

서비스가 구성되기 위해 쌓이는 리소스들

업무 과정에서 생기는 문서들은
회의나 토론, 자료 조사, 요구사항 정의, 기능 정의, 산출물, 테스트 케이스 등
으로 기능을 구현하고
사용자에 맞춰서 서비스의 비즈니스 로직을 정의하며
구현된 각 기능들을 사용해 사용자 경험으로 연결한다



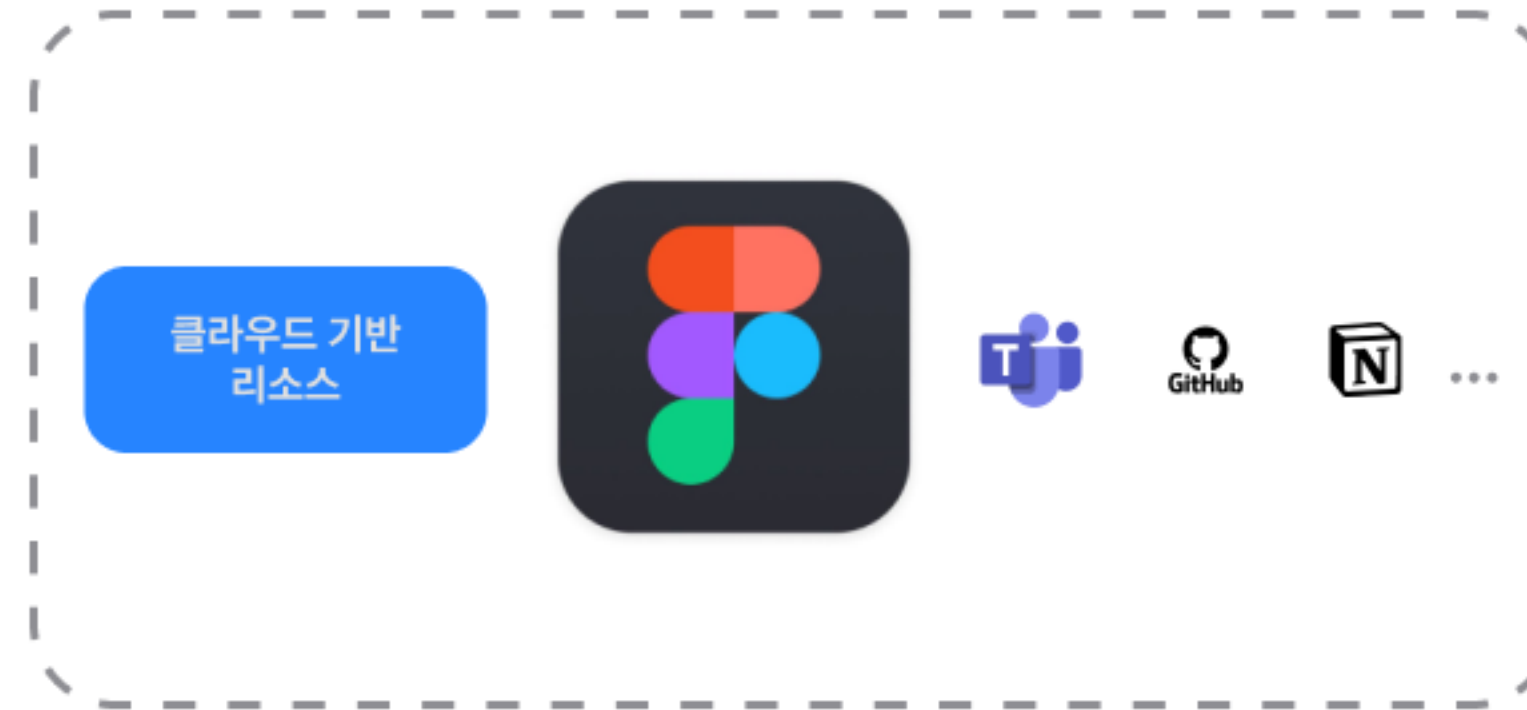
문제 해결 과정에서 생성되는 문서나 형식은 기본적으로 자유로움

기술에 대해 경험이 있고 협업을 위한 설계 시스템을 다양한 경험을 기반으로 템플릿, 컨벤션 등을 설계하는게 이상적이지만
현실적으로 리소스 관리가 자율에 맞겨지는 경우가 많다

작업을 시작할 때, 다른 사람의 작업을 가져와야 할 때, 회의 할 때, 다른 작업을 병행해야할 때
업무 관련 된 정보들을 수집하는 것에 불편함을 경험함

내가 작업하는 환경 Figma, Notion, Github, Teams

업무 과정에서 생기는 문서는 피그마를 기반으로 일정을 확정하며
회의나 토론으로 기능을 관련 담당자들에게 기능을 전달하고
요구 기능에 맞춰서 비즈니스 로직을 정의하며
구현된 각 기능들을 사용해 사용자 경험으로 연결한다



문제 해결 과정에서 생성되는 문서나 형식은 기본적으로 자유로움

대부분의 리소스가 클라우드 기반으로 동작하고 기능 정의는 시각적으로 확정된 피그마와
회의나 사전 지식으로 기능을 작업자가 정의해서 알아서 작업하는 구성을 띄고 있음

**작업을 시작할 때, 다른 사람의 작업을 가져와야 할 때, 회의 할 때, 다른 작업을 병행해야할 때
업무 관련 된 정보들을 수집하는 것에 불편함을 경험함**

문서를 모으자 Figma 에

모든 기능이 정의된 후 작업하는 것은 워터폴 방식에 가까우며
많은 시간이 필요하다

스프린트 방식으로 서비스를 설계하고 있으며 다양한 외부 요인으로 인해
기능들이 자주 변경되는 상황이다

클라우드 기반
리소스



...

로컬 파일 기반
리소스



...

로직과 요구 기능이 가장 많이 보이면서 가장 많은 접점이 일어나는 곳은 피그마
가장 많은 계층과 기획이 들어간 구성이 존재하며 의사 결정도 피그마에서 일어난다
로직 정의도도 피그마 기반으로 보는게 가장 빠르다

다른 리소스들은 대부분 URL로 복사 할 수 있는 환경이고 API를 통해 내부 데이터를 가져오는 게 가능하고
로컬도 URI 스키마 기반으로 연결되는 것이 있었다

피그마에 들어난 로직을 베이스로 문서를 작성하자

Figma 구성에 의미 부여하기

시스템과 컨벤션

시스템

기존에도 피그마에서 디자인 시스템이 있다

디자인 시스템 최소단위를 구성하는 => 컬러칩, 텍스트 스타일, 아이콘, 디바이스 호환 변수, 사이즈 컨벤션
기능 구성할 때 쓸 수 있는 => 기능 컴포넌트 (Input, buton 등)
페이지를 구성하는 => 레이아웃 컴포넌트 (헤더, 푸터, 사이드바, 네비 등)

그리고 메모를 적는 기능 역시 존재한다
에디터가 메모를 추가할 수 있는 => 컴포넌트, 컴포넌트 셋, 로컬 변수, 로컬 스타일, 프로토 타입
데브 모드에서 플러그인으로 접근 할 수 있는 => dev resource, plugin data, local data

계층에 의미 부여하기

파일 => 도메인 단위나 서비스 단위로 명칭
페이지 => 디자인 시스템을 구성하는 페이지로 사용하거나, IA 에서의 메뉴 단위로 정의하거나, QA 로직 정의로 쓰거나, 카테고리 묶거나 컴포넌트를 정의하거나 등등 용도 별 정의
섹션 => 경로로 쓰거나 카테고리 묶는 등으로 정의
컴포넌트 => 재사용 에셋 정의에 사용

피그마 기능과 연결

컴포넌트가 정의될 경우 이름은 [파일 이름 / 페이지 이름 / 섹션 이름 / 컴포넌트 셋 이름] 으로 호출된다

페이지 이름과 IA 를 연결할 경우 그 페이지의 이름은 고유한 것으로 취급할 수 있다
같은 페이지의 이름으로 여러 번 생성할 수 있고
같은 섹션 이름으로 여러번 생성 할 수 있다

원하는 만큼 쪼개더라도 이름을 기반으로 하나의 식별자를 구성할 수 있고
섹션이나 페이지, 컴포넌트 등의 피그마 시스템의 고유 아이디를 기반으로 대상의 식별자를 사용할 수 있다

WBS나 지라 등에서 각각의 기능 정의에 고유한 아이디를 부여하고 관리하는 개념이 존재함

파일 > 페이지 > 내용 의 순서로 계층 구조로 데이터의 소속을 참조하고 있다

고유 식별 할 수 있는 값을 저장하고 그 값을 기반으로 데이터를 넣어서 각 리소스를 고유 식별 할 수 있고 탐색할 수 있다
정보를 저장하는 서비스는 이미 고유식별 구조가 정의되어있다

적절한 계층에 메모를 저장하고 볼 수 있게 하자

실험을 통해 움직이고자 하는 핵심 KPI

정보 탐색 소모 시간 감소
정보를 찾기 위해 필요한 필요 지식 양, 누락 확률 감소

실험을 통해 얻고자 하는 궁극적인 배움 및 목표

문서화를 도입하지 않는 것은 그것을 감독하거나 관리하는 시스템을 유지하고 교육하고 운영하는 것 역시 비용이기 때문임

비즈니스 로직이나 기획을 기반으로 관련 자료들을 정렬하는 것은 원활하고 장기적인 서비스 관리 측면에서 중요하다
사람을 기반으로 시스템이 영원히 유지될 수 없기 때문

최소한의 리소스로 점진적으로 문서 기반의 업무 처리 프로세스와 커뮤니케이션을 회사에 정착시켜나가는 것과
최소단위로는 내가 생산하는 업무 리소스를 로직에 정렬하는 것을 목표로 함

피그마 계층과의 연동

페이지 기반, 또는 페이지와 섹션 이름을 기반으로 페이지에 등록된 메모 만 보는 인터페이스
계층 자체도 도메인을 표현하기 때문에 도메인에 연결된 데이터도 알 수 있게 한다

메모 분류 기능

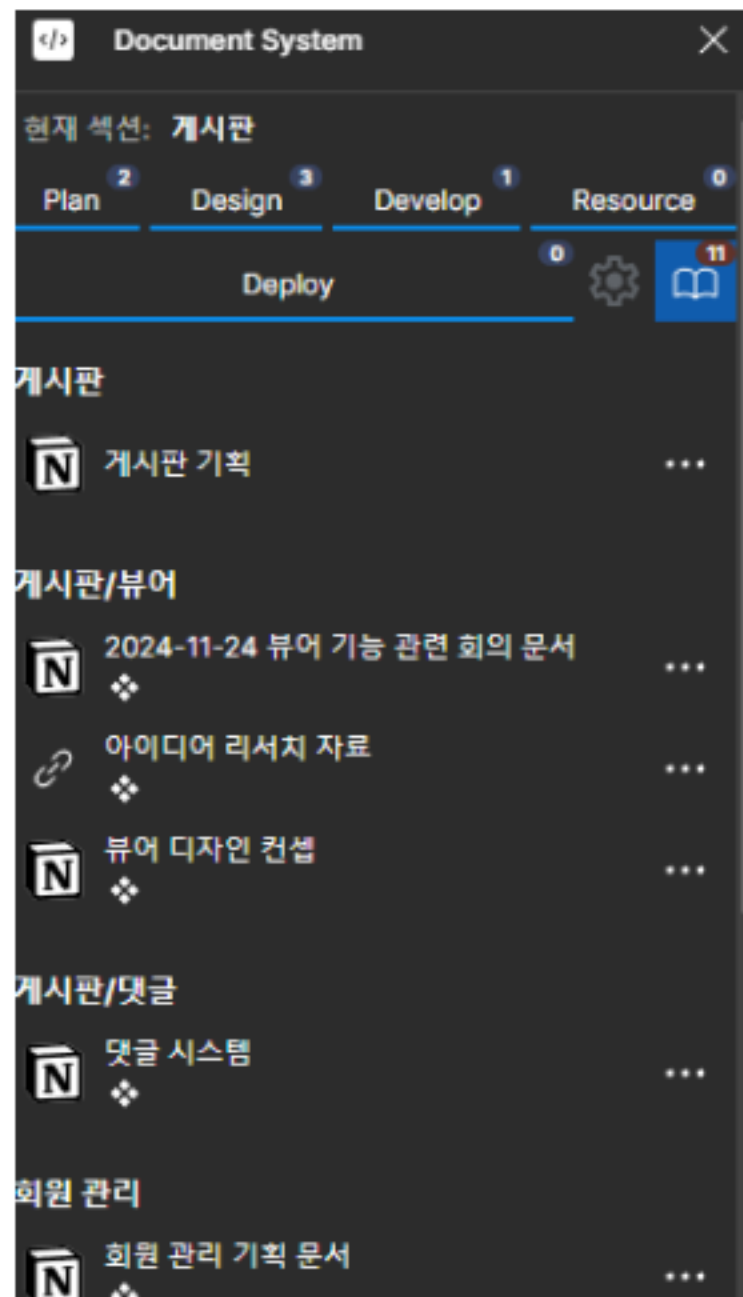
1. 계획 (plan)
 - 기획이나 요구 사항 정의한 것들에 대한 링크
2. 기능 메모 (design)
 - 기능 개발이나 디자인 등을 할 때 얻어지는 레퍼런스나 기술 조사 메모 등에 대한 링크
3. 산출물 경로 (resource)
 - 작업을 했으면 피그마 또는 코드나 , 기타 작업물들이 있는데 그 작업물에 대한 링크
4. 프로덕트 링크 (deploy)
 - 작업을 했고 코드가 나오면 배포를 하는데 그 배포된 곳에서 작업물을 볼 수 있는 링크

피그마 위치 포커스 기능

메모 분류에서 3번 기능의 겨우는 피그마에서 피그마로 포커스하는 것도 가능하다
이러한 맥락과 유사하게 메모와 그 메모가 서술하는 대상에 포커스하는 기능을 말함
메모에 서술 대상 피그마 요소를 선택해서 저장하는 것으로 구성

식별자 부여 기능

임의로 계층을 부여해야할 때가 있음
페이지/섹션 기준으로 메모 관리가 성립할 때 가능
임의 계층의 메모를 공유 받도록 설정



기대 효과

정해진 기준 (피그마) 기반으로 필요한 문서에 쉽게 접근 할 수 있음
문서가 없는 것을 빨리 파악하고 아무나 문서를 추가하며 작업할 수 있음
(dev mode 에 있어도 사용 가능)

피그마 계층과의 연동

컨벤션을 지켰을 때 그 활용이 더 빨리 와닿는다
피그마 숨은 꿀기능인 섹션을 더 잘 사용할 수 있게 된다
(Frame 123456 , 같은게 줄어듦)

메모 분류 기능

- 1 번의 계획은 모두에게 공유되어야 할 것
- 2 번의 기능 정의는 업무 관련자를 위해
- 3 번의 산출물은 관리자나 작업자를 위해
- 4 번의 프로덕트 링크는 QA와 버그 파악 위해

피그마 위치 포커스 기능

빠르게 관련 기능을 파악 할 수 있음
계층 식별자와 컴포넌트가 연결되는 것이기 때문에
식별자 기능과 연동할 수 있음 (복잡해지지 않도록 기능 정의를 단순하게 할 필요가 있음)

식별자 부여 기능

대상 객체에 식별자를 부여할 경우
피그마 위치 포커스 기능과 연동해서 관련 메모를 얻도록 구성할 수 있음



자세한 시스템 설계 필요함

구조적으로 봤을 때 계층 기반 메모 루트 식별자 추출은 페이지 / 섹션 단위임
 자잘한 컴포넌트들에 고유 식별자를 부여할 수 있게 되면 너무 많은 연결이 발생함으로 최소 메모 저장단위를 페이지와 섹션으로 잡았음

포커싱은 프레임 등도 잡을 수 있음
 특정 메모가 특정 컴포넌트에 대한 서술일 때를 위해 구성됨

식별자 부여는 루트 식별자를 임의로 설정할 수 있다는 기능이고
 섹션에 쓰라고 있는 기능임
 QA할 때 아니면 쓸 일 없을 것으로 보여짐