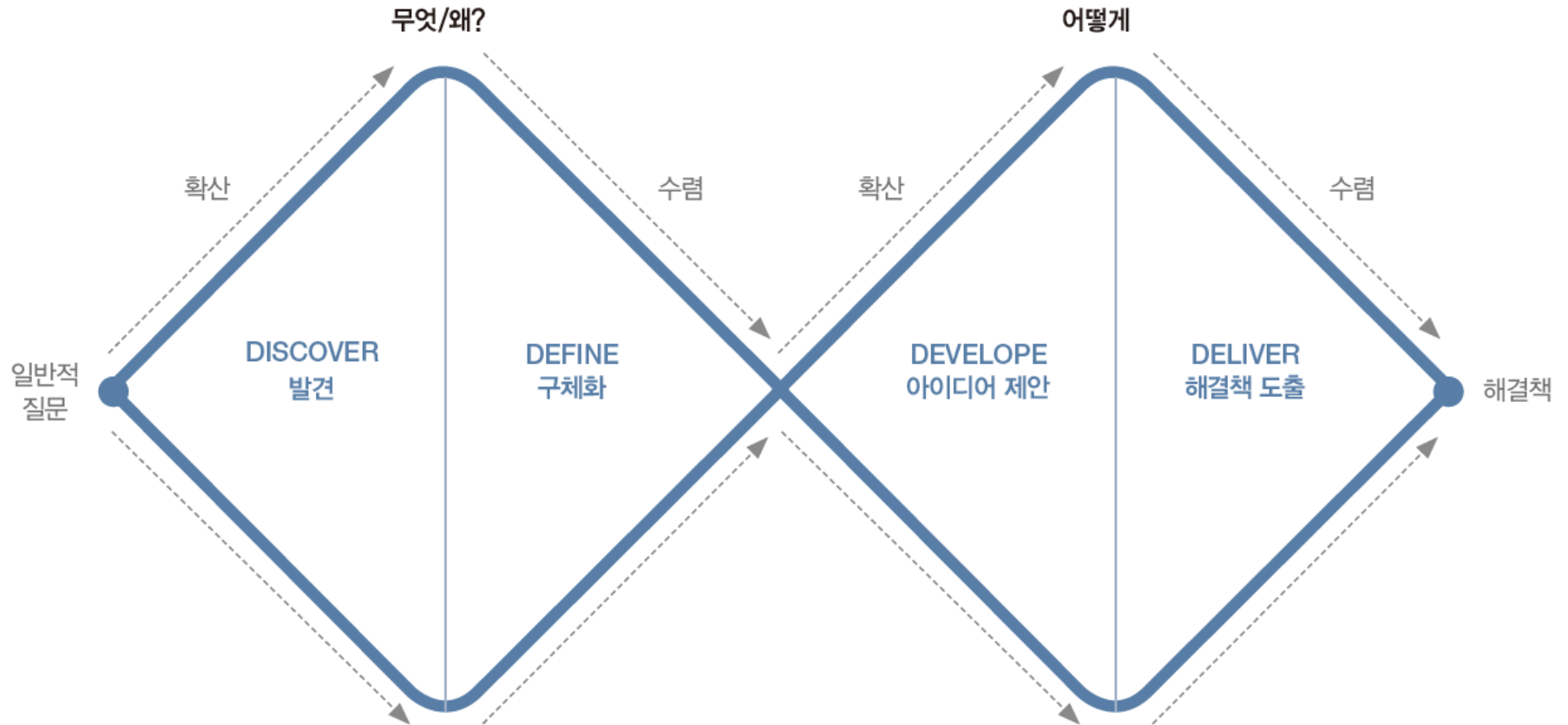
✓ 사용자 의견에서 또 다른 문제를 발견했다면 다시 문제 해결을 위한 크고 작은 확산과 수렴을 반복함

앱 서비스 설계



- 앱 서비스 설계 프로세스

- 더블 다이아몬드 방법론



영국 디자인 카운슬의 더블 다이아몬드 방법론



앱 서비스 설계

● 앱 서비스 설계 프로세스

● 더블 다이아몬드 방법론

더블 다이아몬드 방법론의 단계별 목표·내용·방법론

단계	발견	구체화	아이디어 제안·개발	해결책 도출
목표	<ul style="list-style-type: none">• 주제 설정• 현황 파악• 문제점 수집• 욕구 발견	<ul style="list-style-type: none">• 현황 정의• 문제점 정의• 데이터를 활용한 니즈 도출	<ul style="list-style-type: none">• 아이디어의 구체화• 콘셉트 도출• 비즈니스 모델 개발	<ul style="list-style-type: none">• 서비스 확정• 발표• 상호 공유
내용	<ul style="list-style-type: none">• 사실 발견• 이해관계자 욕구 이해• 생태계 파악• 가치사슬	<ul style="list-style-type: none">• 문제점 재정의• 요구사항 정리• 수집된 데이터 관계 파악• 핵심 아이디어 도출	<ul style="list-style-type: none">• 아이디어 제안• 비즈니스 모델 수립	<ul style="list-style-type: none">• 시각화• 제작• 검증• 평가
방법론	<ul style="list-style-type: none">• 데스크 리서치• 새도잉• 다이어리• 인터뷰	<ul style="list-style-type: none">• 5Whys• 퍼소나• 어피니티 다이어그램	<ul style="list-style-type: none">• 브레인스토밍• 비즈니스 모델• 프로토타이핑	<ul style="list-style-type: none">• 스토리보드• 서비스 블루프린트• 롤 플레이

앱 서비스 설계



● 앱 서비스 설계 프로세스

하나 더 알기

UX 디자인 관점으로 바라보기

- 문제 발견 단계 → 주제에 따라 제품이나 서비스 시장을 분석하고 타깃을 설정
- UX 디자인의 관점은 제품을 사용하는 사람들의 실제 상황에서 문제를 발견하고 사용자의 판단이나 심리에 영향을 주는 요인과 감정적인 부분까지 고려함
- 목표를 설정하면 각 환경에 맞는 기법을 개발해 리서치를 실행하고 그 결과를 모델링함
- 상황과 목표에 따라 프로토타이핑은 여러 가지가 나올 수 있으며 프로토타이핑까지 진행하고 내린 결론으로 테스트를 함

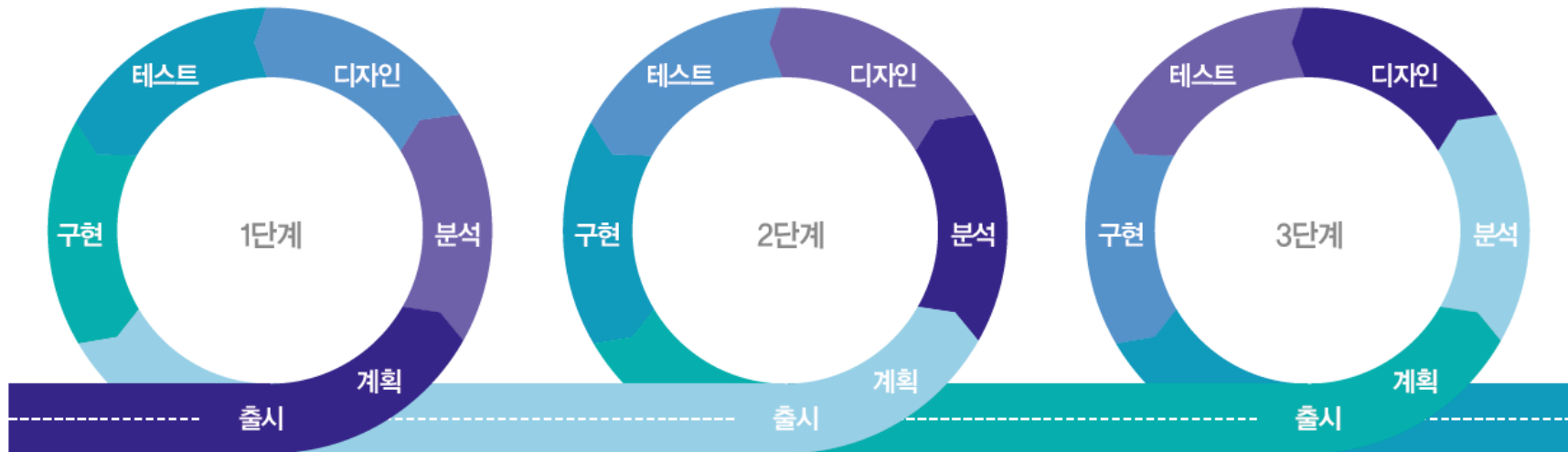
앱 서비스 설계



● 앱 서비스 설계 개발 방법론

● 애자일 프로세스

- 사용자의 요구사항을 끊임없이 반영하고 점진적으로 제품을 업데이트하여 서비스가 추구하는 가치를 구현하는 프로세스



애자일 프로세스

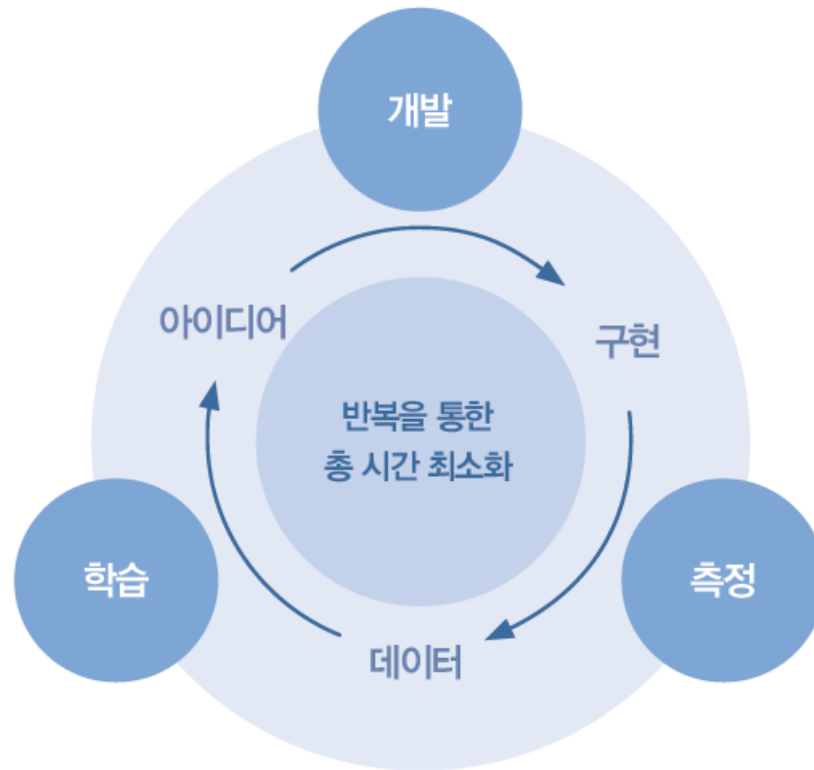
앱 서비스 설계



● 앱 서비스 설계 개발 방법론

● 린 프로세스

- 기업이 가정한 시장 상황을 테스트하기 위해 프로토타입을 만들고, 사용자의 피드백을 받아 프로토타입을 빠르게 발전시키는 프로세스



린 프로세스

SW development project



요구사항 분석(Requirement Analysis)

- 요구사항 명세서 작성

SW 설계(Design)

- 설계 명세서 작성

구현(Development)

- 소스코드, 단위테스트 결과서 등 산출물

테스트(Testing)

- 단위 테스트, 통합 테스트 결과서

이행(Implementation)

- 사용자 매뉴얼 작성

유지보수(Maintenance)

- 오류 발생 시 복구 및 패치, 디자인 업데이트, 유지보수 계획서 등 작성
- 수리 유지보수, 적응 유지보수, 완전지향 유지보수, 예방 유지보수 등



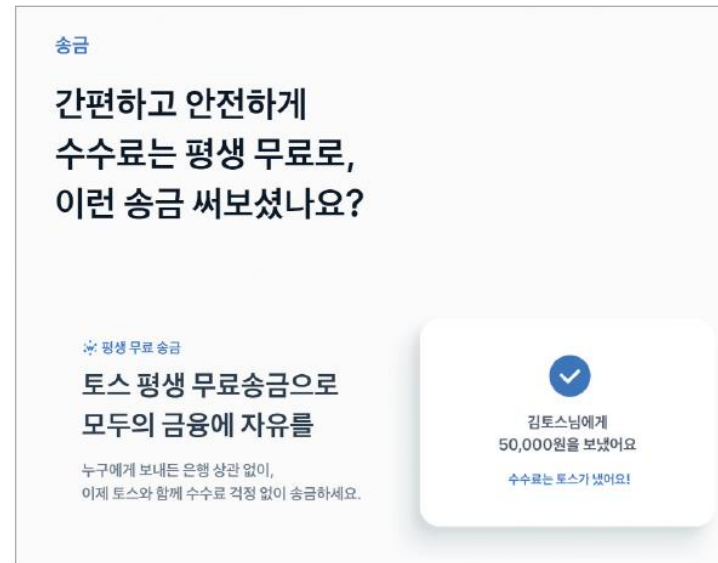
앱 서비스 설계 기획

● 앱 서비스 설계 기획의 이해

- 앱 서비스 설계를 개발하려면 우선 어떤 서비스를 만들 것인지 정해야 함
- 처음부터 개발할 플랫폼의 가치를 높게 설정하고 규모가 커질 가능성을 고려한 주제를 기획하는 것도 의미 있지만 일상에서 발견한 문제에 먼저 접근해 보는 것이 더 좋음



(a) 배달의 민족 'B마트'



(b) 토스의 '송금' 항목

일상 속 문제의식을 서비스로 구현한 기업 사례

앱 서비스 설계 기획



● 실습 I 앱 서비스 설계 기획하기

예제 미리보기

서비스 디자인 기획 실습

Discover

문제점 수집

- 평소에 불편했던 점
- 문제라고 여겼던 점
- 나와 내 주변의 이슈

- 레스토랑 노쇼
- 불면증
- 수면 부족
- 데이트 코스 찾기
- 유명 맛집 줄 서기
- 일회용품 줄이기
- 스트레스 관리
- 다이어트
- 독서 습관
- 반려동물 병원비

주제 선정

- 수집한 문제점에서 해결할 주제 정하기

- 유명 맛집 입장
- 유명 맛집 줄 서기

현황 파악

- 주제와 관련된 정보 리서치
- 문제 발생 원인 찾기

- 방송 출연으로 유명해짐
- 많은 고객
- 오랜 시간 줄 서기
- 날씨 변수
- 대기 시간을 알 수 없음
- 줄 설 장소가 마땅치 않음
- 대규모 인원으로 쓰레기와 소음 발생

Define

가설 설정

- 현황 파악으로 해결하려는 핵심 문제를 정의

사람들에게 인기가 많은 유명 맛집이나 장소에 예약 서비스가 없어 오랜 대기 시간이 발생

프로젝트 정의

- 제품 또는 서비스 소개와 설명 요약

유명 맛집, 핫플레이스의 예약과 원격 줄 서기를 지원하는 앱 서비스

앱 서비스 설계 기획



- 앱 서비스 설계 기획의 이해

- 문제 발견

- ✓ 문제점 수집 :

- 서비스 이용 방식에 따라 실제 사용자에게 영향을 미치는 서비스 경험과 페인 포인트를 파악하여 UX 디자인 관점으로 분석 → 분석 결과로 서비스에서 발견한 사용자 경험 이슈를 정리

- ✓ 주제 선정 :

- 수집한 문제점들에서 공통적인 부분을 찾아내거나 해결할 가치가 높은 순서를 정해 주제를 선정

- ✓ 현황 파악 :

- 현황 파악은 선정한 주제에서 문제가 발생하는 원인을 찾아보는 과정,
모든 문제에 접근을 할 때 'Why'로 시작하여 근본적인 목적을 설정해야 함



앱 서비스 설계 기획

● 앱 서비스 설계 기획의 이해

● 문제 발견

문제 발견 과정의 단계별 내용

문제 발견			
단계	문제점 수집	주제 선정	현황 파악
고려 사항	<ul style="list-style-type: none">• 평소에 불편했던 점• 문제라고 여겼던 점• 나와 내 주변의 이슈	<ul style="list-style-type: none">• 수집한 문제점에서 해결할 주제 정하기	<ul style="list-style-type: none">• 주제와 관련된 정보 리서치• 문제 발생 원인 찾기
예시	<ul style="list-style-type: none">• 레스토랑 노쇼• 불면증• 수면 부족• 데이트 코스 찾기• 유명 맛집 줄 서기• 일회용품 줄이기• 스트레스 관리• 다이어트• 독서 습관• 반려동물 병원비	<ul style="list-style-type: none">• 유명 맛집 입장• 유명 맛집 줄 서기	<ul style="list-style-type: none">• 방송 출연으로 얻은 유명세• 많은 고객• 오랜 시간 줄 서기• 날씨 변수• 대기 시간을 알 수 없음• 줄 설 장소가 마땅치 않음• 대규모 인원으로 쓰레기와 소음 발생



앱 서비스 설계 기획

● 앱 서비스 설계 기획의 이해

● 문제 정의

- 문제 발생의 원인으로 파악한 문제점 중에서 서비스로 구현하여 해결할 문제를 정의
- 가설 설정 : 해결이 필요한 핵심적인 문제를 정의하여 서비스를 제공하는 목적을 설정하는 단계
- 프로젝트 정의 : 서비스를 통해 해결하고자 하는 문제와 제품 또는 서비스에 대한 간략한 요약, 설명을 작성

문제 정의 과정의 단계별 내용

문제 정의		
단계	가설 설정	프로젝트 정의
고려 사항	현황 파악으로 해결하려는 핵심 문제를 정의	제품 또는 서비스 소개와 설명 요약
내용	사람들에게 인기가 많은 유명 맛집이나 장소에 예약 서비스가 없어 오랜 대기 시간이 발생	유명 맛집, 핫플레이스의 예약과 원격 줄 서기를 지원하는 테이블링 앱 서비스

사용자 시나리오



- 사용자 시나리오의 이해

- 사용자 시나리오 : 현재 사용자들의 행동을 묘사하고 페르소나(Persona)가 자신의 목적을 달성하는 방법으로 제품을 어떻게 사용할지 시나리오로 작성하는 것을 말함
- 실제 제품이나 서비스를 사용할 만한 가상의 인물을 설정하면 사용자에게 공통적으로 나타나는 이용 패턴이나 이용 심리를 이해하는 데 도움이 됨
- 따라서 설정한 페르소나를 기준으로 사용자의 감정과 상세한 니즈를 파악하며 사용자 시나리오를 작성

사용자 시나리오



● 사용자 시나리오의 이해

● 사용자 시나리오의 구조

- ✓ 사용자가 경험할 이야기를 미리 그리면서 서비스 전체의 윤곽을 표현하는 것
- ✓ **여정** : 사용자가 서비스를 경험하는 단계를 순서대로 작성하는 과정
- ✓ **목표 또는 문제점** : 사용자가 제품이나 서비스를 사용하여 성취하려는 목표를 작성
- ✓ **행동 단계** : 사용자가 자신의 목표를 이루기 위해 어떤 행동을 할지 자세하게 입력
- ✓ **감정** : 사용자가 특정 단계에서 느끼는 감정이나 생각을 구체적으로 입력하고 그 감정을 그래프로 표시
- ✓ **기회** : 사용자 시나리오의 핵심 부분으로 사용자의 경험을 개선하거나 강화할 수 있는 해결책이 무엇인지 작성

사용자 시나리오



● 사용자 시나리오의 이해

● POINTS 분석

✓ 사용자 시나리오를 작성하고 사용자의 입장에서 불편을 만드는 문제점과 이를 개선할 수 있는 기회 요소는 무엇인지를 분석하는 과정

- **문제**^{Problem} 사용자 시나리오에서 드러난 문제 상황과 사용자가 당면하는 어려움, 구조적 제약 사항을 말한다.
- **기회**^{Opportunity} 디자인으로 접근한 해결 방안 또는 사용자에게 서비스 형태로 제공될 수 있는 내용을 말한다.
- **통찰**^{Insight} 관찰로 발견한 사용자나 디자이너가 미처 몰랐던 새로운 내용을 말한다.
- **필요**^{Needs} 사용자들이 당면한 문제를 해결할 때 필요한 것들을 말한다.
- **테마**^{Theme} 한 문장으로 정리한 핵심 아이디어나 핵심 주제를 말한다.
- **시스템**^{System} 서비스나 제품 자체가 제공하는 것 외에 뒷받침 되는 서비스나 제도를 말한다.




사용자 시나리오

● 페르소나 설정

- 페르소나는 인물의 성격과 성장 배경, 환경 등의 묘사로 구성됨
- 가상의 인물이 해당 프로젝트의 서비스나 제품을 사용하는 목표와 필요 그리고 서비스의 부재로 느꼈던 불편함 등의 내용을 포함하여 작성

● 페르소나 프로필

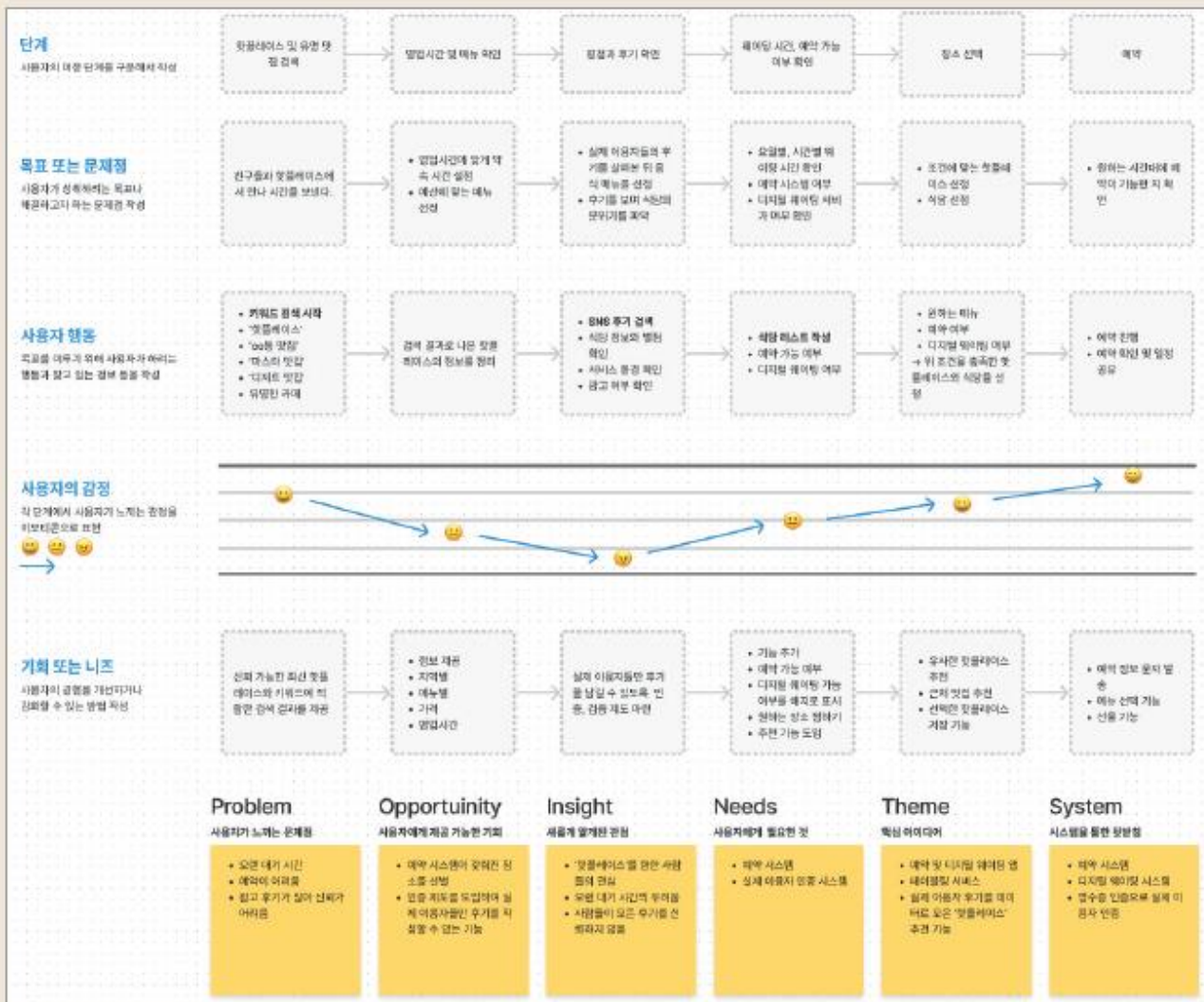
김소미		기본 정보	
		<p>나이 : 27세</p> <p>직업 : 출판 마케터</p> <p>가족 : 미혼</p> <p>성격 : 성실하고 낙천적</p> <p>생활 : 대학을 졸업하고 출판사 마케터로 직장 생활을 한다. 최근 들어 혼자 살기 시작하면서 퇴근 후 다양한 요리를 연습하고 사진을 찍어 SNS에 올리는 것을 취미로 삼고 있다. 자연스레 요리와 음식에 관심이 많아져 다양한 맛집을 방문하는 것을 즐긴다. 주말에는 친구들과 요즘 뜨는 곳(핫플레이스)에 있는 맛집을 약속 장소로 잡는다.</p>	
목표		필요	문제점
핫플레이스에서 대기 시간 없이 친구들과 자유롭게 만나고 싶다.		핫플레이스를 쉽게 발굴하고 대기 시간 없이 이용하고 싶다.	최근 유명 장소와 맛집에 예약 서비스가 없는 곳이 많아 대기 시간이 1시간 이상이다.

사용자 시나리오



● 실습 I 사용자 시나리오 작성하기

예제 미리보기

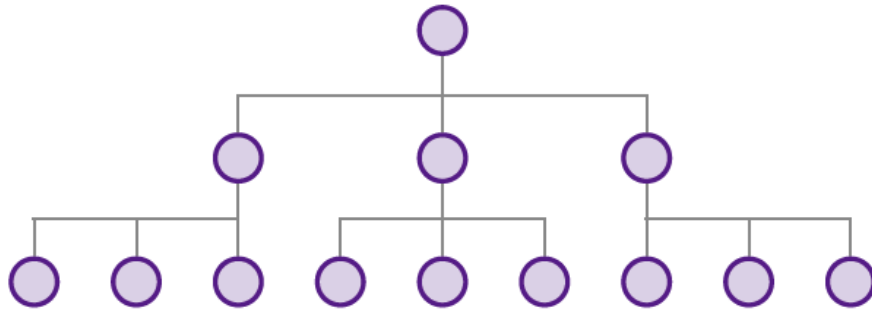


정보 구조 설계



● 정보 구조 설계의 이해

- 정보 구조 : 사용자가 원하는 정보를 불편함 없이 사용할 수 있도록 만든 뼈대
- 정보 구조 설계는 서비스에서 제공하는 정보의 구성을 특정 주제별로 그룹화하고 레이블링하여 콘텐츠 흐름에 맞는 계층 구조를 만드는 것



(a) 계층 구조 (메뉴 트리 구조)



(b) 순차 구조

정보 구조 설계 시스템의 예시

정보 구조 설계



● 실습 I 정보 구조 설계하기

예제 미리보기



정보 구조 설계



● 실습 I 정보 구조 설계하기

● 테스크 플로우 작성하기



● 페이지 플로우 작성하기





정보 구조 설계

- 실습 I 정보 구조 설계하기
 - 카테고리별 정보 구조 설계하기





- 플로우 차트

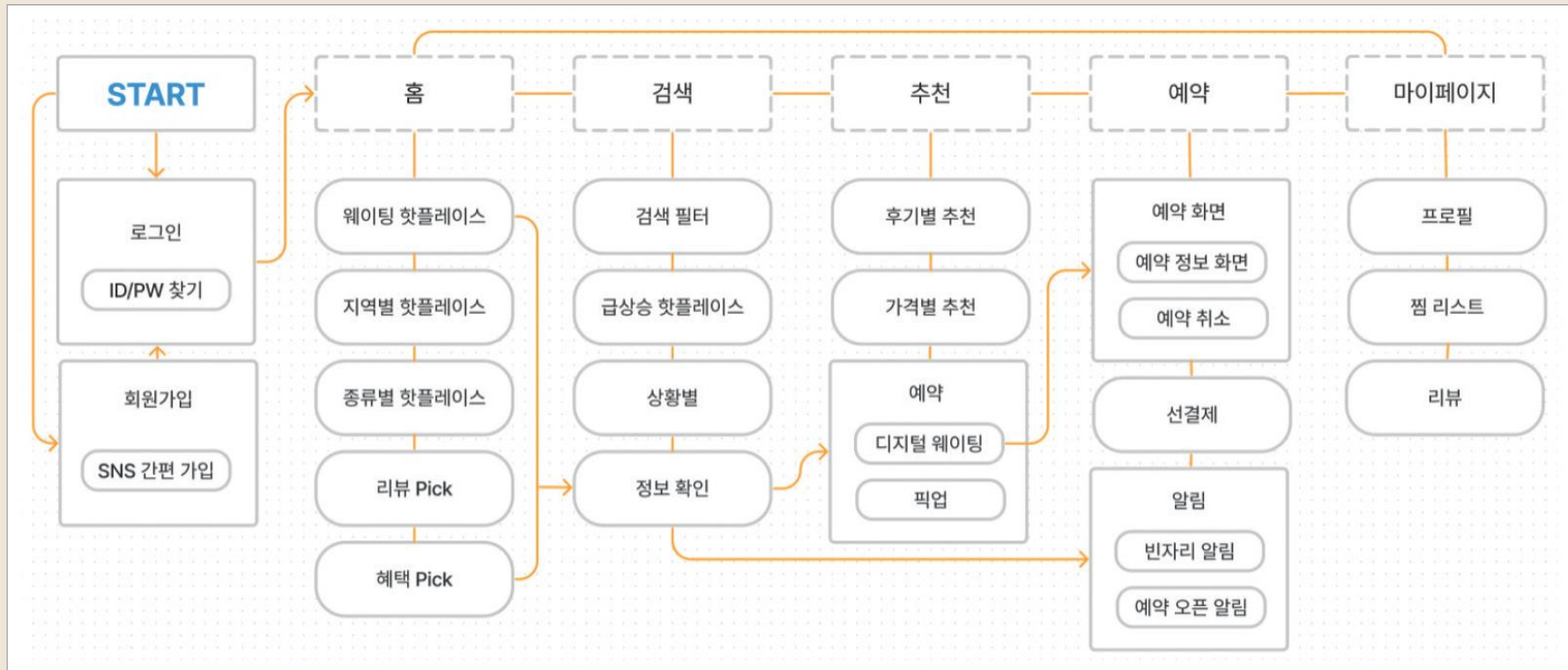
- 플로우 차트 : 사용자에게 제품이나 서비스를 이용하는 동선을 알려주는 순서도
- 전 세계적으로 통용되는 표준 기호를 사용하여 작성
- 플로우 차트를 표준 기호에 맞게 작성하지 못하더라도 정보 구조 설계로 구성한 메뉴 트리에서 화살표로 이동 동선을 연결하는 방법이 있음

정보 구조 설계



























● 실습 I 플로우 차트 그리기

예제 미리보기





하나 더 알기

기호	이름	기호	이름
	진행		종이 테이프
	종결		디스플레이
	결정		수동 조작
	문서		예비
	데이터		병렬 모드
	사전에 정의된 프로세스		반복 제한
	저장된 데이터		페이지상의 참조
	내부 저장소		오프 페이지상의 참조 형태
	순차적 데이터		예/아니요 결정 기호
	직접 데이터		조건
	수동 입력		제어 전송
	카드		주석

와이어프레임



● 와이어프레임의 이해

- 와이어프레임 : 각 화면 단위의 UI를 설계하는 작업으로, 화면의 설계와 화면 간의 관계와 이동 동선까지 모두 표현함

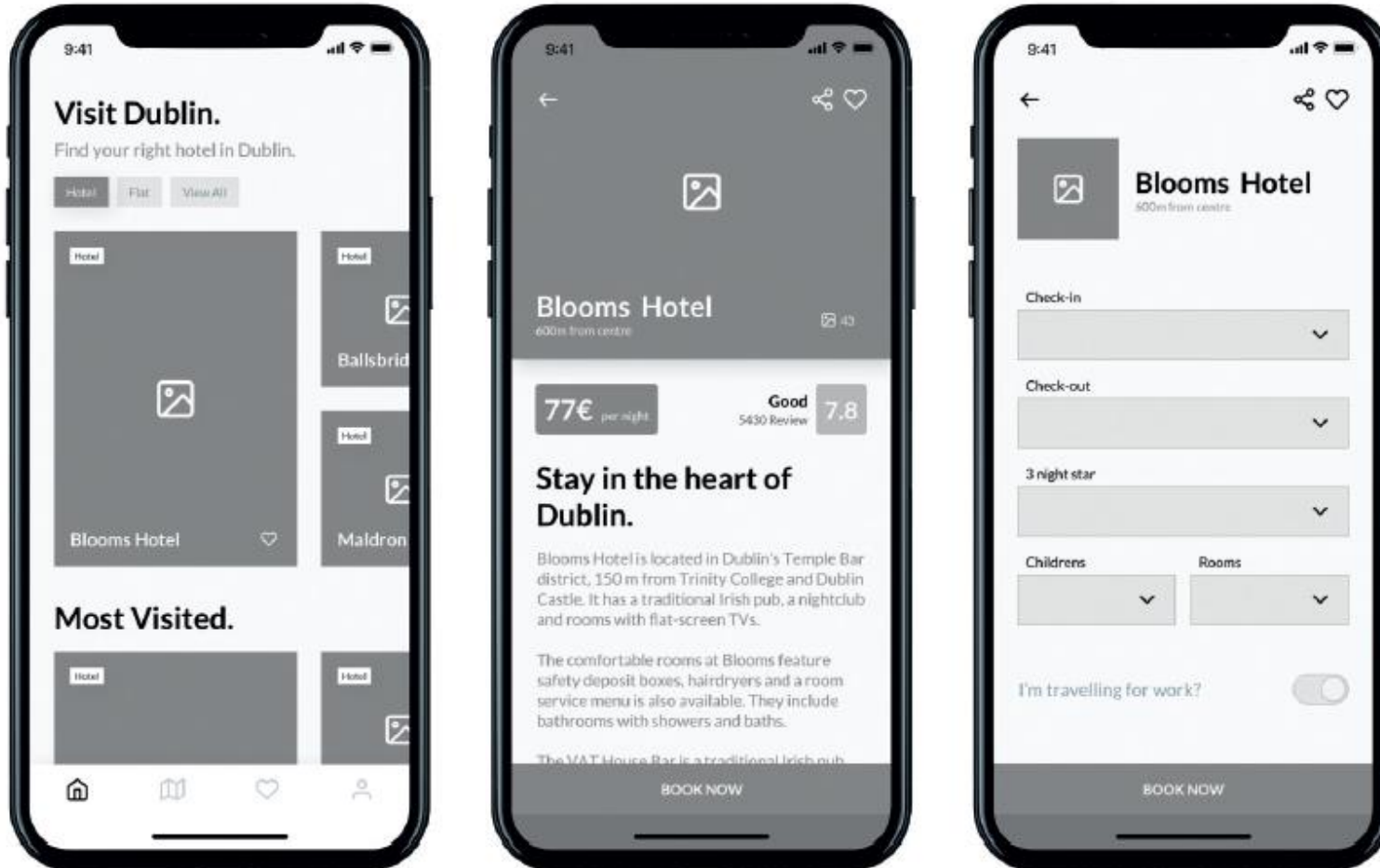


(a) 낮은 충실도의 와이어프레임(핸드 드로잉)

와이어프레임



● 와이어프레임의 이해



(b) 높은 충실도의 와이어프레임(소프트웨어 사용)

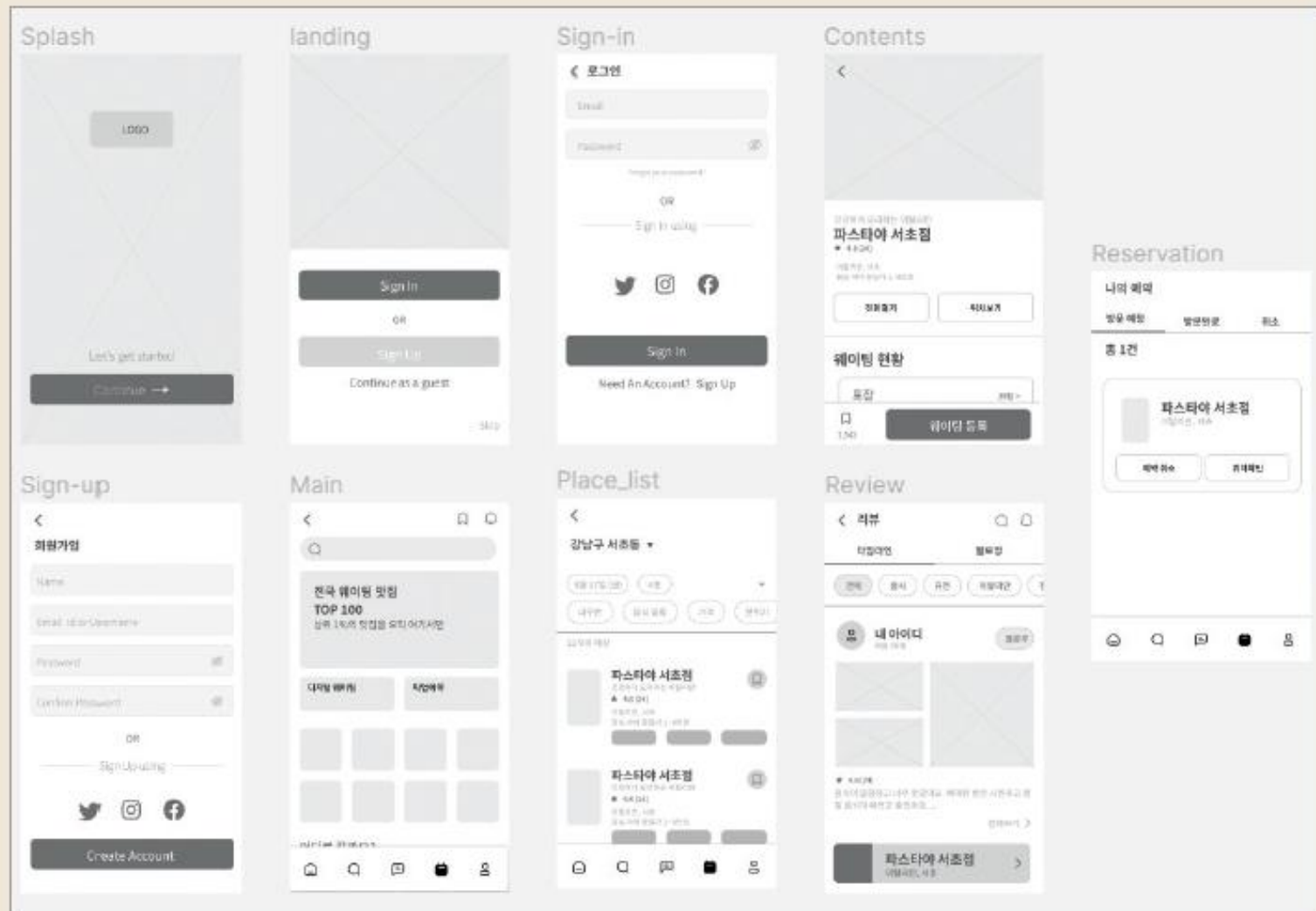
와이어프레임 작성

와이어프레임



● 와이어프레임의 이해

예제 미리보기

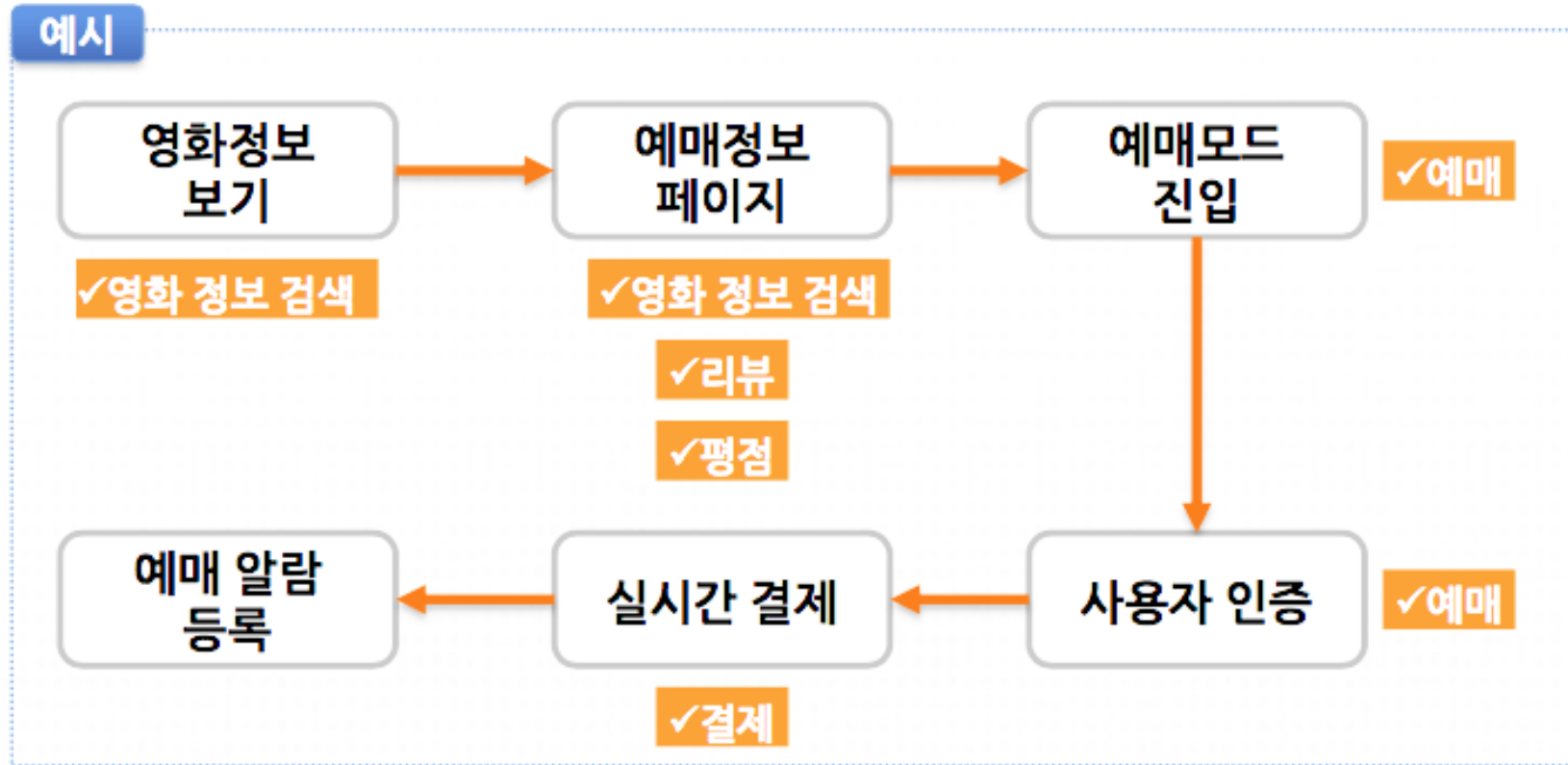


■ 와이어프레임 구성의 예

앱 설계 예시



- User Task-Flow 작성 단계

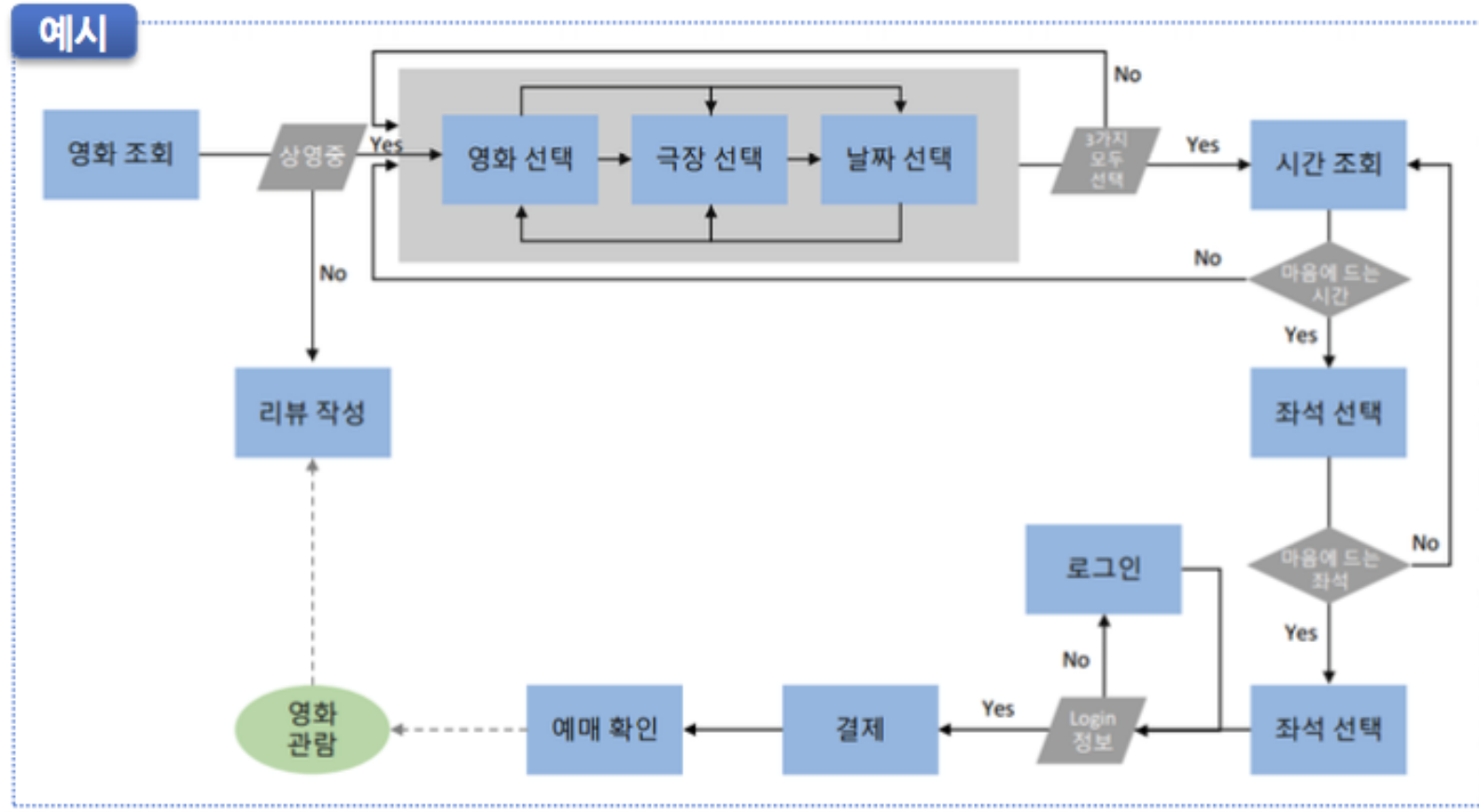


- 기능수행을 위한 사용자의 행위 순서(흐름)로 나열하고 검증하는 단계

앱 설계 예시



● Process Diagram 작성 단계

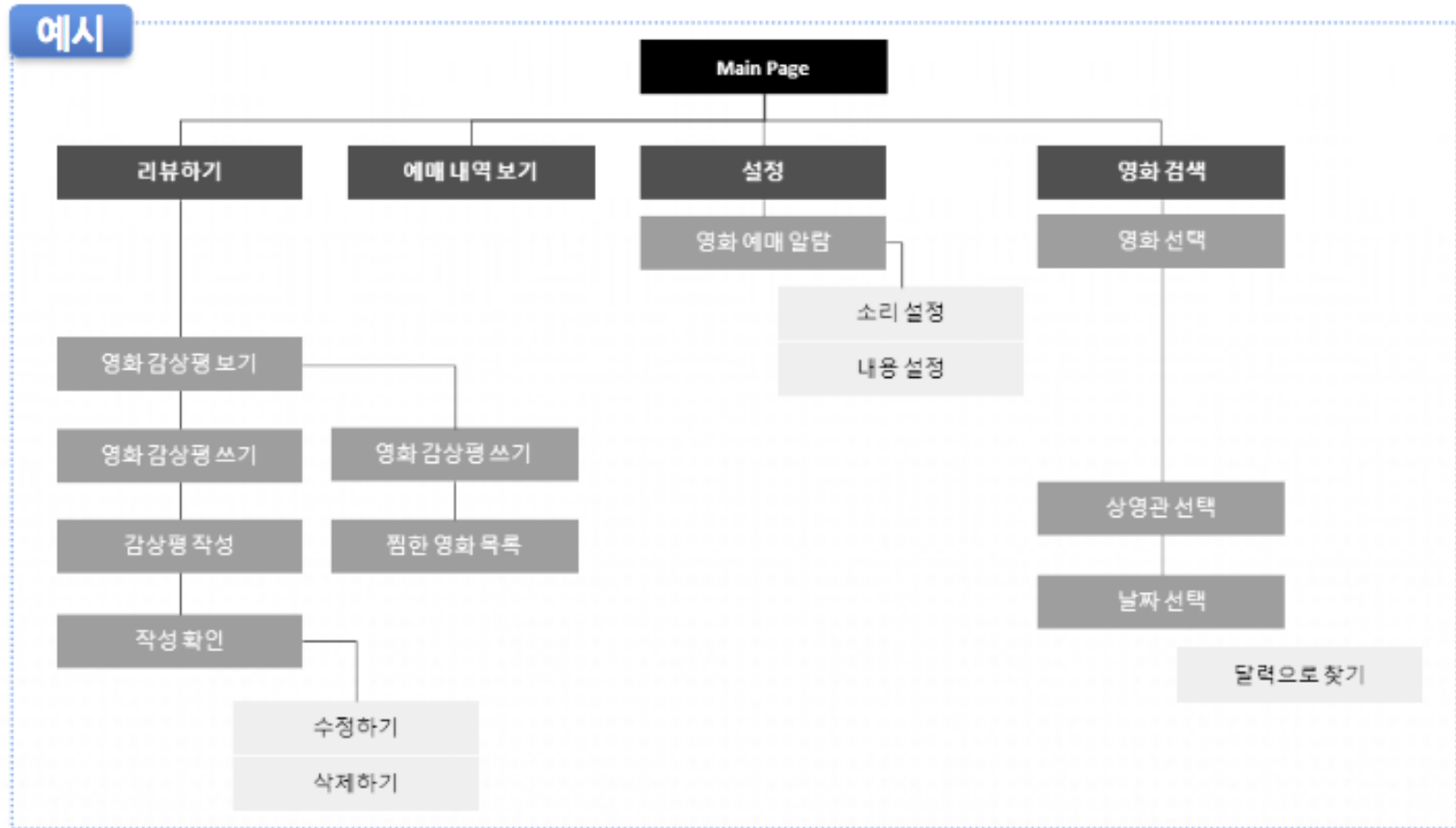


- 각 기능들 사이 관계(조건과 실행)를 정립하여 구조화하는 단계

앱 설계 예시



● IA(Information Architecture) 구조화



● 각 기능들을 정보의 위계와 관계망으로 구조화

앱 설계 예시



- Wireframe 설계 단계



- 해당기능 구현을 위한 인터페이스와 실제 화면에 그림

데이터 기반 문제 해결



문제 해결



- 문제를 해결한다는 것은 어떤 목적이나 목표를 달성하기 위한 해법이나 도움을 만들어 낸다는 것

문제 이해

현재 상태

문제가 발생한
처음의 상태



목표 상태

문제를 해결했을 때의 상태

**목표 상태에 도달하기 위해
수행해야 할 작업을 찾아야 함.**



문제 확인



현재 상태와 목표 상태를 파악



문제 해결에 필요한 핵심 요소 추출



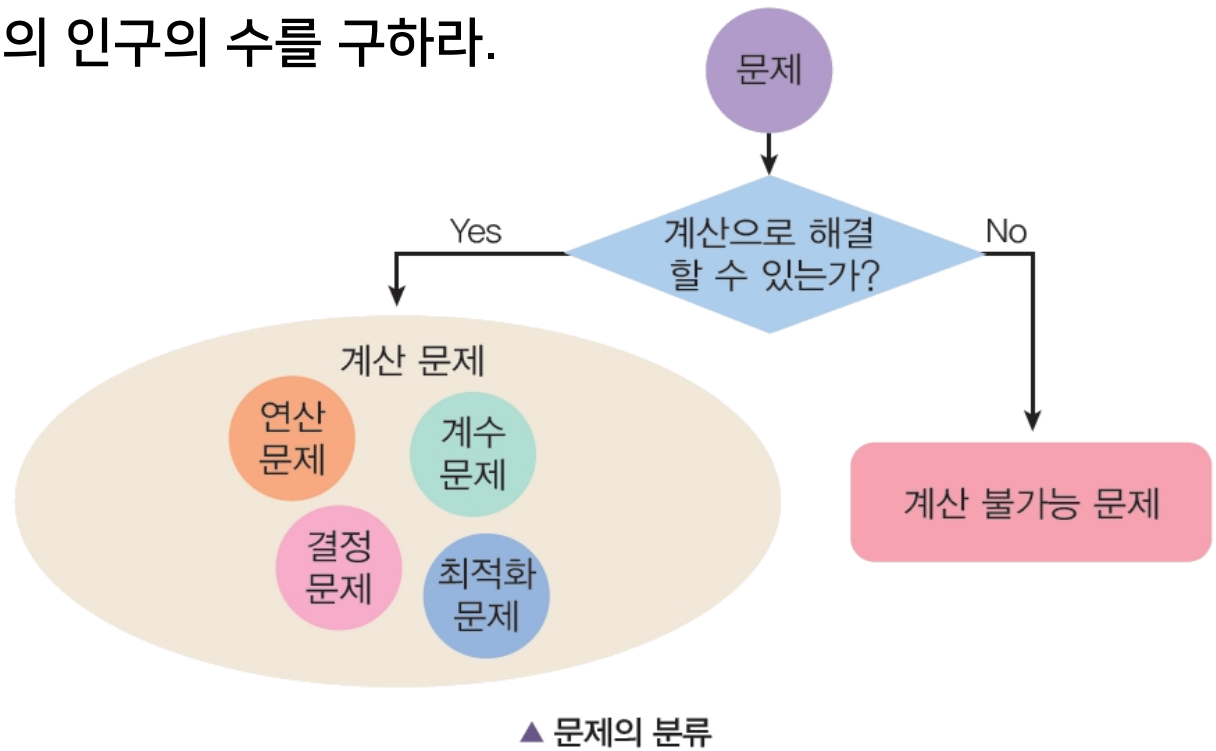
문제 분석

- 복잡한 문제를 해결할 때 해결해야 할 문제가 무엇인지, 문제를 해결하기 위해 어떤 행동을 수행해야 하는지 등을 명확하게 구분하는 과정



컴퓨터를 이용한 문제의 종류

- 컴퓨터의 계산 능력을 이용하여 해결 가능한 문제를 뜻하며, 절차적으로 해결할 수 있는 모든 문제라고 정의
- 계산 문제(O) : 12부터 1327까지의 자연수 중에서 소수의 개수는 몇 개인가?
- 계산 문제(X) : 현재 지구의 인구는 70억 명을 돌파하였다.
앞으로 10년 후의 인구의 수를 구하라.





데이터 기반 문제 정의의 특성

- 데이터를 수집하고 분석하여 문제를 정의하고 해결하는 과정
- 데이터를 통해 문제를 탐색하고, 데이터가 지닌 패턴, 통계, 관계를 바탕으로 해결책을 찾는 방식
- 문제 해결 과정에서 머신 러닝, 통계 분석, 데이터 시각화 등의 기법이 활용

불명확한
문제

비정형
데이터 처리

데이터
수집과 가공

통계기반
데이터 분석

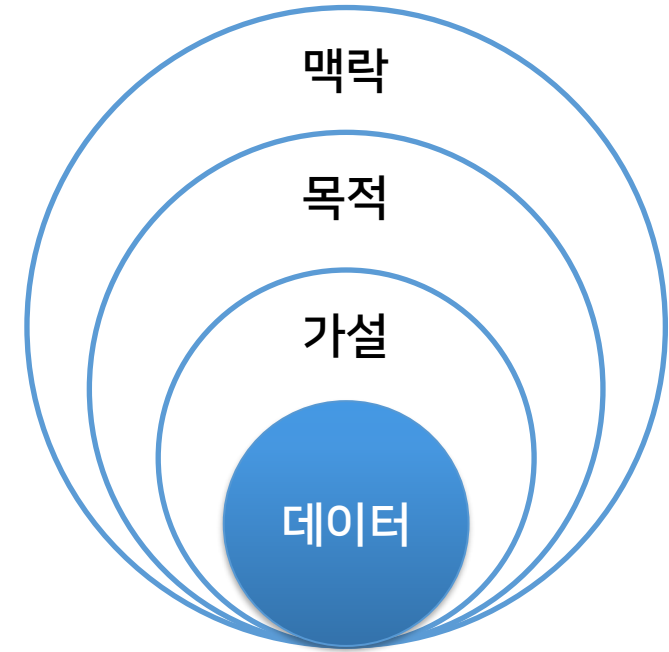
예측과 추론

복잡한
의사결정



데이터 기반 문제 정의

- 문제를 정확하게 정의하고 목적을 설정하는 과정
- 데이터 분석의 목표 → 데이터로 무엇을 할 것인가?
 - 현상의 이해
 - 현상의 일반화
 - 현상의 예측
- 데이터 활용 대상 → 데이터는 누가 쓸 것인가?
- 데이터 → 어떤 형태의 데이터인가?

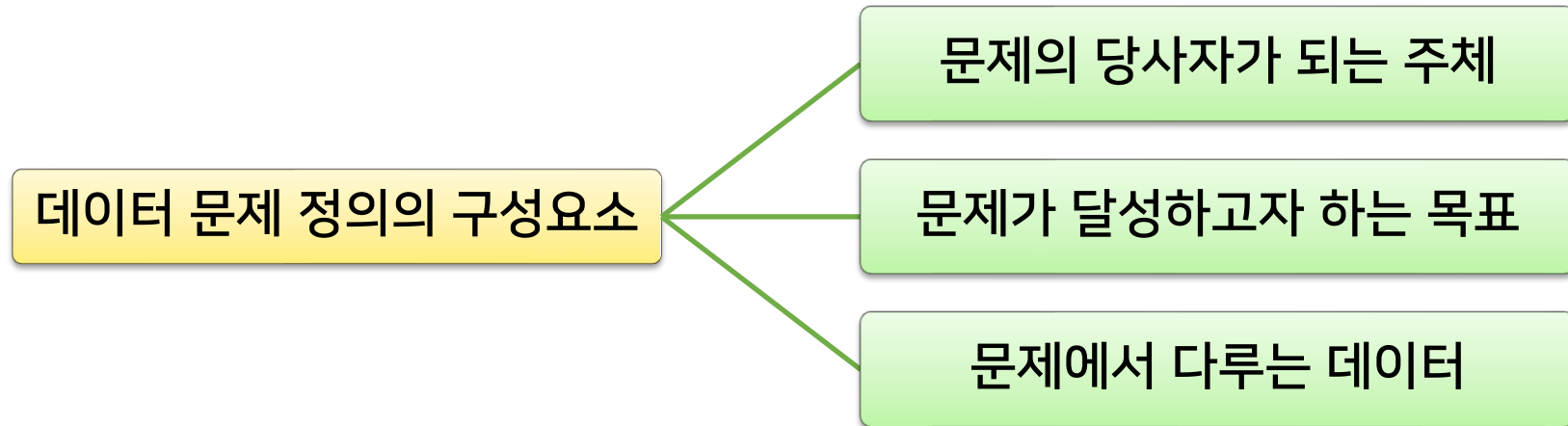


“데이터는 그 자체가 목적이자 답이 되지 않는다”

만약 나에게 문제 해결을 위해 한시간이 주어진다면, 나는 55분동안 문제에 대해 생각하고 5분동안 해결책에 대해 생각하겠다.

— 알버트 아인슈타인

데이터 기반 문제의 구성요소

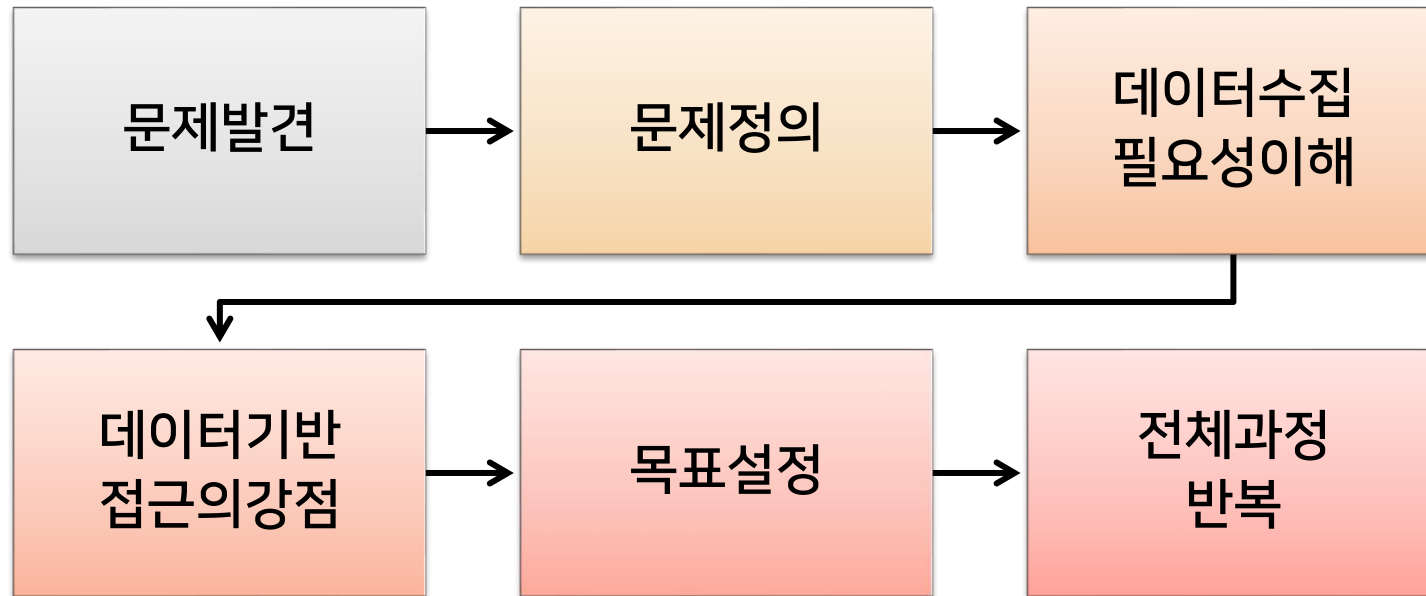


- 구성요소에 따라 문제의 성격 및 해결을 위한 도구와 방법이 결정됨
- 주체 : 개인이 주체 / 조직이 주체
- 데이터 : 데이터의 존재 유무 / 데이터의 크기나 형태
- 목표 : 현상의 이해(탐험적 데이터 분석) / 현상의 일반화(통계적 추론) / 현상의 예측(기계학습 모델링)



데이터 기반 문제 정의 과정

- 데이터를 활용하여 해결해야 할 구체적인 문제를 명확히 설정하는 과정
- 데이터 분석의 목적을 분명히 하고, 해결하려는 문제의 핵심을 이해하는 것이 중요





데이터 기반 문제 해결 과정의 예

● 문제 인식 (Problem Recognition)

- 첫 번째 단계는 문제를 인식하는 것
- 우리가 풀어야 할 문제 중 하나의 예시로, "왜 학생들이 학교 급식을 좋아하지 않을까?"라는 **질문**에서 시작할 수 있음
- 많은 학생들이 급식 메뉴에 대해 불만을 가지고 있다면, 학교는 학생들이 어떤 메뉴를 더 좋아하고, 어떤 메뉴를 싫어하는지 알고 싶어 할 것임.
- 이 단계에서는 " 학생들이 좋아하는 급식 메뉴가 무엇인지 " 를 파악하는 것이 목표
- 예시:
 - 문제: 학생들이 급식을 먹고 나서 불만을 자주 이야기해요.
 - 목표: 학생들이 더 만족하는 메뉴를 제공하고 싶어요.



데이터 기반 문제 해결 과정의 예

● 문제 정의 (Problem Definition)

- 문제를 인식한 후에는 이를 구체적으로 정의해야 함.
- 이 단계에서 " 학생들이 좋아하는 급식 메뉴는 무엇일까? " 또는 " 어떤 메뉴가 인기가 없을까? " 같은 구체적인 질문을 만들어야함.
- 이런 질문을 통해 문제를 더 명확하게 만들 수 있음
- 예시:
 - 추상적인 질문: "급식 만족도를 높이고 싶다"는 질문은 너무 막연해요.
 - 구체적인 질문: "학생들이 좋아하는 메뉴는 무엇인가?" 또는 "학생들이 만족하지 않는 메뉴는 무엇인가?"처럼 구체적으로 바꿔볼 수 있어요.



데이터 기반 문제 해결 과정의 예

● 데이터 수집 (Data Collection)

- 문제를 해결하기 위한 데이터를 수집해야 함.
- 이때 필요한 데이터는 학생들이 급식 메뉴에 대해 어떻게 느끼는지에 대한 정보임.
- **관련 데이터 준비를 위해 공개 데이터를 찾아 수집하거나 데이터를 생산(함설문조사 등을 진행)할 수 있음.**
- 예를 들어, 학생들에게 각 메뉴에 대해 1점에서 10점까지 점수를 매기게 해서 어떤 메뉴가 인기가 있고, 어떤 메뉴가 인기가 없는지를 알아보는 설문을 진행함
- 예시:
 - 설문조사: 각 급식 메뉴에 대해 "맛있다"부터 "별로다"까지 점수를 매기는 설문조사를 진행해요
 - 데이터: 각 학생이 준 점수와 어떤 메뉴가 평가되었는지 데이터를 모아요.



데이터 기반 문제 해결 과정의 예

● 데이터 전처리 (Data Preprocessing)

- 수집된 데이터를 분석하기 쉽게 정리해야함.
- 예를 들어, 설문조사에 빠진 답변이 있다면 그 데이터를 어떻게 처리할지 결정해야 하고, 잘못된 점수가 있을 수도 있어서 그런 문제들을 해결해야함.
- 이 과정을 통해 **정확하고 신뢰할 수 있는 데이터**를 만들 수 있음.
- 예시:
 - 설문조사에서 누락된 답변이 있는 경우, 그 답변을 제외할지 아니면 평균 점수로 채울지 결정해요.
 - 이상한 점수(예: 1에서 10점인데 15점을 준 경우)는 어떻게 처리할지 정해요.



데이터 기반 문제 해결 과정의 예

● 탐색적 데이터 분석 (**Exploratory Data Analysis, EDA**)

- 수집된 데이터를 분석하기 위하여 데이터를 시각적으로 표현하거나 통계적으로 분석함
- 예를 들어, 어떤 메뉴가 인기 있는지 알아보기 위해 각 메뉴의 평균 점수를 구하거나, 점수를 그래프로 그려서 어떤 메뉴가 높은 점수를 받았고, 어떤 메뉴가 낮은 점수를 받았는지 확인함
- 예시:
 - 각 메뉴의 평균 점수를 계산해요. 예를 들어, 스파게티는 8점, 미역국은 5점이라고 할 수 있어요
 - 이 점수를 그래프로 나타내서 한눈에 어떤 메뉴가 인기가 많은지 알 수 있어요.



데이터 기반 문제 해결 과정의 예

● 가설 설정 및 검증 (Hypothesis Setting and Testing)

- 가설이란, 예상하는 답
- 가설을 세우고 검증하는 단계가 필요
- 예를 들어, "학생들은 매운 음식을 좋아하지 않을 것이다"라는 가설을 세우고, 데이터를 통해 이 가설이 맞는지 틀린지 추론하여 확인 함
- 예시:
 - 가설: "학생들은 매운 음식을 별로 좋아하지 않을 것이다."
 - 검증: 매운 음식에 대한 점수를 모아서 평균을 계산해요.
실제로 매운 음식이 낮은 점수를 받았으면 가설이 맞는 것이고,
높은 점수를 받았다면 틀린 거예요.



데이터 기반 문제 해결 과정의 예

● 모델링 및 분석 (Modeling and Analysis)

- 데이터를 바탕으로 모델을 생성하고, 새로운 데이터를 예측 분석함
- 모델은 예측을 도와주는 도구
- 예를 들어, 앞으로 새로운 메뉴를 추가할 때, 학생들이 이 메뉴를 좋아할지 예측할 수 있는 모델을 만들 수 있음. 이를 통해 학교는 어떤 메뉴를 더 자주 제공해야 할지 결정을 내릴 수 있음.
- 예시:
 - 모델을 사용하여 "만약 이 메뉴가 추가된다면, 학생들은 이 메뉴에 몇 점을 줄까?"를 예측할 수 있어요.
 - 모델은 과거 데이터를 바탕으로 새로운 메뉴의 점수를 예측하고, 이를 통해 더 나은 결정을 내릴 수 있어요.



데이터 기반 문제 해결 과정의 예

● 결과 해석 및 결론 도출 (Interpretation and Conclusion)

- 분석 결과를 해석하고 결론을 도출.
- 데이터를 바탕으로 예를 들어, 어떤 메뉴를 더 자주 제공해야 할지, 또는 어떤 메뉴를 바꾸거나 없애야 할지 결정할 수 있음.
- 점수가 높은 메뉴는 더 자주 제공하고, 낮은 점수를 받은 메뉴는 변경함.
- 예시:
 - 결과: 스파게티와 치킨 메뉴는 점수가 높으니까 자주 제공하고, 점수가 낮은 미역국은 개선하거나 다른 메뉴로 바꿀 수 있어요.
 - 이 데이터를 바탕으로 결론을 내리고, 급식 메뉴를 어떻게 개선할지 결정해요.



데이터 기반 문제 정의

- 어떤 문제에 직면했을 경우 가장 먼저 데이터 자체에서부터 시작할 것인가?
아니면 비즈니스 문제에서 시작할 것인가를 고민
- '데이터 활용 목적'을 분명히해야함
- '목적 중심의 데이터 사고방식'이 중요
- 데이터 활용 프로세스는 발견한 현상을 구체적인 목적과 문제로 정의하여 이에 맞는 데이터를 결정해 현황을 파악하고, 결론을 도출하여 해결 방안을 모색하는 과정





데이터 리터러시

1. 데이터 기획

해결하고자 하는 주제 및 문제를 데이터로 공감
목적에 맞는 데이터의 정의, 필요한 데이터 지표 설계

2. 수집 · 관리

필요한 데이터를 빠른 시간 내 검색, 선별하여 확보
데이터를 분석이 가능한 형태로 구조화, 정제

3. 가공 · 분석

이미 있는 결과를 도출할 수 있도록 가공 및 구조 분석
목적에 맞는 분석 방법(알고리즘)을 선정하여 실행

4. 시각화 · 해석

데이터를 시각화로 요약하여 그래프, 차트 등으로 표현
데이터의 숨겨진 의미 혹은 문제를 발견하고 해석, 비판

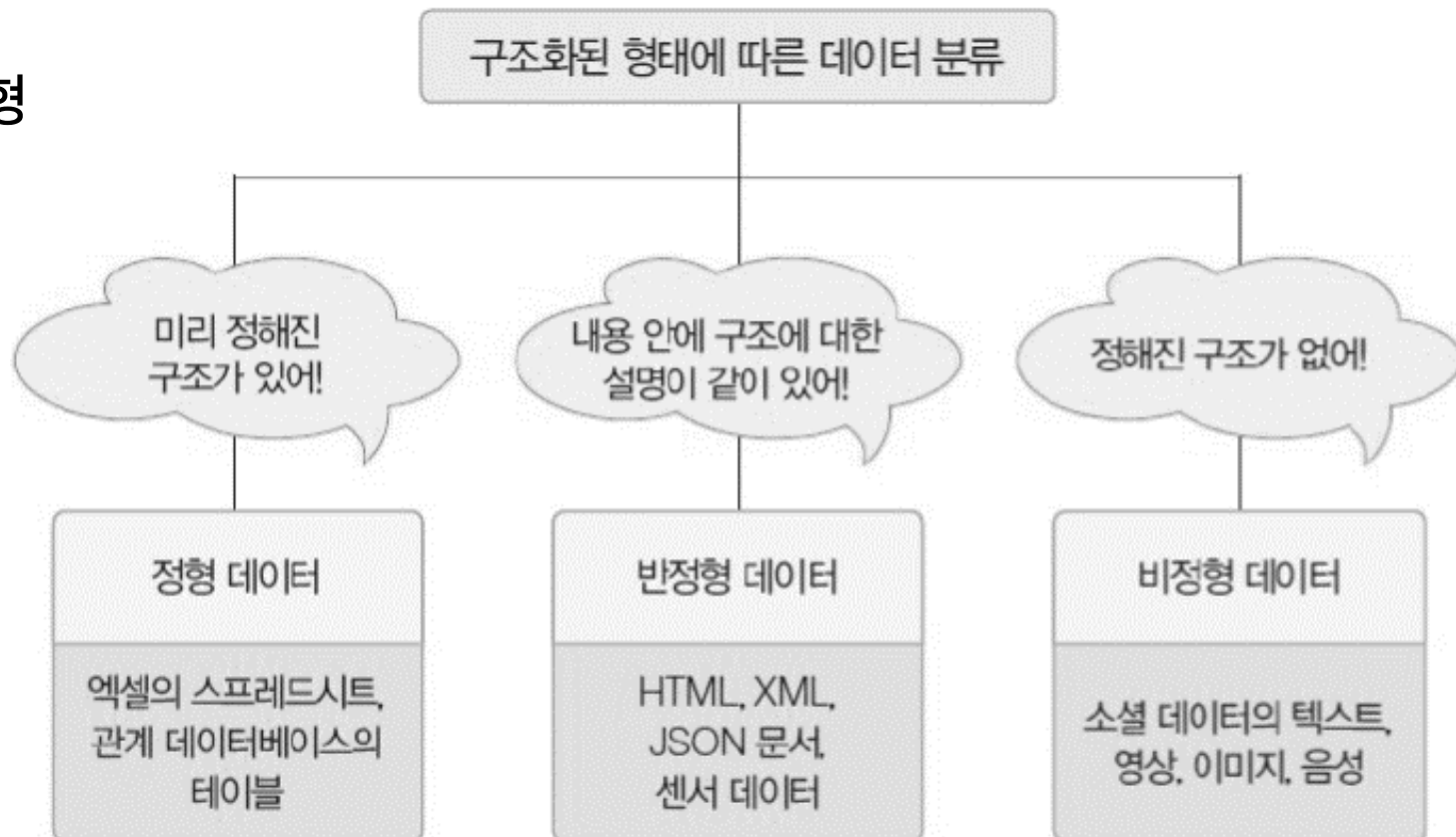


데이터

● 데이터 정의

- 자료(資料, Data, 데이터)는 수, 단어 등의 형태로 된 의미 단위
- 바탕이 되는 재료, 자료를 의미 있게 처리하여 정보(Information)로 활용

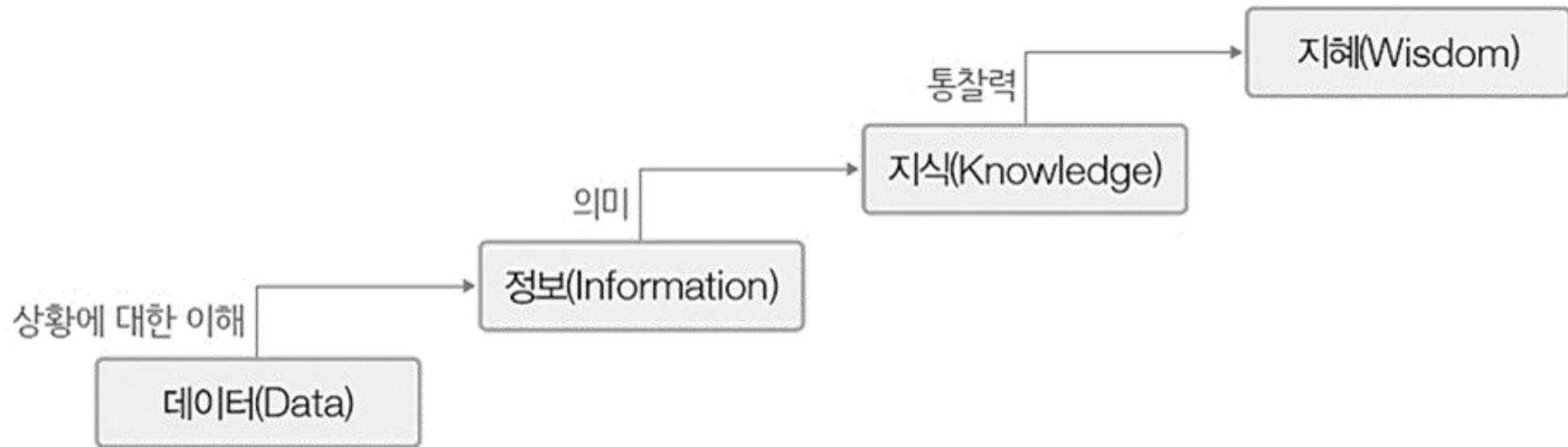
● 데이터 유형





데이터 과학의 필요성

- 4차 산업혁명 시대 → 디지털 대전환 시대 → 인공지능 대전환 시대
 - 빅데이터, 사물 인터넷, 인공지능 등 핵심 기술의 중심에 데이터가 있음
 - 21세기의 원유는 데이터
- 데이터의 방대한 규모와 다양한 형태
 - 전통적인 방식으로 수집하고 저장하는데 한계가 있음
- 다양해진 데이터 활용에 대한 요구
 - 단순히 데이터를 분류 검색하는 것을 넘어, 방대한 양의 데이터 속에 숨겨진 규칙과 패턴을 찾아내 문제 해결에 활용하고 미래의 일을 예측하여 미리 준비하기를 원함





빅데이터

- 기존 **데이터베이스**가 저장하고 관리할 수 있는 범위를 넘어서는 대규모의 다양한 데이터
- 대규모 데이터를 저장 및 관리하는 기술과 가치 있는 정보를 만들기 위해 분석하는 기술 까지 포함
- 빅데이터 기본 특징 : 3V

- **규모(Volume)**

- ✓ 테라바이트(TB) 단위 이상의 대량 데이터

- **속도(Velocity)**

- ✓ 데이터의 수집 분석을 정해진 시간 내에 처리해야 함

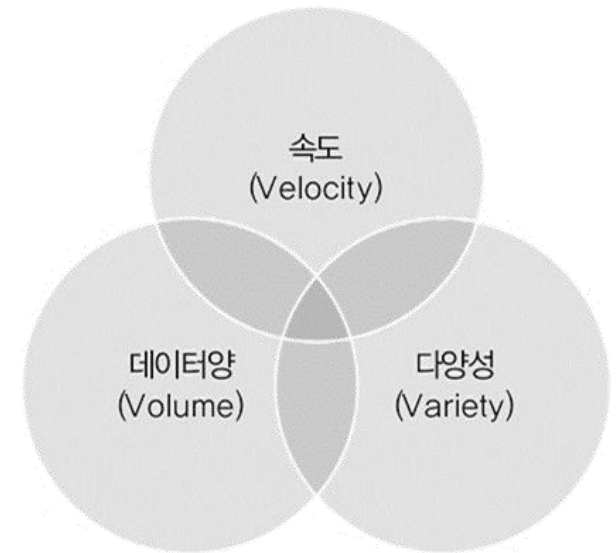
- **다양성(Variety)**

- ✓ 형태의 다양성이 존재

- **정형 데이터** : **관계 데이터베이스**와 같이 정형화된 시스템에 저장된 데이터 형태

- **반정형 데이터** : 정형화된 시스템에 저장되어 있지 않지만 내부적으로 스키마를 어느 정도 포함하고 있는 XML, HTML 등을 의미

- **비정형 데이터** : 구조가 정해져 있지 않은 데이터





빅데이터 기술

● 빅데이터의 기술 – 저장 기술

- 데이터의 용도와 환경에 맞게 선택
 - 관계 데이터베이스 – 정형 데이터 관리
 - 하둡(Hadoop)
 - NoSQL
- 비정형 데이터 관리하는데 유리함

● 빅데이터의 기술 – 분석 기술

- 데이터 마이닝(data mining)
 - 분류 분석(classification analysis)
 - 군집 분석(cluster analysis)
 - 연관 분석(association analysis)

● 빅데이터의 기술 – 표현 기술

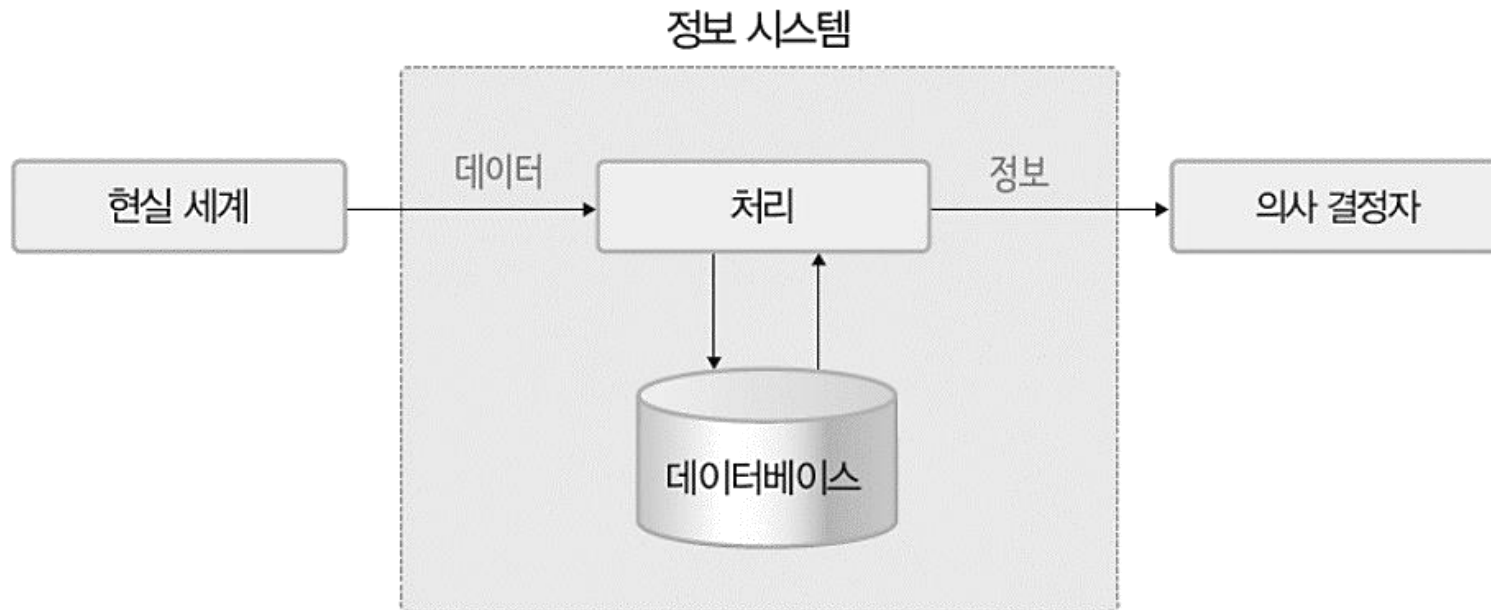
- 데이터 시각화
 - 데이터 분석을 통해 추출한 의미와 가치를 시각적으로 표현
- 인포그래픽





정보시스템과 데이터베이스

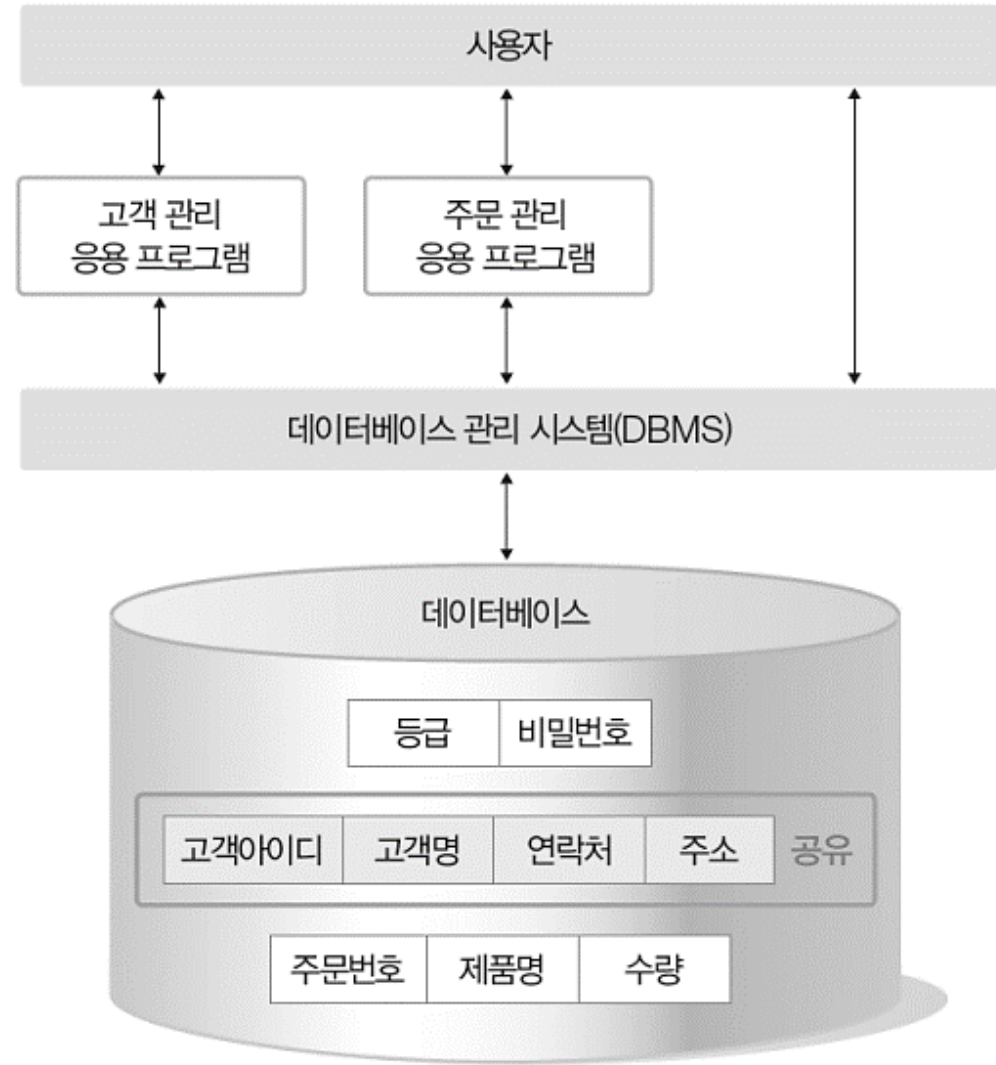
- 조직 운영에 필요한 데이터를 수집하여 저장해두었다가 의사 결정이 필요할 때 처리하여 유용한 정보를 만들어주는 수단을 정보 시스템
- 정보 시스템 안에서 데이터를 저장하고 있다가 필요할 때 제공하는 핵심 역할은 데이터 베이스가 담당





DBMS(데이터베이스관리시스템)

- 파일 시스템의 문제(데이터 중복과 데이터 종속)를 해결하기 위해 제시된 소프트웨어
- 조직에 필요한 데이터를 데이터베이스에 통합하여 저장하고 관리



DBMS의 유형



- **관계형(Relational) DBMS,**
 - 테이블(table)이라는 최소 단위로 구성 테이블은 하나 이상의 열(column)과 행(row)으로 구성
- **계층형(Hierarchical) DBMS,**
 - 처음으로 등장한 DBMS 개념으로 1960년대에 시작, 지금은 잘 사용하지 않는 유형
- **망형(Network) DBMS,**
 - 계층형 DBMS의 문제점을 개선하기 위해 1970년대에 등장
- **객체지향형(Object-Oriented) DBMS,**
 - 기존 관계형(Relational) 데이터베이스 개념에 객체 개념을 추가한 것으로 1980년대에 등장
- **객체관계형(Object-Relational) DBMS**
 - 관계형 데이터베이스와 객체 지향 프로그래밍 언어 간의 호환되지 않는 데이터를 자동으로 매핑(연결)해주는 프로그래밍 기법
- RDBMS는 정형데이터 유형을 관리하며 가장 많이 사용



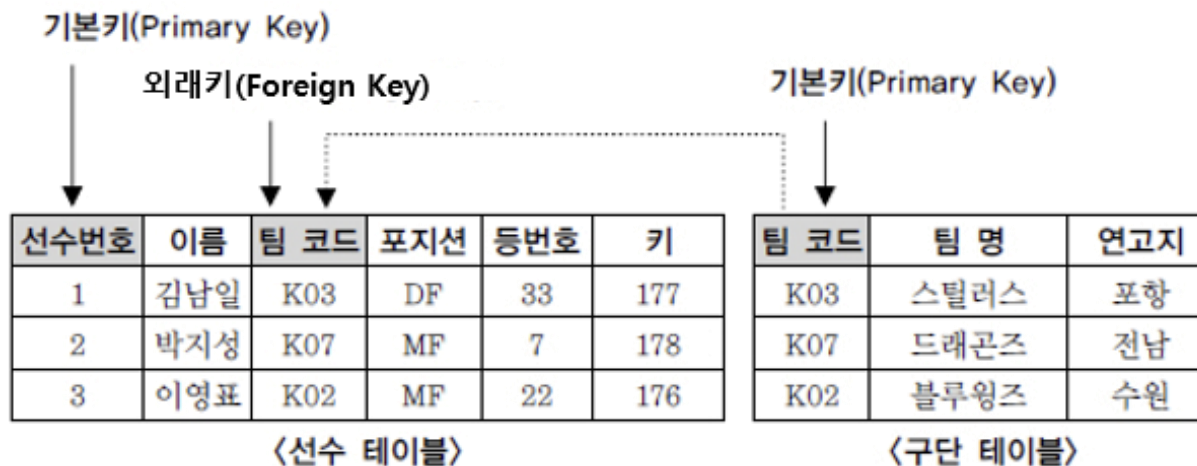
DBMS의 발전 과정

- 1세대 : 네트워크 DBMS, 계층 DBMS
- 2세대 : 관계 DBMS
- 3세대 : 객체지향 DBMS, 객체관계 DBMS
- 4세대 : NoSQL · NewSQL DBMS → 비정형 데이터를 처리하는데 적합, 확장성 높음





- 테이블, 행, 열의 정보를 구조화하는 방식 → 정형데이터 구조
- 대표적인 RDBMS: MS-SQL, MySQL, MariaDB, Oracle, Access 등
- 열(column): 각각의 열은 유일한 이름을 가지며, 자신만의 데이터 형식을 가짐
- 행(row): 행은 관계된 데이터의 묶음을 의미, 한 테이블의 모든 행은 같은 수의 열을 가짐
- 키(key): 하나의 테이블을 구성하는 여러 열 중에서 특별한 의미(식별 등)를 지닌 하나 또는 여러 열의 조합을 의미
- 관계(relationship): RDBMS에서 한 테이블 키와 다른 테이블의 행을 식별할 수 있도록 연결
 - 일대일(one to one) 관계 / 일대다(one to many) 관계 / 다대다(many to many) 관계





대표적인 DBMS

- **Oracle**: 1979년 오라클사에서 만들어 판매중인 상업용 DB이며, 점유율 1위 DBMS
- **MySQL**: 1995년 상용버전으로 서비스 시작, 저렴하고 오픈소스이며 무료 프로그램이라는 장점, 점유율 2위
- **MS-SQL**(Microsoft SQL Server): Microsoft 사에서 개발한 사업용 DB
- **PostgreSQL**: 1980년대 버클리 대학의 프로젝트로 시작, 강력한 오픈소스 객체-관계형 DBMS라는 특징
- **IBM DB2**: IBM사의 Transaction 및 Warehousing Workload에 고급 데이터 관리 및 분석 기능을 제공하는 관계형 데이터베이스
- **SQLite**: 임베디드(스마트폰 등)에 사용될 목적으로 개발된 RDMBS, 적은 메모리 상에서 DBMS를 구현하는데 사용
- **Microsoft Access**: PC단위에서 소규모 DBMS로 활용, Office 제품군으로 활용

ORACLE®
DATABASE





- 구글에서 제공하는 모바일 및 웹 애플리케이션 개발 플랫폼
- 실시간 데이터베이스, 인증, 호스팅, 클라우드 기능 등을 포함
- 실시간 데이터베이스는 데이터가 변경될 때마다 즉시 업데이트되는 데이터베이스
- 파이어베이스는 JSON 형식으로 데이터를 저장
 - 예) 사용자 정보를 저장할 때는 { 'users': { 'user1': { 'name': '홍길동', 'age': 30 } } }와 같은 구조로 저장
- WebSocket을 사용하여 클라이언트와 서버 간의 지속적인 연결을 유지
 - 데이터가 변경되면 서버는 모든 연결된 클라이언트에 변경 사항을 푸시
- 데이터 구조 최적화: 중복 데이터를 피하고, 필요한 데이터만 요청
- 쿼리 최적화: 필요한 데이터만 가져오도록 쿼리를 작성

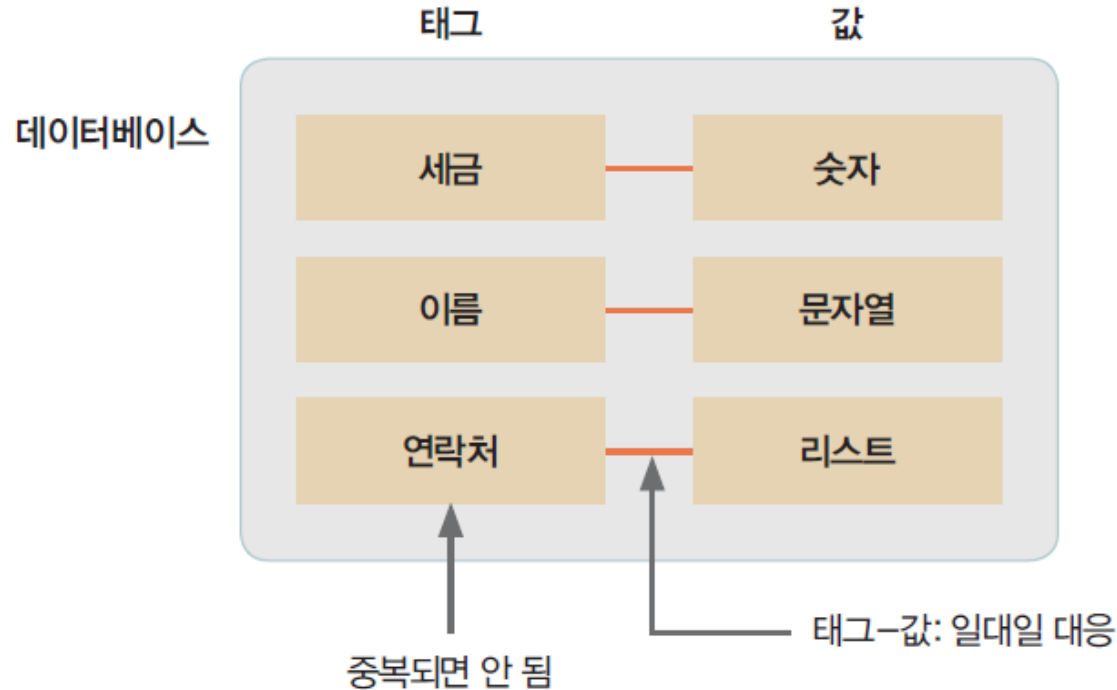


DB를 이용한 앱



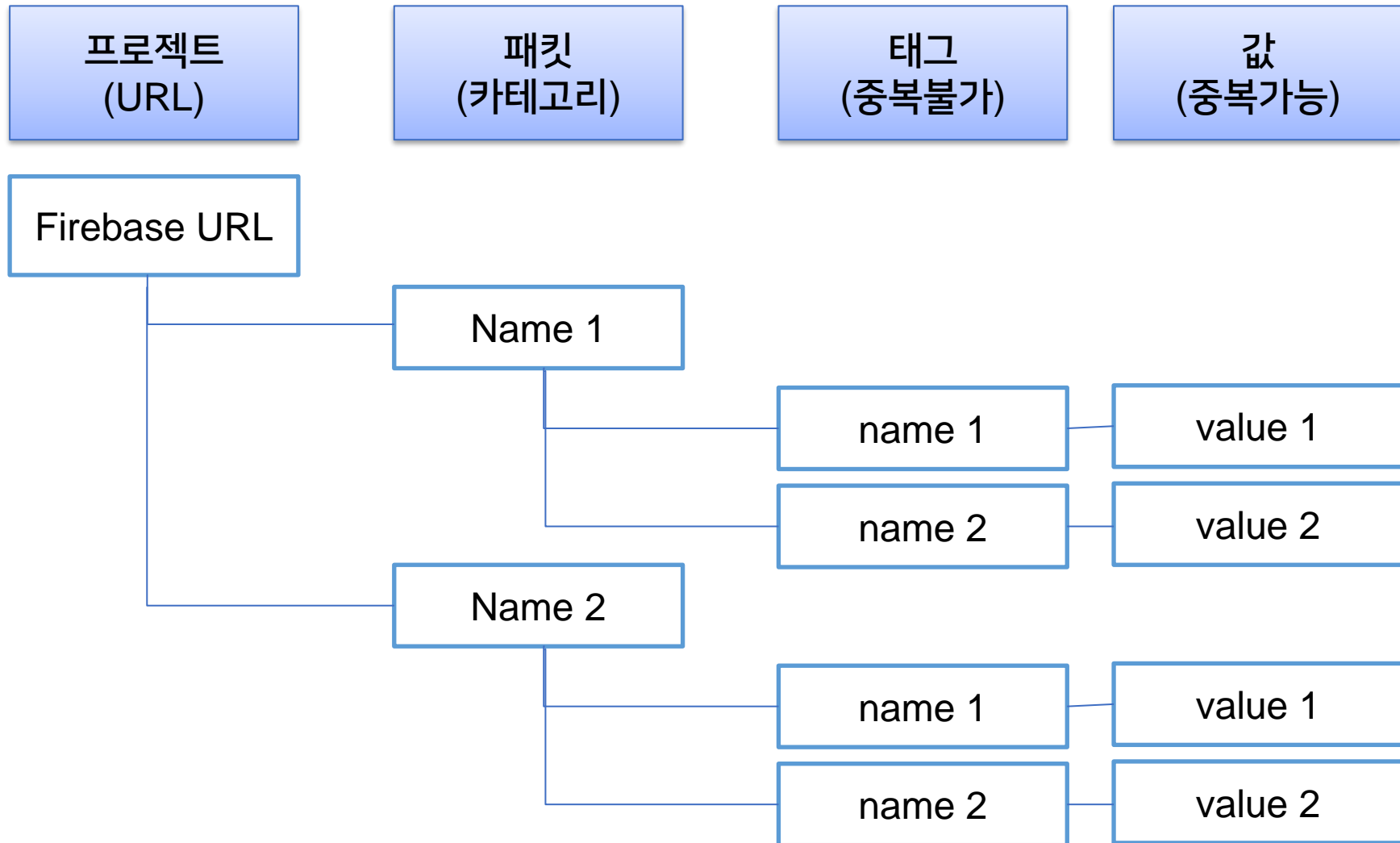
● 기초 지식 이해

- 데이터베이스 : 데이터를 체계적으로 분류하여 저장하고 검색하는 등 데이터 관리 기능을 제공
- 앱 인벤터가 지원하는 데이터베이스 컴포넌트들은 기본적으로 '태그-값'을 쌍으로 저장하고 있는 테이블을 데이터베이스로 간주



앱 인벤터 데이터베이스 컴포넌트의 내부 구조

Firestore 구성



DB를 이용한 앱 구조 예시

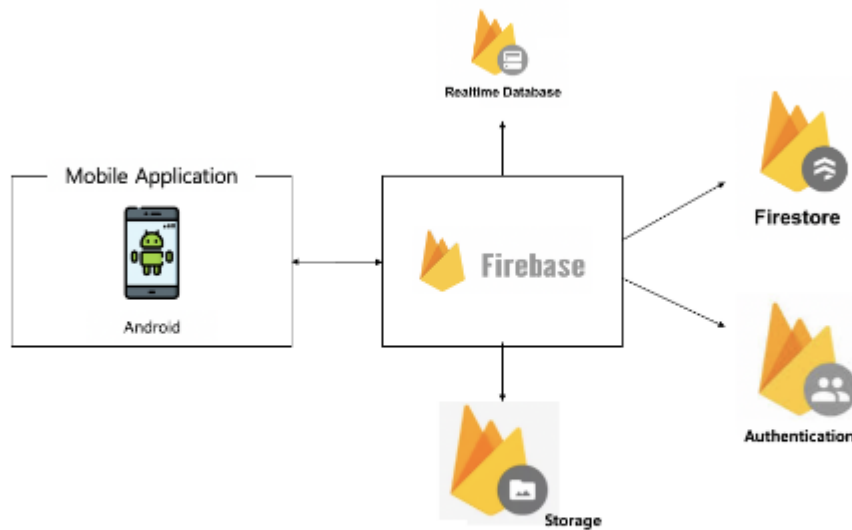


개발 환경

```
framework: ionic3, cordova  
language: typescript  
database: firebase
```



1. 시스템 구성도

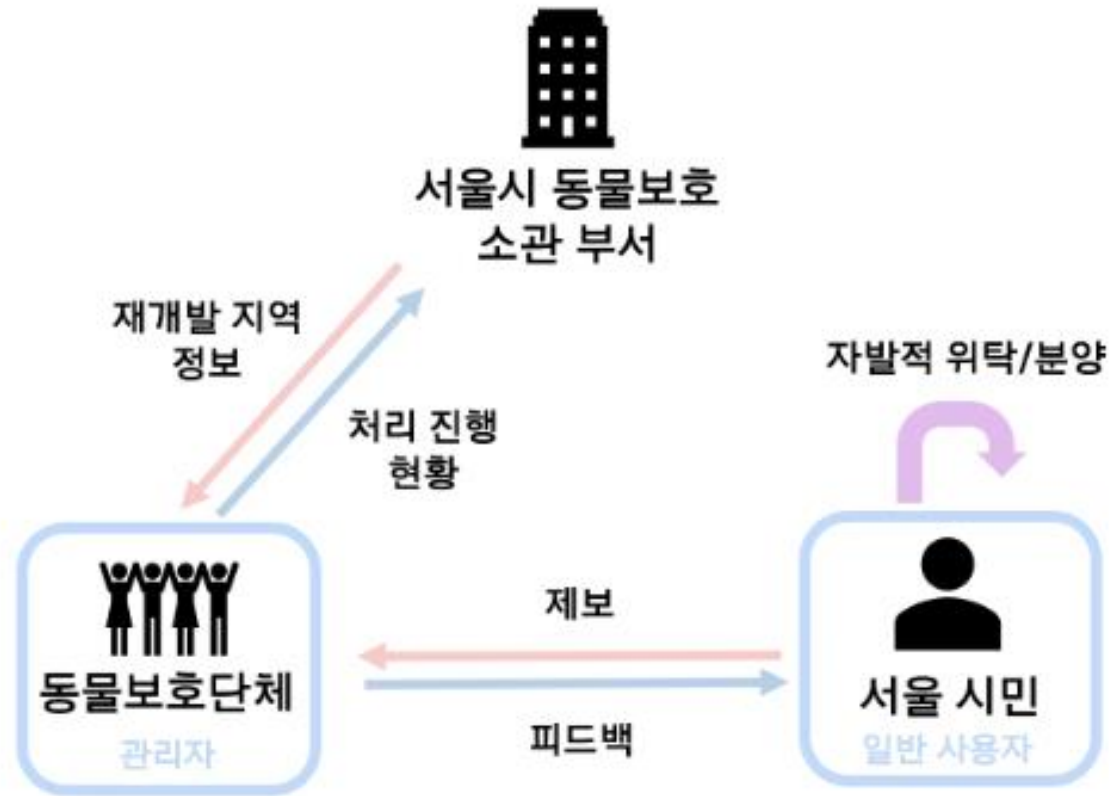


사용자는 모바일 어플리케이션, 즉 안드로이드를 통해 앱에 접속하여 글을 작성하고 확인할 수 있습니다. 데이터 베이스로 사용한 Firebase는 사용자 인증 시스템을 지원합니다. JSON 트리 구조를 이용하는 실시간 데이터베이스(Realtime Database)를 사용하여 채팅이 가능하고, 게시글과 제보같은 데이터는 Firestore에 저장했습니다. 사진 업로드에는 Storage가 쓰였습니다.

DB를 이용한 앱 구조 예시



2. 서비스 구조



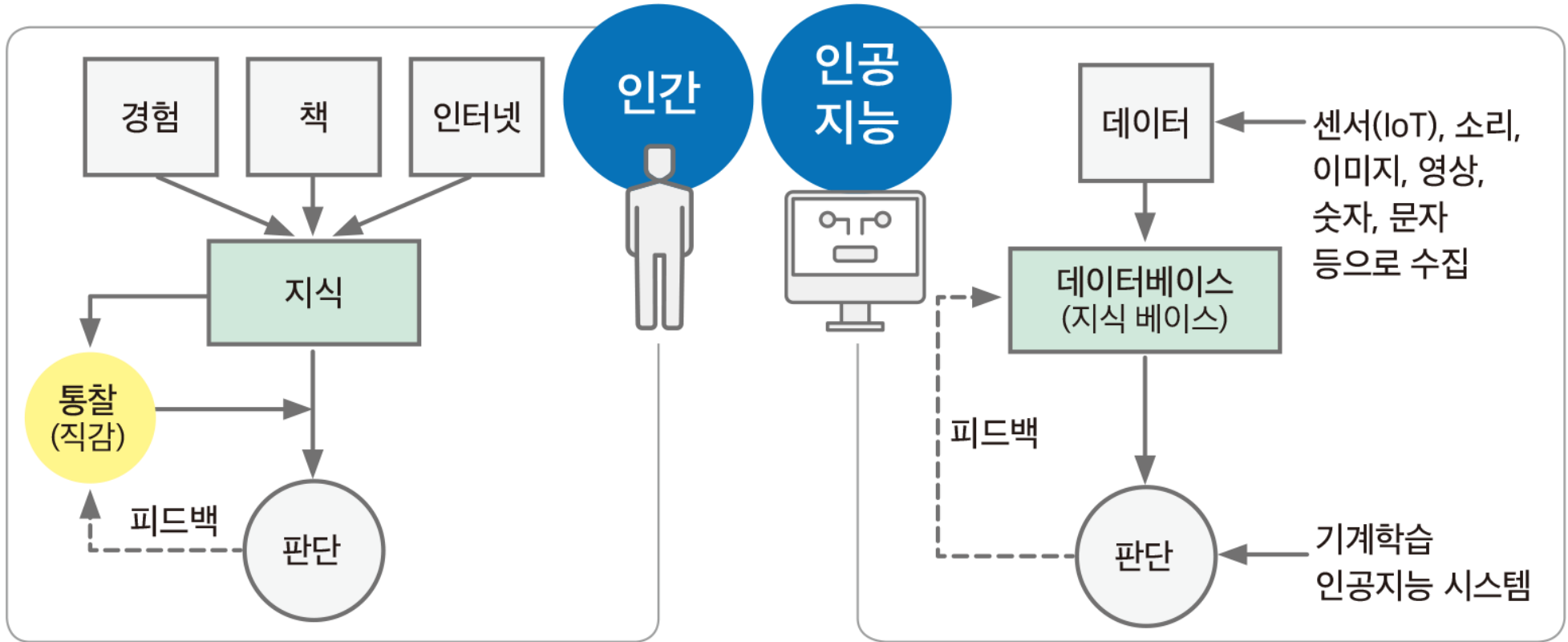
인공지능 기반 앱 설계



인공지능이 인간 지능보다 우수할까?



인간지능과인공지능





소프트웨어

- 기계가 수행해야 할 일을 순차적으로 기록한 프로그램과 데이터
- 상황에 맞게 점진적으로 개선 가능
- 오류 발생시 언제든지 수정 및 보완이 가능하고, 더 좋은 소프트웨어로 발전할 수 있음.

인공지능 소프트웨어

- 지식, 데이터, 알고리즘을 통해 스스로 답을 찾아가는 지능화된 형태
- 더 똑똑하고 합리적으로 다양한 문제 해결



인공지능이 적용된 소프트웨어와 적용되지 않은 소프트웨어

일반적인 소프트웨어

아직은 인공지능이 적용된 제품보다 인공지능이 적용되지 않은 일반 제품이 더 많아요.



개발자



일반 소프트웨어



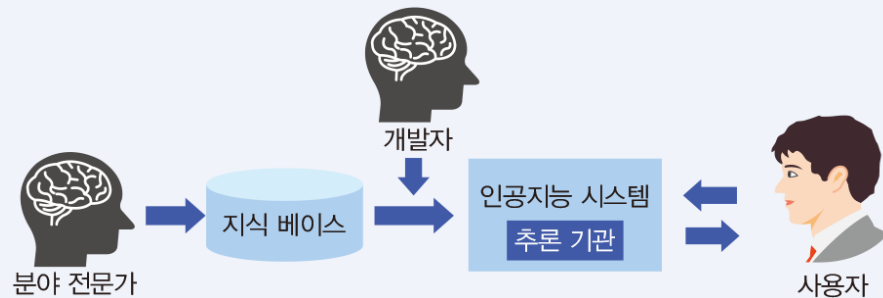
사용자

- 개발자가 문제 해결 절차를 코딩하여 소프트웨어를 제작한다.
- 사용자가 원하는 작업을 요청하면 미리 코딩된 알고리즘으로 처리하여 결과를 제공한다.
- 예외 상황을 처리하기가 힘들다.



인공지능이 적용된 소프트웨어와 적용되지 않은 소프트웨어

인공지능 소프트웨어(지식 기반 인공지능 소프트웨어)



전문가의 지식을 모아 놓은 지식 베이스로 만들어진 인공지능이 사용자의 요구에 대한 결과를 제공한다.

예 제빵사의 빵 굽는 비법을 모아 지식 베이스를 만들고, 이를 통해 만들어진 인공지능이 센서를 통해 입력된 내용을 바탕으로 어떤 동작을 수행해야 할지를 결정한다.



인공지능이 적용된 소프트웨어와 적용되지 않은 소프트웨어

인공지능 소프트웨어(지식 기반 인공지능 소프트웨어)



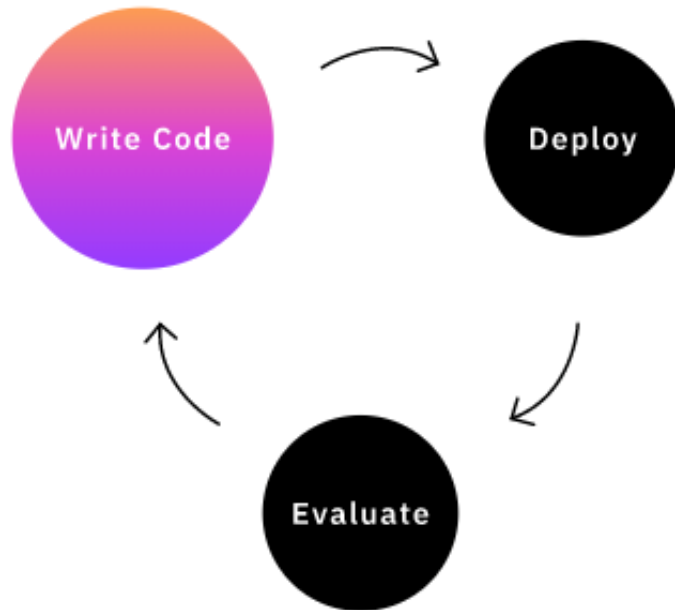
데이터(답) 학습 알고리즘을 통해 스스로 규칙을 만들고, 사용자의 요구를 처리한다.

예 폰 카메라의 소프트웨어는 사진을 찍는 환경의 조도와 피사체와의 거리 등에 대한 설정을 미리 학습된 데이터를 바탕으로 변경하여 최적의 환경으로 촬영하고, 사진에 찍힌 인물을 인식하여 카테고리별로 분류한다.

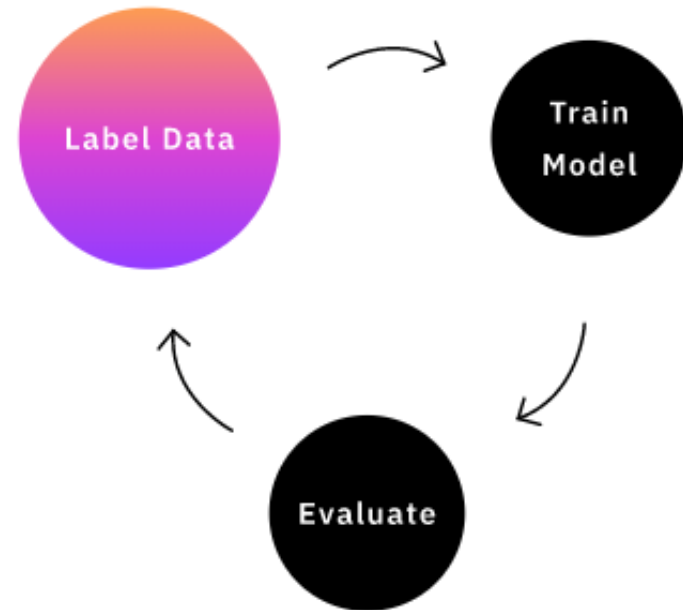
일반 SW 개발 사이클과 인공지능 SW 개발 사이클



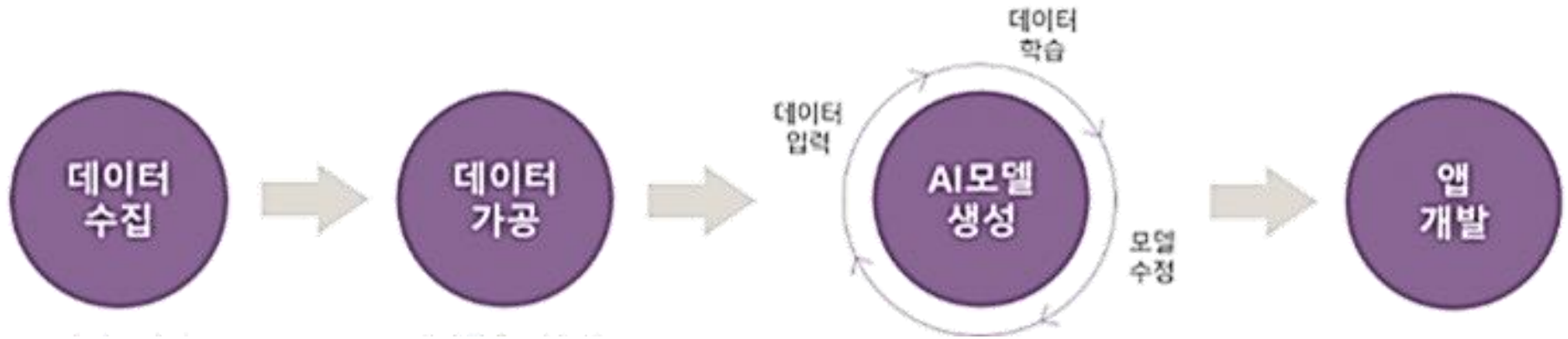
TRADITIONAL DEV CYCLE



AI DEV CYCLE



인공지능 SW 개발 과정



- 인공지능 서비스를 위한 인공지능 모델 생성 과정
 - 관련 데이터 수집
 - 데이터 가공
 - 인공지능 모델 생성
- 생성된 모델 기반 앱 개발



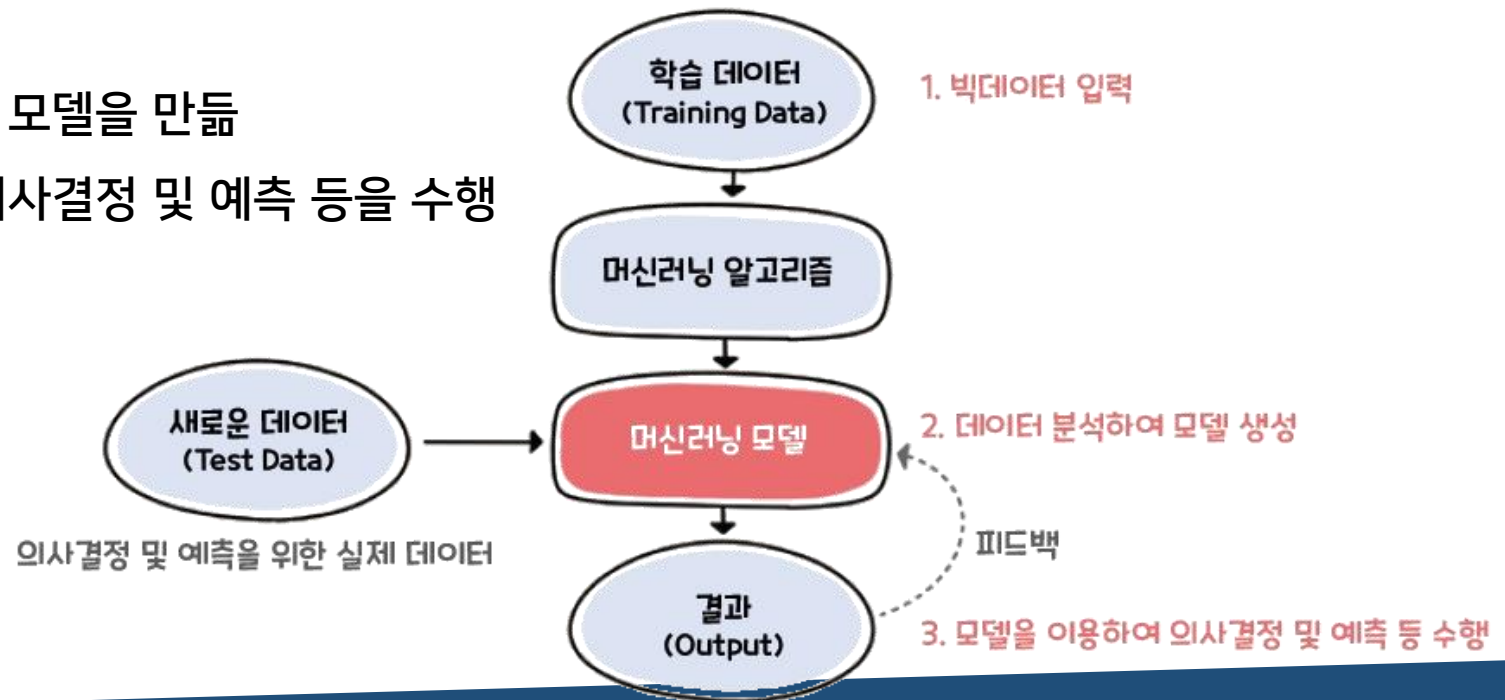
인공지능 머신러닝

- 머신러닝(ML: Machine Learning)

- 컴퓨터를 인간처럼 학습하게 함으로써 인간의 도움 없이도 컴퓨터 스스로가 새로운 규칙을 발견할 수 있도록 하는 기술.
- 머신러닝은 기본적으로 알고리즘을 이용해 데이터를 분석하고, 분석을 통해 학습하며, 학습한 내용을 기반으로 판단이나 예측을 함

- 머신러닝이 스스로 학습하여 데이터를 처리하는 과정

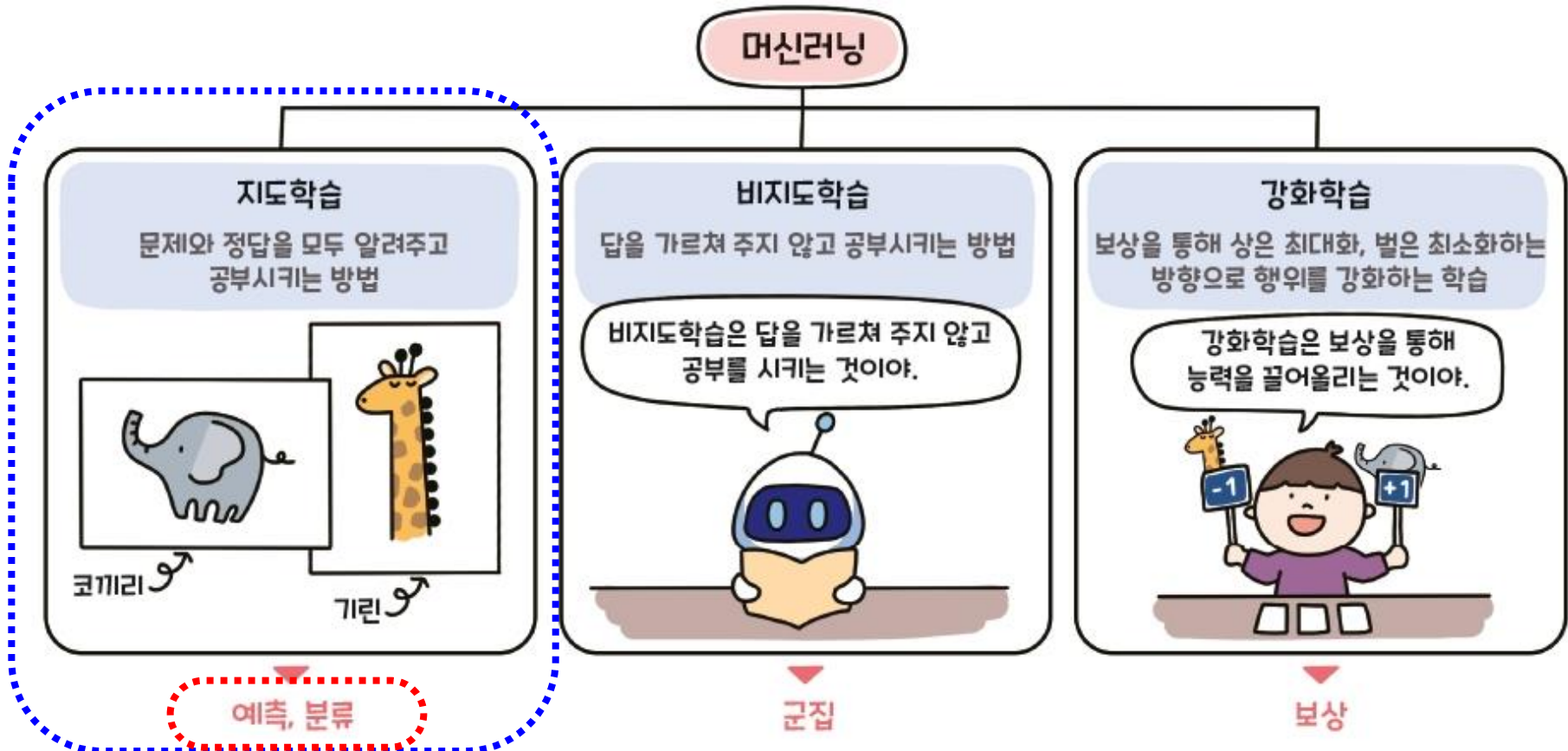
- 빅데이터를 입력
- 데이터를 분석하여 모델을 만듦
- 모델을 이용하여 의사결정 및 예측 등을 수행





머신러닝 분류

- 지도학습 : 예측이나 분류를 위해 사용
- 비지도학습 : 군집을 위해 사용
- 강화학습 : 환경에서 취하는 행동에 대한 보상을 이용하여 학습을 진행



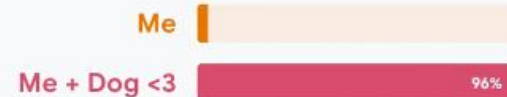
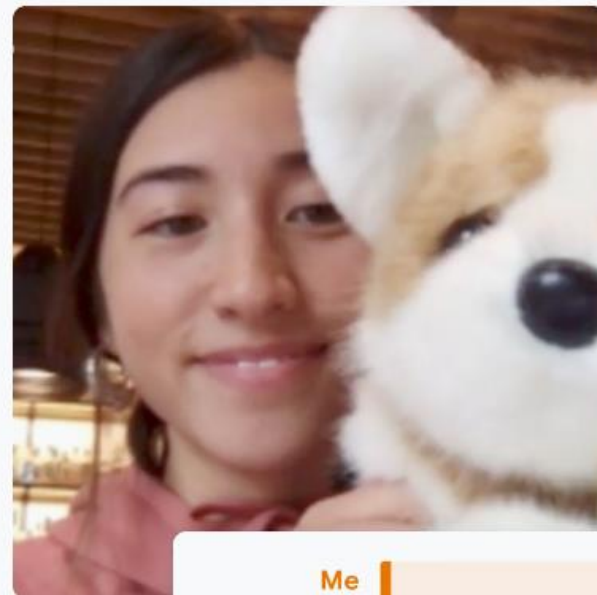


Teachable Machine

이미지, 사운드, 자세를 인식하도록 컴퓨터를 학습시키세요.

사이트, 앱 등에 사용할 수 있는 머신러닝 모델을 쉽고 빠르게 만들어 보세요. 전문지식이나 코딩 능력이 필요하지 않습니다.

시작하기






이미지분류 Model생성

Personal Image Classifier

1 Add Training Data 2 Select Model 3 Add Testing Data 4 View Results

Add label

X  강아지 (31 examples) 파일 선택

X  고양이 (20 examples) 파일 선택

Upload Model

Next



이미지분류 Model생성

Personal Image Classifier

1 Add Training Data 2 Select Model 3 Add Testing Data 4 View Results

Choose Model:

Create Model:

Convolution 7,7,256 → 3,3,5

Flatten 3,3,5 → 45

Fully Connected 45 → 100

Fully Connected → Number of Labels

Add Layer

Train model

Loss: 0.00904

Training Time: 00:00:13.311

Hyperparameters:

- Learning Rate:
- Epochs:
- Training Data Fraction:
- Optimizer:

Back

Next

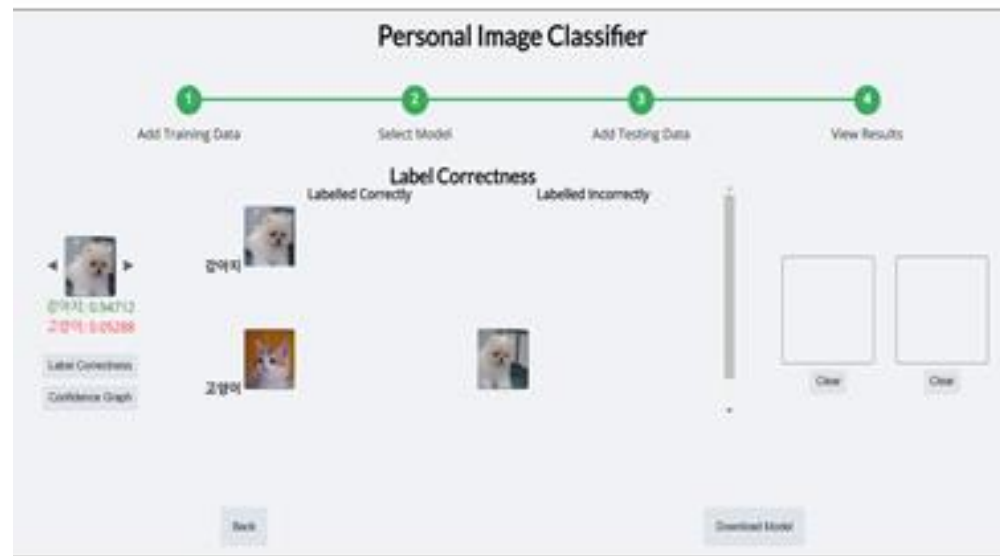




이미지분류 Model생성



이미지분류 Model생성



앱인벤터 개인 이미지 분류 인공지능 모델 학습



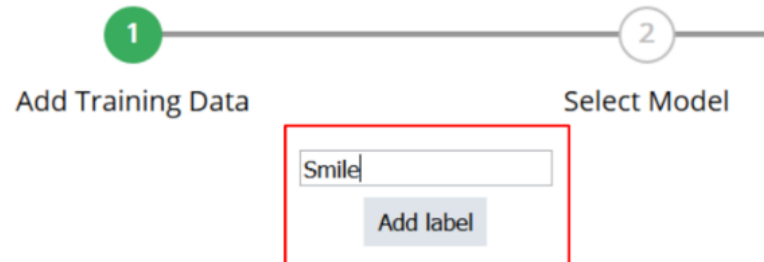
- <https://appinventor.mit.edu/explore/resources/ai/personal-image-classifier-part1>

브라우저를 열고 <https://classifier.appinventor.mit.edu/oldpic/> 으로 이동합니다.

기억하세요 - 모델은 아직 아무것도 모르기 때문에 훈련부터 시작해야 합니다. 진행 상황 표시줄을 통해 진행 상황을 추적할 수 있습니다.



2번째 라벨 추가 상자를 사용하여 모델이 학습할 수 있도록 표정을 추가하세요(창의적으로 하세요!). 처음 시작할 때는 두세 개의 표현만으로 모델을 만들어 보세요. 하지만 원하는 만큼 추가할 수 있습니다.

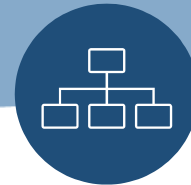
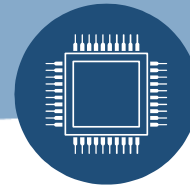


이제 이 라벨들에 대한 예시를 제시할 수 있습니다. 레이블을 클릭하고 웹캠 앞에서 포즈를 취한 다음 "예시 추가"를 클릭하여 모델이 배울 수 있는 이미지를 해당 레이블에 추가합니다(일반적으로 더 좋지만 나중에 다시 방문할 수 있습니다).

※ 참고 웹 도서 : 더북 모두의 앱 인벤터 <https://thebook.io/080226/0372/>

Thank you!

See you next time.



담당교수 : 유 현 주
comjoo@uok.ac.kr

