bb 안내는 사후 수단입니다.
사용자가 생각하는 대로 행동했을 때 기대하는 걸 제공해 주는 것이 우리의 역할입니다. 사람들은 문이 안 열릴 때 비로소 손잡이 근처에 뭐라고 적혀 있는지 읽는다는 걸 잊지 마세요. 99

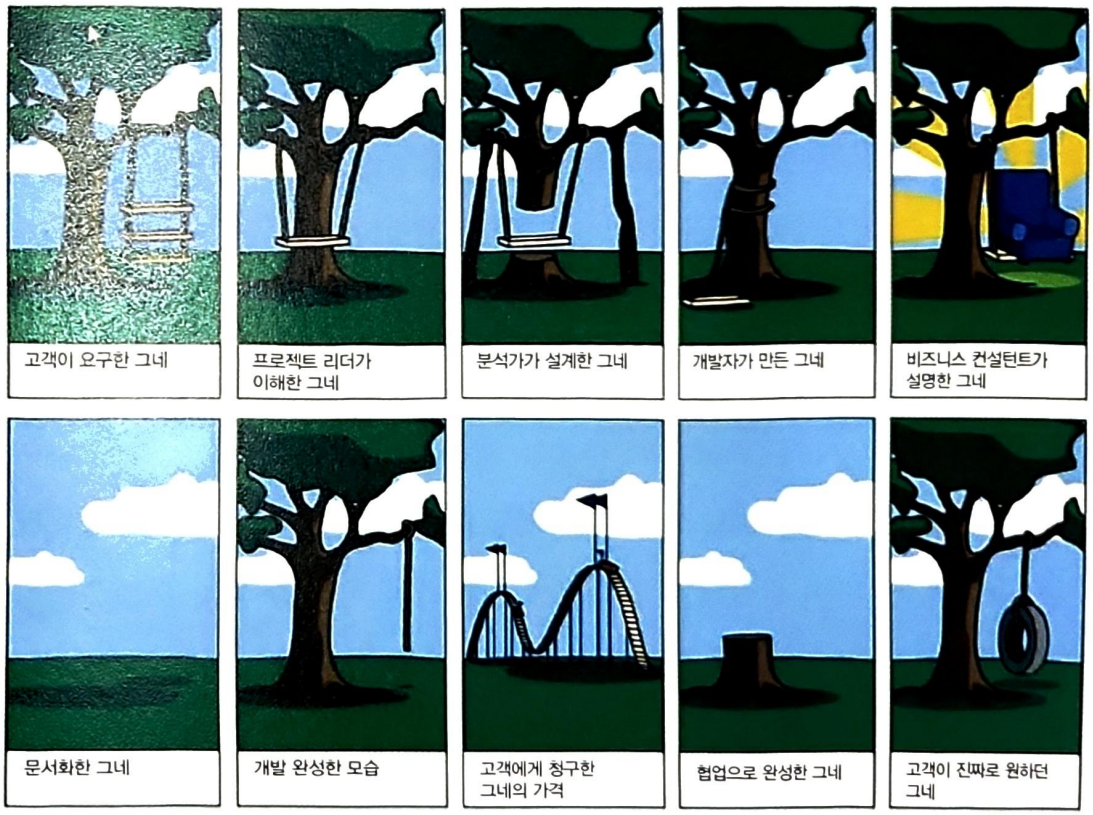
_프로덕트 디자이너, 이진재

5-1 피처를 구체화하고 팀을 일치시키는 유저 스토리
5-2 논리적 흐름을 그리는 유저 플로 & 와이어프레임
5-3 개발자에게 사랑받는 서비스 기획서 작성하기
5-4 운영까지 고려하는 백 오피스 기획하기
5-5 사용자가 길을 잃지 않는 '정보 구조' 구성하기
5-6 피처의 완성도를 높이는 QA와 테스트 케이스

5-1 피처를 구체화하고 팀을 일치시키는 유저 스토리

유저 스토리

제품 · 서비스를 개발하는 조직은 PM, 개발자, 디자이너 등 서로 다른 업무를 맡은 다양한 직군의 사람들로 구성됩니다. 따라서 하나의 피처에 대한 생각이 서로 다를 수 있습니다. 가령 다음 그림과 같이 '나무에 매단 그네'라는 사용자의 요구에도 담당자마다 생각하는 해결 방법이 무척 다양할 수 있기 때문에 '그네'의 범위와 생김새, 기능 등을 팀원들이 함께 정의하고 구체화하는 과정이 필요합니다.



하나의 요구 사항을 두고 여러 이해 관계자의 입장 차이를 보여 주는 예시

이런 상황에서 필요한 것은 팀원 간의 **컨센서스**consensus입니다. 컨센서스는 '의견 일치'나 '동감'을 뜻하는 단어인데, 실무에서는 주로 팀원 간의 합의를 뜻합

니다. **유저 스토리**user story는 컨센서스를 이끌어 내는 데 유용한 도구입니다.

유저 스토리란 쉽게 말하면 하나의 피처 단위를 설명하는 짧고 간결한 문장을 뜻합니다. 앞서 As is와 To be로 도출한 피처는 대부분 추상적인 형태를 띠고 있다가 기획과 디자인을 거쳐 구체화되기 시작합니다. 애자일 방법론을 적용한 조직에서는 유저 스토리를 이용해 구현할 피처들의 목록을 작성하고, 각 유저 스토리가 어느 정도의 복잡도와 범위로 구현되어야 하는지에 대해 내부 인원이 논의해서 효과적으로 컨센서스를 이끌어 낼 수 있습니다.

용어 사전 애자일

애자일agile은 워터폴의 단점을 보완하기 위해 등장한 개발 방법론으로 '포괄적 문서보다 작동하는 소프트웨어를, 계약과 협상보다 고객과의 협력을, 계획을 세우는 것보다 변화에 대응하는 것'을 추구합니다. 주로 제품 · 서비스를 빠르게 개발한 후 시장의 검증이 필요한 스타트업에서 많이 사용합니다.

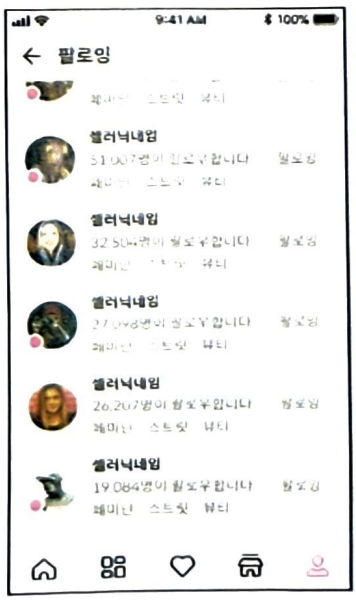
유저 스토리는 짧은 문장이지만, 다음의 3가지 요소를 포함해야 합니다.

- **AS**: 사용자
- **I WANT**: 사용자의 행동(피처)
- **SO THAT**: 그것을 함으로써 얻는 이득(니즈 · 목적)

AS는 사용자, I WANT는 사용자가 하려는 행동, 즉 피처를 뜻합니다. 그리고 SO THAT은 사용자가 I WANT에서 한 행동으로 얻으려는 결과입니다. 예시로 '볼라'의 퍼소나였던 김나래를 대상으로 유저 스토리를 작성해 보겠습니다. 먼저 '구글 계정으로 로그인하기'라는 피처의 유저 스토리를 살펴보면 AS(사용자)에 '김나래', I WANT(사용자의 행동, 피처)에는 '구글 계정으로 간편 로그인', SO THAT(니즈 · 목적)에는 '빠르게 로그인하기 위해서'가 들어갑니다.

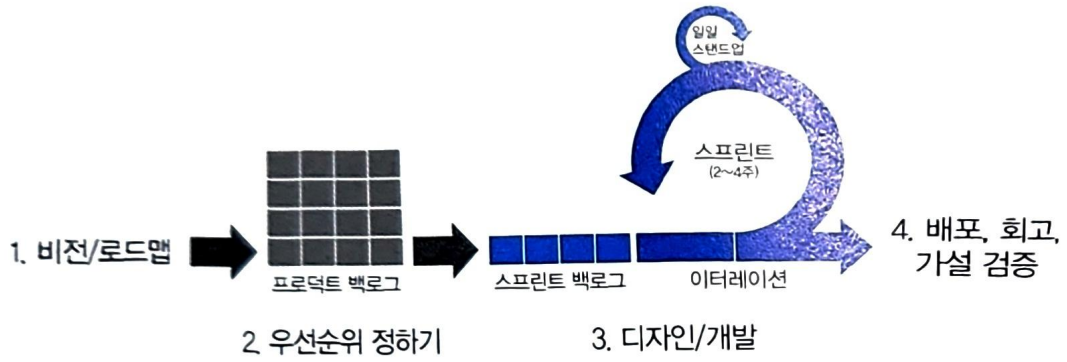
- AS: 김나래
- I WANT: 구글 계정으로 간편 로그인
- SO THAT: 빠르게 로그인하기 위해서

이렇게 짧은 문장으로 구성된 유저 스토리로 특정 기능을 서술할 수 있습니다. 예를 하나 더 들어 보죠. 팔로우한 셀러가 라이브 방송을 시작하면 푸시 알림을 보내는 피처를 개발한다고 가정해 보겠습니다. 마찬가지로 AS에는 퍼소나가 들어가고, I WANT에는 '셀러를 팔로우하여 방송 시작 알림 받기'가, SO THAT에는 사용자의 목적인 '셀러의 방송을 놓치지 않기'가 들어갑니다.



- AS: 김나래
- I WANT: 팔로우한 셀러의 라이브 방송 알림 받기
- SO THAT: 팔로우한 셀러의 방송을 놓치지 않기

실무에서는 여러 개의 유저 스토리 중 다음 스프린트에 구현할 피처를 선정하고 유저 스토리의 구체적인 범위와 복잡도를 논의하곤 합니다. 앞서 '01장 사용자 중심 기획'에서 살펴봤듯이 애자일 방식으로 개발하는 조직에서는 앞으로 개발할 피처들을 백로그backlog로 관리하면서 이 중 우선순위가 높은 피처를 선정해 2~4주간의 스프린트 과정을 통해 개발합니다. 이 과정을 그림으로 표현하면 다음과 같습니다.



애자일 제품 개발 프로세스

먼저 조직의 비전과 목표에 따른 프로덕트 로드맵을 잡은 다음 쌓아 둔 백로그(피처 더미)에서 우선순위가 가장 높은 피처를 선택합니다. 그런 다음 2~4주간 개발하는 과정(스프린트)을 거치는데 이때 피처 하나하나가 바로 유저 스토리가 될 수 있습니다. 조직의 개발 문화에 따라 하나의 피처를 여러 개의 유저 스토리로 쪼개는 경우도 있습니다.

스프린트로 들어가는 유저 스토리

유저 스토리를 통해 피처를 개발하는 과정은 크게 'To Do(할 일) → Work in progress(진행 중) → Done(완료)'으로 나뉘고 마지막엔 테스트를 거쳐 배포까지 이릅니다.

쌓여 있는 백로그들은 우선순위에 따라 다음 스프린트에 배치됩니다. 이때 하나 또는 그 이상의 유저 스토리로 조금 더 큰 단위의 피처나 프로젝트를 완성하는데요. 이를 **에픽**epic이라 합니다.

Tip. 스프린트에서 스토리 할당 업무는 주로 PM이나 PO 역할을 맡은 팀원이 담당합니다.

유저 스토리의 집합, 에픽

유저 스토리를 상세하게 작성하는 법, 걸킨

걸킨gherkin은 유저 스토리를 작성할 때 활용할 수 있는 일종의 구문 양식으로, 애자일 문화가 자리잡은 조직에서 종종 사용하는 방법입니다. 앞서 살펴본 유저 스토리의 3가지 구성 요소(AS, I WANT, SO THAT)만으로 피처가 정확하게 구현되었는지 판단하기가 어렵다면 사용자의 행동과 결과까지 기술하는 걸킨 방식의 유저 스토리 작성법이 도움이 됩니다. 작성 양식은 다음과 같습니다.

① 제목(title) : {사용자}는 {작업}을 할 수 있다.		우선순위(상, 중, 하)
② 유저 스토리 (background)	• AS(사용자) • I WANT(행동) • SO THAT(니즈 · 목적)	
③ 구현에 대한 기준 (acceptance criteria)	• Given(주어진 상황) • When(조건 및 행동) • Then(결과) * 한 문장당 하나의 행동을 작성하고 추가 행동은 or나 and로 연결한다.	
④ 비고(notes)	기타(팀원 또는 프로젝트 구성원이 알아야 할 사항)	

먼저 ① 제목은 '~은(는) ~을(를) 할 수 있다.'라는 식의 간결한 문장을 사용합니다. ② 유저 스토리는 원래 방식대로 AS, I WANT, SO THAT으로 작성합니다. 이때 유저 스토리는 피처의 배경 역할을 합니다. ③ 구현에 대한 기준은 유저 스토리가 정확하게 구현됐는지 판단하는 기준이 됩니다. 유저 스토리와 마찬가지로 3가지 요소로 구성되는데, 각 요소는 주어진 **상황**(Given), **조건 및 행동**(When) **결과**(Then)입니다. 구현에 대한 기준을 기술할 때는 한 문장에 하나의 행동을 작성하고, 추가 행동을 작성하고 싶을 땐 or나 and로 연결합니다. 단, or는 분기를 작성할 때만 사용하는 것이 좋습니다. '만약 ~한다면'의 형태로 추가 작성할 수 있습니다. 마지막으로 ④ 비고에는 팀원이 알아 두어야 할 내용을 작성합니다. 예를 들어, 디자인 링크를 첨부하거나 이전에 있던 결정 사항 등을 기록하기도 합니다.

또, 템플릿 오른쪽 상단에는 피처 우선순위를 적어 두기도 하는데 이때 우선순위는 팀원 간 합의를 바탕으로 결정해야 합니다. 이렇게 템플릿에 따라 칸을 채우면 걸킨 방식의 유저 스토리를 완성할 수 있습니다.

조금 복잡하게 느껴지겠지만 예시를 통해 더 배워 볼까요? 앞서 작성한 퍼소나를 이용해 걸킨 템플릿을 작성하면 다음과 같습니다.

① 제목(title): 김나래는 셀러 페이지에서 셀러를 팔로우할 수 있다.	우선순위: 중
② 유저 스토리 (background)	• AS: 김나래 • I WANT: 셀러를 팔로우하면 해당 셀러의 방송 시작 알림을 받을 수 있다. • SO THAT: 셀러의 방송을 놓치지 않기 위해
③ 구현에 대한 기준 (acceptance criteria)	• Given: 셀러 페이지에 김나래가 진입했을 때 • When: 셀러 프로필 오른쪽의 [팔로우] 버튼을 터치하면 • Then: 김나래는 셀러를 팔로우하는 상태가 된다.
④ 비고(notes)	버튼 UI 스타일은 피그마 링크를 확인해 주세요.

먼저 ① 제목에 사용자가 수행하는 태스크를 간결하게 한 문장으로 작성합니다. 그런 다음 ② 유저 스토리에는 '셀러의 라이브 방송 알림을 받기 위해 셀러를 팔로우한다.'라는 배경을 작성합니다. ③ 구현에 대한 기준에는 유저 스토리를 구현했을 때 이를 검증할 기준을 작성합니다. 사용자가 셀러 페이지에 진입했을 때(Given, 주어진 상황), 셀러 프로필 오른쪽에 있는 [팔로우] 버튼을 터치하면(When, 조건 및 행동) 해당 셀러를 팔로우하는 상태가 되고 이후 셀러가 라이브 방송을 시작하면 푸시 알림을 받는 것(Then, 결과)이 구현할 피처의 기준이자 최소한의 조건입니다. ④ 비고에는 팀원에게 공유할 내용을 메모 형태로 남겨 둡니다. 이렇게 걸킨 양식을 이용해 손쉽게 꼼꼼한 유저 스토리를 완성할 수 있습니다.

Memo 유저 스토리와 결단 행동항목을 퍼치 구체화하기

[illegible] 유저 스토리와 결단

[illegible] 유저 스토리를 작성해 프로젝트의 방향성을 정하는 [illegible] 이해 작성한 유저를 유저 스토리를 개발해 보세요. 우리는 WAN 등 THAT[illegible]을 근거한 유저 스토리를 작성해 보고 이를 토대로 결단 항목을 이용한 스토리텔링 개발해 보세요. 예시의 방향성을 살펴보는 것도 좋습니다. 단, 이미 작성한 유저 스토리는 있으, 결단에 맞춰 새롭게 작성한 유저 스토리도 있으니 [illegible]해서 프로젝트에서 사용해 보세요.

사용자 여정 지도

[illegible] 하는 과

[illegible] 하고 나

사용자 여정 지도 [illegible] 서비스

[illegible] 보여

[illegible] 작업하는 데 도움이 됩니다.

[illegible] 사용자가 제품 서비스를 이용하는 과정을

[illegible] 제품도 만족도 불편 사항, 니즈 그리고 개

[illegible] 성합니다. 이 방식이 정형화되진 않았

[illegible] 다양한 방식으로 시각화할 수 있습니

[illegible] 어떻게 작성하는지 예시를 살펴

[illegible] 사용자 여정 지도입니다.

[illegible] **유저 플로우** [illegible] **정보 구조**

[illegible]

유저 플로우 & 정보 아키텍처

유저 플로우 [illegible]

정보 아키텍처 [illegible]

[illegible] 세분화하여 [illegible] 분석을 [illegible] 세분성들을 종합적으로 [illegible] 세분성이 [illegible] 심네이드 작업을 시작하기 전에 [illegible] 방향을 [illegible] 위한 방도로 사용자 퍼널 시도를 시행합니다.

[illegible] 첫 번째 행인 유저 스토리에 살펴 작성한 유저 스토리들을 [illegible] 해 보세요. 그리고 그중 핵심 기능을 만들어내는 스토리는 조금 [illegible] 해 보세요. 사용자 퍼널 시도가 완성되면 핵심 기능 스토리들을 [illegible] 사용자들이 더 빠르게 경험할 수 있을지 개선 방향을 고민해 볼 수 있습니다. 또한 각 단계별로 사용자 만족도를 표시하여 만족도가 낮은 단계라면 [illegible] 사용자 [illegible]로 볼 수 있습니다. 이때 낮은 지점에서 사용자가

획, 불편한 지점과 니즈를 파악하고 어떻게 개선할지 모색할 수 있습니다. 단, 사용자 만족도는 앞서 리서치 단계에서 수집한 데이터를 활용한 분석 결과여야 합니다.

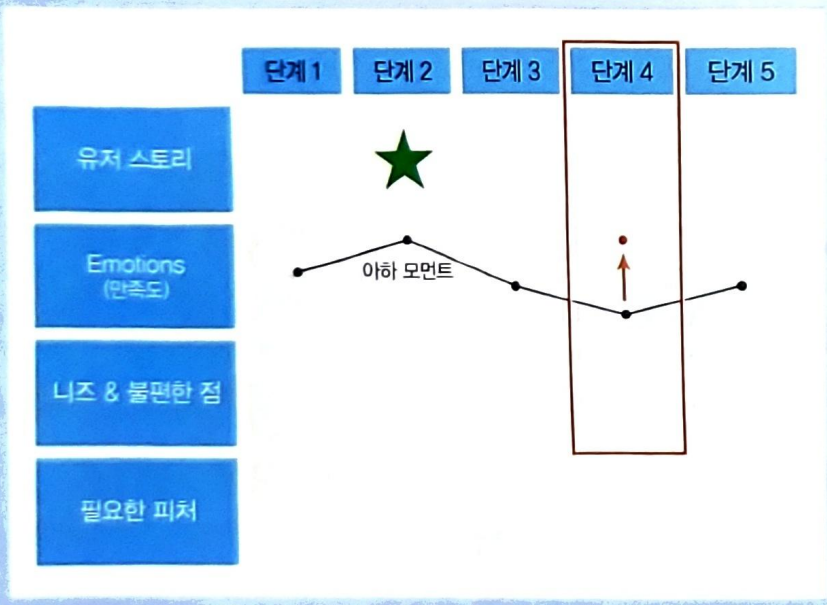
Mission **사용자는 우리 제품·서비스를 어떻게 이용할까?**

템플릿 : 사용자 여정 지도

지금까지 수집한 리서치, 정의해 둔 퍼소나 등을 기반으로 여러분의 제품 · 서비스의 사용자 여정 지도를 작성해 보세요.

상단의 각 단계에는 '1. 유입 → 2. 회원가입 → 3. 탐색 → 4. 주요 과제(구매 등)'와 같이 사용자가 제품 · 서비스를 이용하는 과정을 큰 단위로 구분해 입력하세요. 그런 다음 각 단계별 사용자의 만족도와 니즈, 불편 사항 등을 기입하면 전체 흐름을 한눈에 볼 수 있습니다. 맨 아래의 피처 행에는 각 단계를 거치는 데 필요한 피처를 입력하면 사용자 여정 지도가 완성됩니다.

Tip. 퍼소나, 사용자 시나리오, 사용자 만족도 등은 모두 리서치 단계에서 수집한 데이터를 기반으로 해야 합니다. 예측으로 작성해선 안 됩니다.



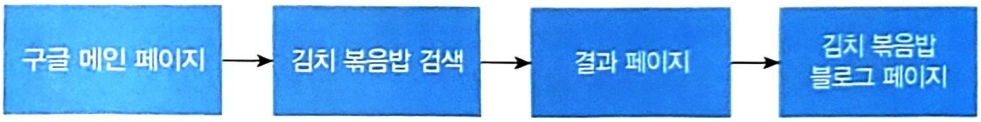
사용자 여정 지도

5-2 논리적 흐름을 그리는 유저 플로 & 와이어프레임

태스크 플로와 유저 플로

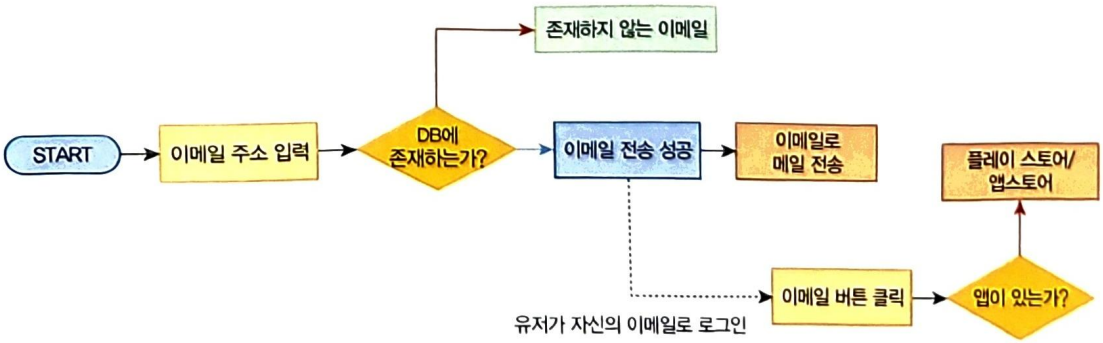
태스크 플로task flow와 **유저 플로**user flow는 사용자가 제품 · 서비스를 이용하면서 수행하는 작업과 화면의 흐름을 나타냅니다. 2가지 모두 흐름을 보여 준다는 역할은 같지만, 목적은 다릅니다.

먼저 태스크 플로는 분기나 오류를 고려하지 않고 큰 그림을 확인하는 것이 목표로, 높은 계층의 단계만 순서대로 나열합니다. 예를 들어 '김치 볶음밥 레시피 검색'하는 과정을 태스크 플로로 표현하면 다음과 같습니다.



김치 볶음밥 레시피를 구글에서 검색하는 과정. 태스크 플로

이 과정에 필요한 피처나 오류 등을 모두 생략하고 간단하게 사용자가 수행하는 작업 위주로만 나열된 것을 볼 수 있습니다. 반면 유저 플로는 분기, 오류 그리고 피처를 구현할 때 필요한 사항까지 고려해 작성합니다. 예를 들어 사용자가 입력한 이메일이 유효한지 확인하는 과정을 유저 플로로 표현하면 다음과 같습니다.

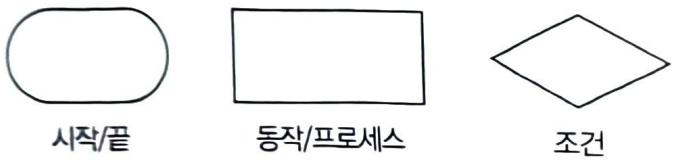


이메일 주소를 확인하는 과정에 대한 유저 플로

이메일 주소 확인하는 과정만 표현하는 데에도 다양한 분기 조건을 고려했음을 알 수 있습니다. 이처럼 유저 플로는 태스크 플로와 달리 모든 분기를 고려하고 과정마다 발생할 수 있는 오류를 예측해야 합니다. 숙달되지 않으면 작성하기 쉽지 않죠. 따라서 초보자라면 태스크 플로 수준의 단순한 플로부터 작성해 보고 이후 여러 가지 분기를 추가해 가며 플로의 복잡도를 늘리는 방향으로 연습하는 것이 좋습니다.

유저 플로 작성하기

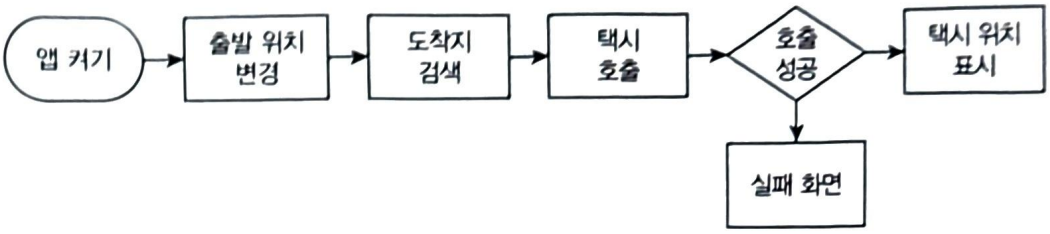
유저 플로를 작성할 때는 시작과 끝을 나타내는 둥근 사각형, 동작이나 프로세스를 나타내는 직사각형, 조건(분기)을 표현하는 마름모 3가지 기호를 주로 사용합니다.



유저 플로 기본 구성 요소

Tip. 유저 플로는 미로(miro.com)나 피그잼(figma.com/figjam)과 같은 도구를 사용하면 좀 더 쉽게 그릴 수 있습니다.

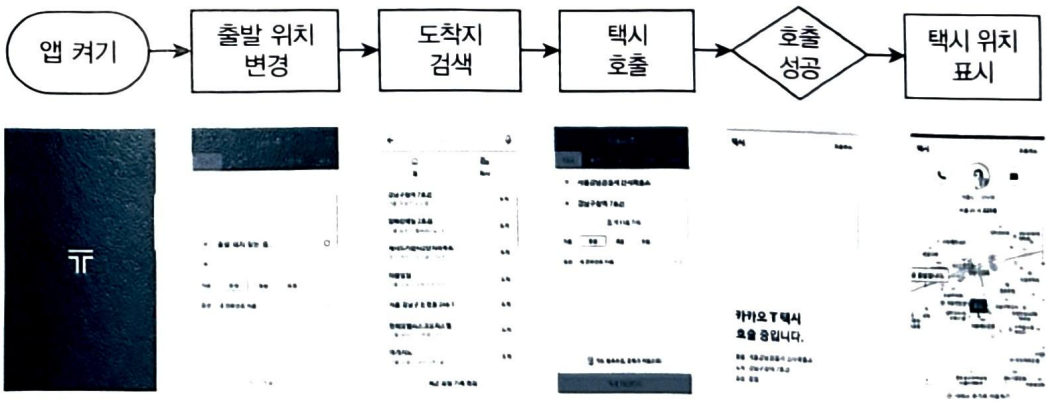
이 3가지 기호를 이용해 간단한 플로부터 복잡한 플로까지 난이도를 높여 가며 작성해 보면 도움이 됩니다. 예를 들어 '카카오 T'로 택시를 호출하는 과정을 유저 플로로 표현해 보면 다음과 같습니다.



'카카오 T'로 택시를 호출하는 유저 플로

먼저 시작을 뜻하는 둥근 사각형 기호에 '앱 켜기'를 담고 동작/프로세스를 표현하는 직사각형에 '출발 위치 변경 → 도착지 검색 → 택시 호출' 등 일련의 과정을 각각 담습니다. 그러다 분기, 즉 성공이나 실패가 나뉘는 지점은 마름모로 표현합니다. 물론 이 유저 플로는 이해를 돕기 위해 간단하게 작성한 것으로, 실제 구현할 때는 훨씬 많은 분기가 생길 수 있습니다.

유저 플로의 역할은 제품 · 서비스의 흐름을 보면서 우리가 고려해야 할 사용자 경험과 발생하는 다양한 경우의 수를 고려하는 것입니다. 이를 기반으로 이후 살펴볼 와이어프레임의 순서를 잡을 수도 있죠. 각 플로 아래 해당하는 와이어프레임(화면)을 배치하면 사용자가 어떤 흐름으로 제품 · 서비스를 이용하는지, 또 미처 생각하지 못한 플로나 오류가 있진 않은지 등을 구체적으로 기획할 수 있습니다.



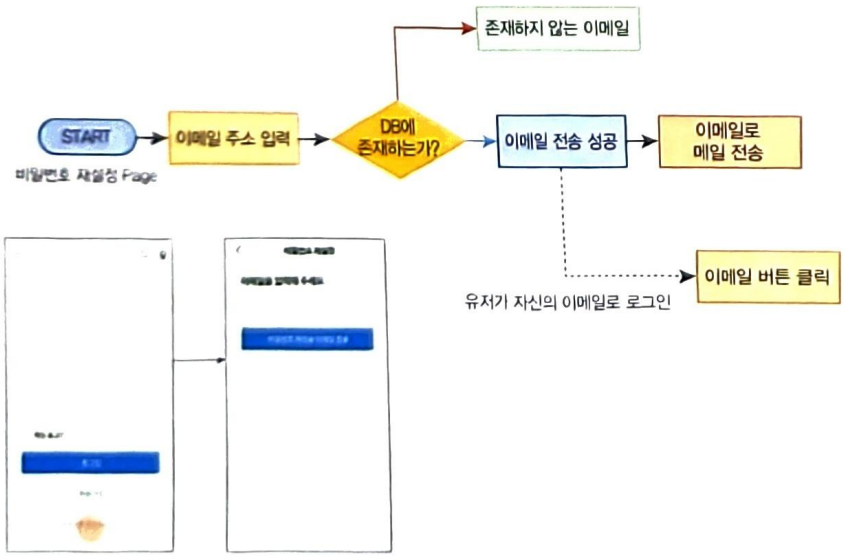
유저 플로 하단에 화면 배치

궁금해요 복잡한 유저 플로는 어떻게 작성하나요?

간단한 유저 플로는 앞서 살펴본 3가지 기호로도 충분하지만, 복잡한 유저 플로를 표현하는 데는 한계가 있습니다. 이때 더 다양한 조건을 표현할 수 있는 방법이 있습니다. 우선 필요한 도형은 3가지로 동일합니다. 여기에 Yes/No를 나타내는 화살표 2개를 더 사용하고 동작/프로세스를 표현하는 직사각형에 색을 사용해 구분하면 더 세부적인 로직을 표현할 수 있습니다. 가령 파란색 사각형은 '사용자가 위치한 페이지', 초록색 사각형은 '서비스에 뜨는 각종 모달 창', 주황색 사각형은 사용자의 클릭이나 입력과 같은 '사용자 행동', 빨간색 사각형은 시스템이 사용자에게 메일을 보내거나 푸시 알림을 보내는 등 프로그래밍으로 처리하는 '시스템 행동'으로 정의할 수 있습니다.

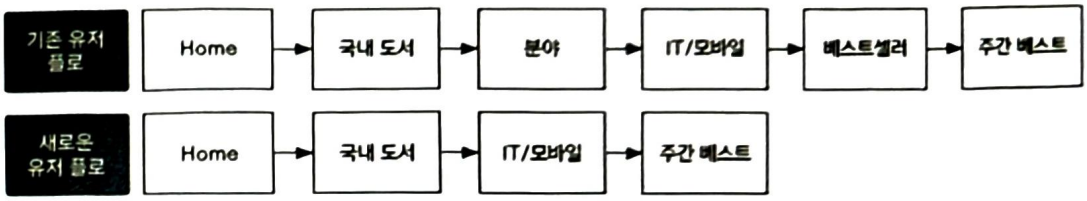
Tip. 모달 창은 사용자가 위치한 페이지에 추가 창을 띄워 이목을 집중시키는 또 다른 화면(레이어)을 가리킵니다.

이처럼 다양한 컴포넌트를 이용하면 보다 다양한 분기를 표현할 수 있어 복잡한 유저 플로도 작성할 수 있습니다.



비밀번호 재설정 유저 플로

유저 플로를 활용하면 태스크 단계를 줄여 문제를 해결하는 데 활용할 수 있습니다. 목적지는 같은데 경로가 많다면 비효율적이겠죠? 또, 도중에 사용자가 이탈하거나 오류가 발생할 확률도 높아집니다.



온라인 서점에서 분야별 주간 베스트셀러를 보는 유저 플로

이처럼 불필요한 단계를 줄임으로써 사용자가 목적을 이루는 데 걸리는 시간을 줄이고 접근성을 높일 수 있습니다. 특히 회원가입이나 결제처럼 사용자의 이탈률이 높은 과정에선 단계를 줄이는 것이 사용자 행동 전환에 큰 영향을 끼치기 때문에 플로 개선을 고려하는 것이 좋습니다.

Mission 유저 플로 작성하기

평소 자주 사용하는 제품 · 서비스 또는 앞서 작성한 유저 스토리를 이용해 유저 플로를 작성해 보세요. 기존 제품 · 서비스의 유저 플로를 그린다면 전체를 한 번에 그리는 것보다 피처별로 나눠서 작성해 보는 것이 좋습니다.

와이어프레임

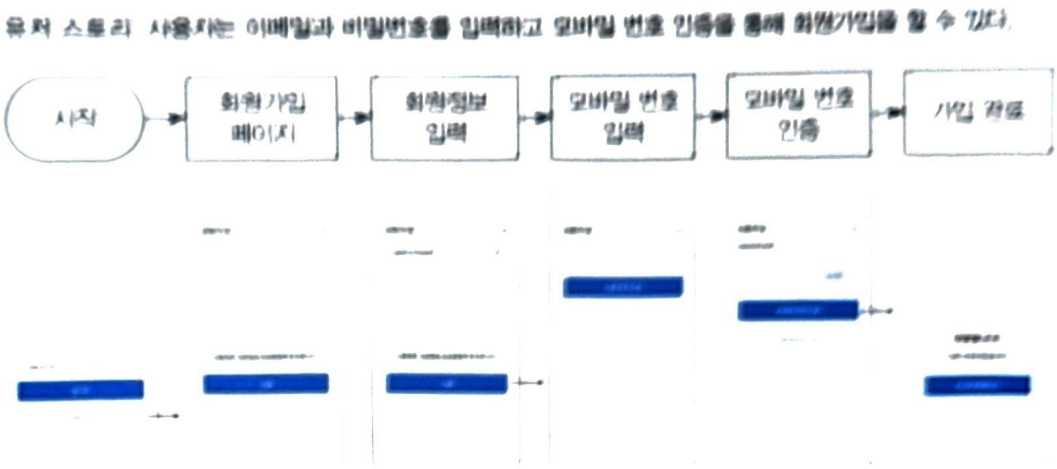
유저 플로를 작성하고 나면 이를 토대로 **와이어프레임**wireframe을 디자인합니다. 와이어프레임은 유저 스토리와 유저 플로를 시각적으로 표현하는 도구로, 마찬가지로 퍼소나를 기준으로 작성합니다. 그래야 방향성을 잃지 않고 사용자에게 필요한 피처를 구체화할 수 있기 때문입니다.

[illegible] 대해 구성 요소의 시각적인 중심도가 낮은 로우 피델리티로 [illegible] 피델리티지만, 필요한 요소들의 위치와 피처의 흐름은 모두 [illegible] 를 잘 잡아야 합니다. 그래야 이후 디자인 작업도 효율적으로 [illegible] 있죠. 최근에는 디자인 팀의 문화에 따라 와이어프레임 단계부터 [illegible] 가 높은 경우도 상당히 많습니다.

Tip 실무자의 선호에 따라 유저 플로보다 와이어프레임을 먼저 그리는 경우도 있고, 바로 하이 피델리티 와이어프레임을 작업하는 경우도 있습니다. 꼭 정해진 규칙은 없으니 효율을 높이는 방향으로 순서를 잡는 것이 좋습니다.

로우 피델리티 와이어프레임(왼쪽)으로 완성한 목업 화면(오른쪽)

와이어프레임을 연습하는 가장 좋은 방법은 유저 스토리 아래 해당 유저 플로를 작성하고 또 그 아래에 해당 페이지의 구성 요소를 그려 보는 것입니다. 예시로 회원가입 과정의 **유저 스토리 → 유저 플로 → 와이어프레임(로우 피델리티)**를 배치해 보겠습니다.



유저 스토리 → 유저 플로 → 와이어프레임(로우 피델리티)을 순서대로 배치

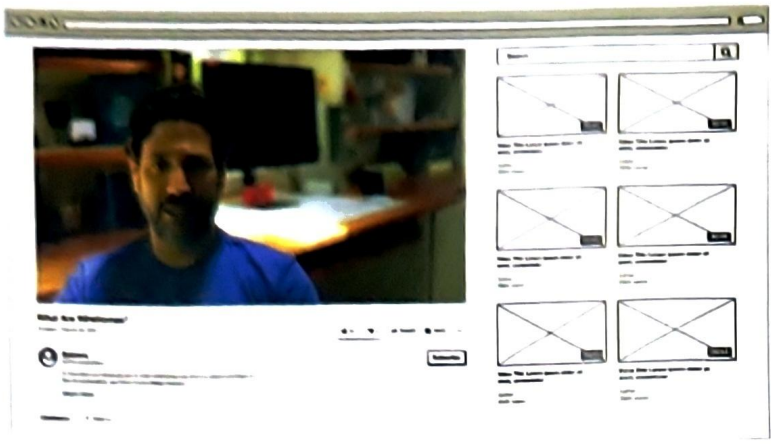
이렇게 기획 시 3가지 요소를 고려하면 각 유저 스토리의 플로와 화면 배치까지 한눈에 볼 수 있어 놓치는 부분을 최소화할 수 있습니다.

와이어프레임 도구

와이어프레임을 그릴 때는 펜과 종이를 이용해 손으로 그리기도 합니다. 하지만 여러 사람이 보고 쉽게 알아볼 수 있고 또 로우 피델리티부터 하이 피델리티까지 표현할 수 있는 디자인 도구를 사용하기도 하는데요. 와이어프레임 제작 시 사용할 수 있는 유용한 툴 몇 가지를 소개합니다.

발사믹

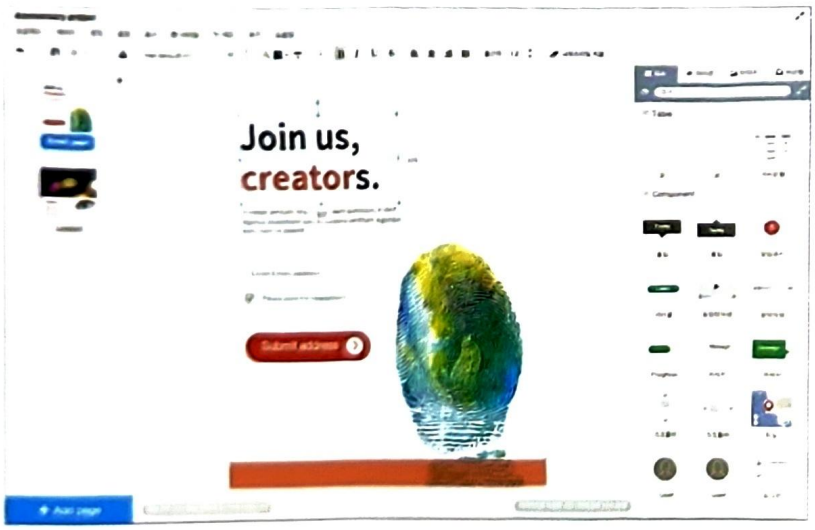
발사믹(balsamiq.com/wireframes)은 출시된 지 오래된 도구지만 웹 기반이라 접근성이 좋고, 발사믹에서 제공하는 컴포넌트를 마우스로 끌어와 배치만 하면 쉽게 그릴 수 있어 실무에서는 여전히 와이어프레임 제작에 많이 사용하는 도구 중 하나입니다. 발사믹은 펜으로 마치 스케치한 듯한 로우 피델리티를 제작할 수 있는 것이 특징입니다.



발사믹 공식 유튜브 채널

오븐

오븐(ovenapp.io)은 카카오에서 제공하는 서비스로 웹사이트에 가입만 하면 무료로 와이어프레임을 그려볼 수 있고, 웹과 앱 모두를 위해 사용할 수 있는 다양한 컴포넌트를 제공합니다. 사용 방법이 간단하기 때문에 로우 피델리티 와이어프레임을 빠르게 제작해 볼 수 있습니다.



오븐 공식 홈페이지

피그마

피그마(figma.com)는 전 세계에서 가장 많이 쓰이는 디자인 도구입니다. 발사믹과 마찬가지로 웹 기반 도구라 접근성이 높은 것은 물론이고 동시 협업이 가능해 팀원 간 소통에 특화되어 있죠. 사용성도 높은 편이라 디자인 도구를 다룬 경험이 있다면 쉽게 익힐 수 있고, 간단한 인터랙션을 적용하여 실제 사용자 테스트가 가능한 수준의 프로토타입까지 구현할 수 있습니다.

또, 사용자가 많은 만큼 데이터가 풍부해 피그마 공식 홈페이지의 [community] 메뉴에서 기획하는 제품 · 서비스의 와이어프레임 키트(kit) 및 디자인 프로젝트를 불러와 조합, 수정하여 와이어프레임을 손쉽게 제작할 수 있습니다.

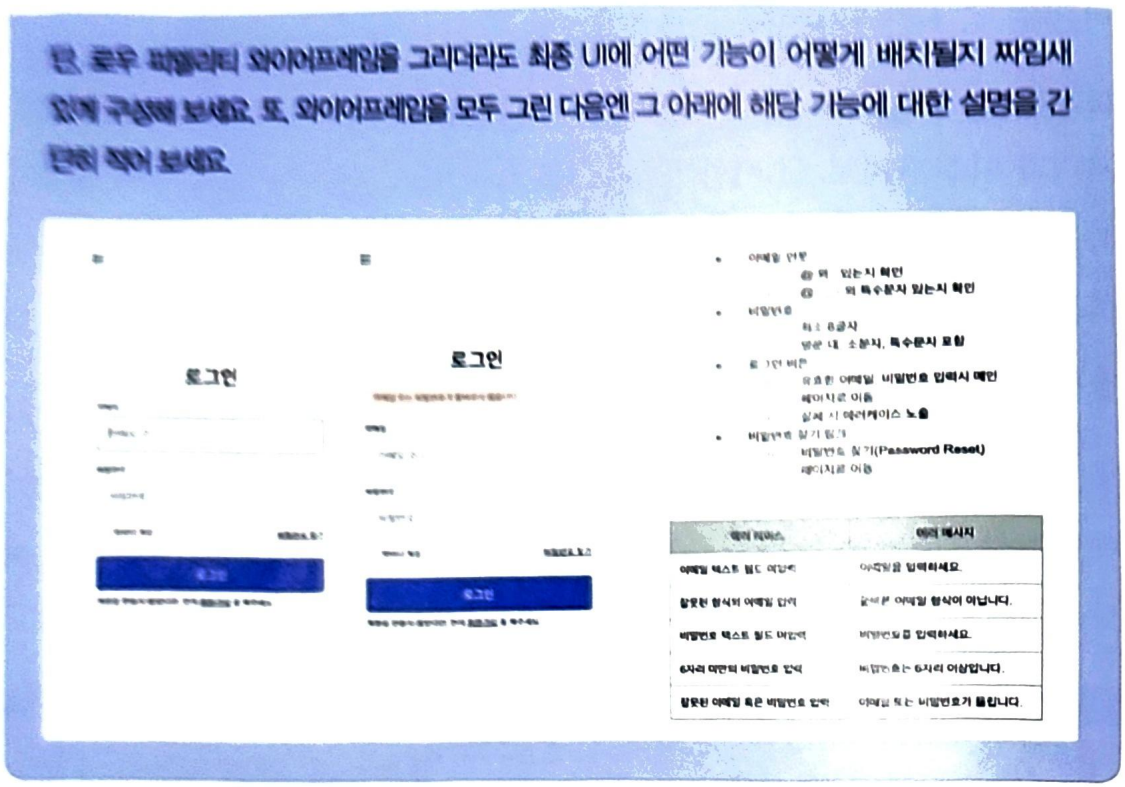
Tip. 다양한 제품 · 서비스의 와이어프레임을 참고해 자신만의 프로젝트를 기획할 수 있도록 피그마 템플릿을 제공하니 노션 템플릿(bit.ly/3FKikaN)을 참고하세요.

피그마 공식 홈페이지

Mission **와이어프레임 제작하기**

템플릿 : 와이어프레임

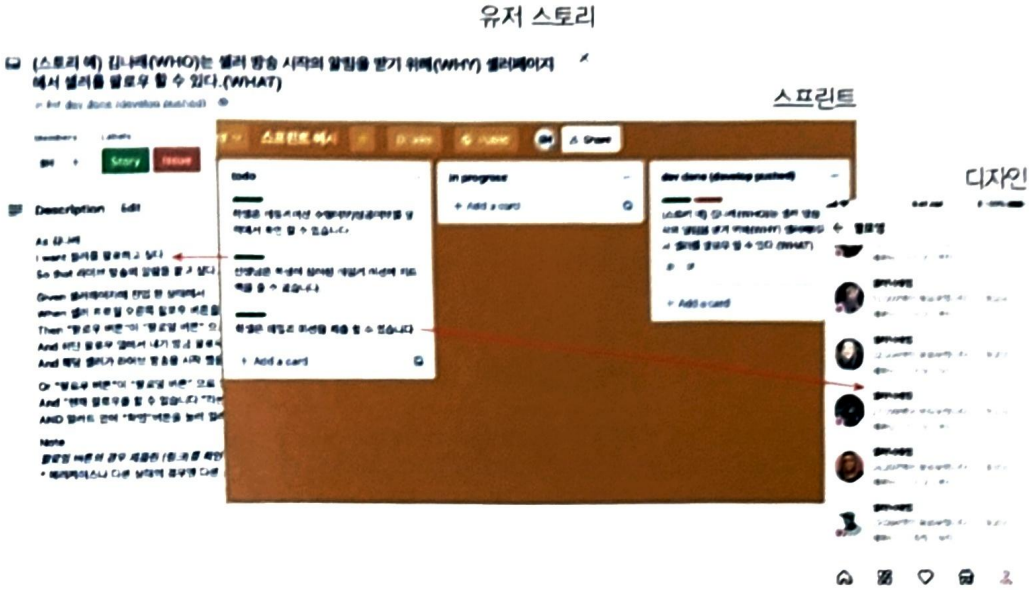
이전 과제에서 만든 유저 스토리를 기반으로 스토리에 해당하는 유저 플로와 와이어프레임을 그려 보세요. 펜과 종이를 들고 직접 손으로 그려도 좋고 앞에서 소개한 와이어프레임 도구 중 원하는 걸 이용해도 좋습니다. 또는 제공하는 템플릿 파일에 간편하게 써도 좋아요.



5-3 개발자에게 사랑받는 서비스 기획서 작성하기

소통을 위한 도구, 기획서

팀 문화에 따라 와이어프레임 단계를 건너뛰고 곧장 UI 디자인 단계로 넘어가기도 합니다. 이는 애자일 방법론이 널리 퍼지면서 기존 워터폴 방법론에서 주로 사용하던 파워포인트 형식의 기획서가 간소화되고, UI 디자인 도구와 시스템이 발전하면서 UI 디자인 작업 과정 또한 매우 빨라졌기 때문입니다. 덕분에 유저 스토리를 기반으로 피처의 범위를 논의하고 피그마와 같은 디자인 도구에서 바로 협업하는 경우가 점차 많아지고 있습니다.



간소화된 디자인 단계

하지만 제품 · 서비스를 구현하는 개발자 입장에서는 디자인만으로 상세 기능이나 의도를 파악하기 어려울 수 있습니다. 따라서 효율적인 의사소통 도구로 기획서를 작성하는 것도 디자인을 잘 설명하는 데 도움이 됩니다. 여기서 기획서란 화면의 구성 요소나 콘텐츠에 대한 설명, 화면 간 이동 흐름, 기능, 사용법 등을 기술한 문서를 뜻하며 **화면 설계서, 스펙 문서, 스토리보드** 등의 이름으로 부릅니다. 즉, 기획서라는 일관된 양식이 있는 것이 아니라 조직의 문화나 개발하는 제품 · 서비스에 적합한 형태의 기획서를 선택하면 됩니다. 제품 · 서비스에 따라 유형, 작성 방식 등도 다양합니다.

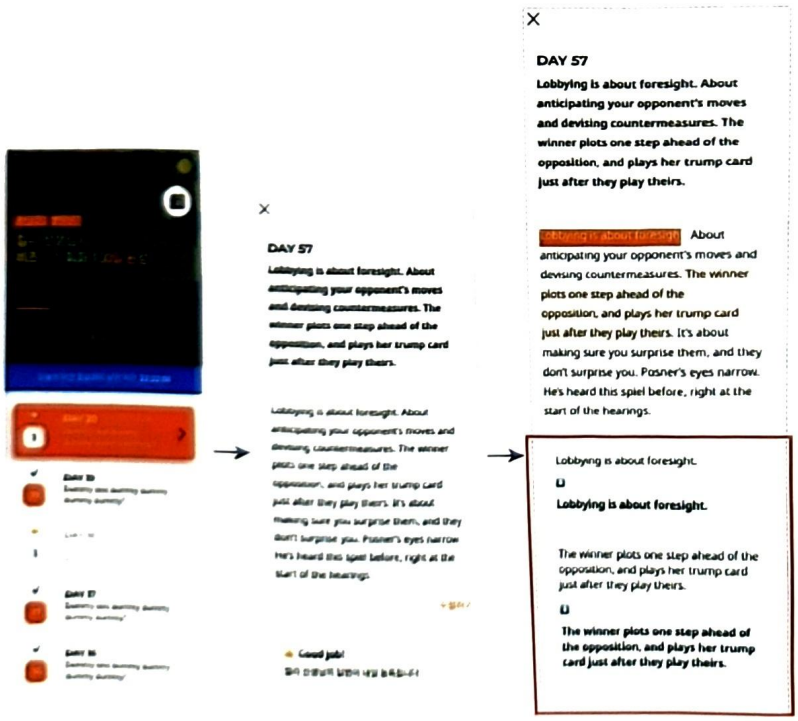
작성 툴 역시 무척 다양합니다. 파워포인트와 같은 오피스 툴부터 최근에는 피그마와 같은 디자인 툴은 물론이고 심지어 협업 툴인 노션을 활용하기도 합니다. 즉, 기획서는 말 그대로 소통을 위한 문서일 뿐 어떤 툴을 사용하느냐는 중요하지 않습니다. 기획 의도와 디자인을 개발자에게 잘 설명하는 것이 가장 큰 목표라는 점을 잊어서는 안 됩니다.

툴이나 방식은 다양할 수 있지만 기획서 역할을 하기 위해 반드시 포함해야

할 요소들은 존재합니다. 이번엔 바로 이 기획서로 개발자와 소통하기 위해선 어떤 요소들이 필요한지 예시를 통해 살펴보겠습니다.

기획서의 필수 요소들

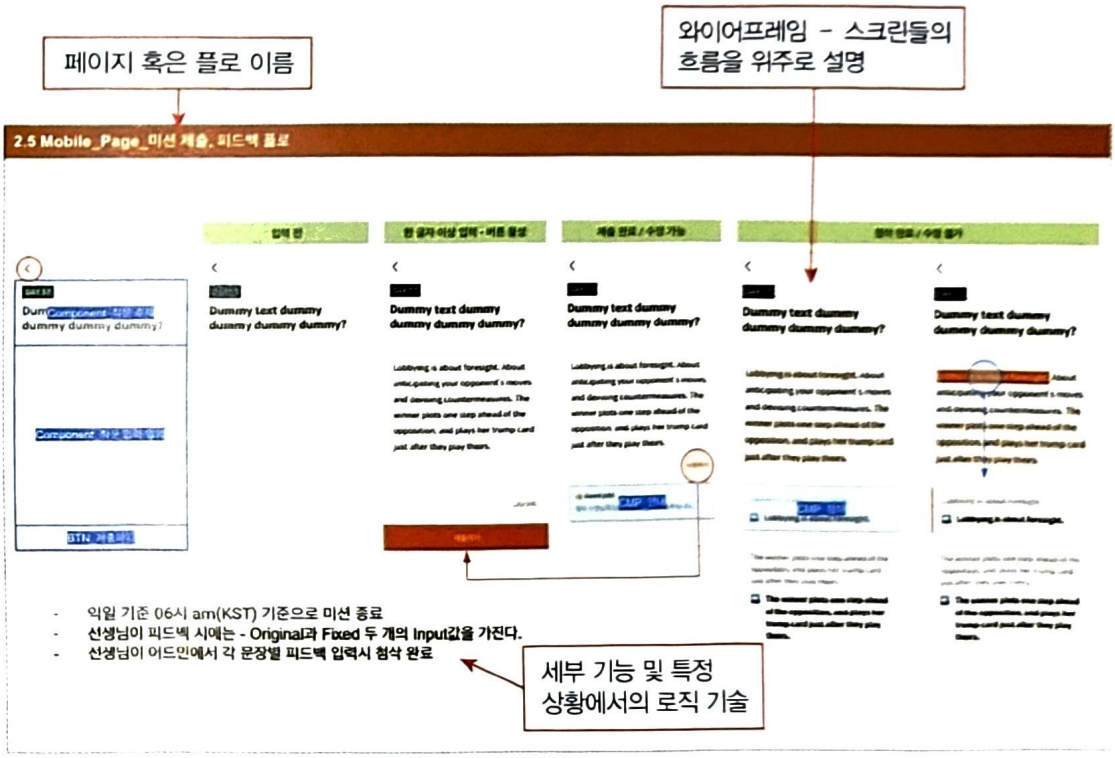
다음 예시는 영어 작문 첨삭 서비스입니다. 학생이 주어진 질문에 대한 영작을 하면 선생님이 첨삭을 제공하는 플랫폼입니다. 이런 경우 학생과 선생님 모두 중요한 사용자이므로 기획 단계에서 학생과 선생님에 대한 퍼소나를 모두 정의해야 합니다. 이처럼 퍼소나는 기획부터 개발까지 모든 단계의 기반이 되므로 반드시 리서치 단계에서 수집한 데이터를 기반해 꼼꼼하게 작성하는 것이 중요합니다. 이 서비스의 주요 화면을 살펴보면 다음과 같습니다. 학생이 매일 주어지는 미션에 대한 답변을 영작해서 제출하면 첨삭된 문장과 선생님의 피드백을 받을 수 있습니다.



내 클래스 영어 작문 제출 첨삭 완료

영어 작문 첨삭 서비스

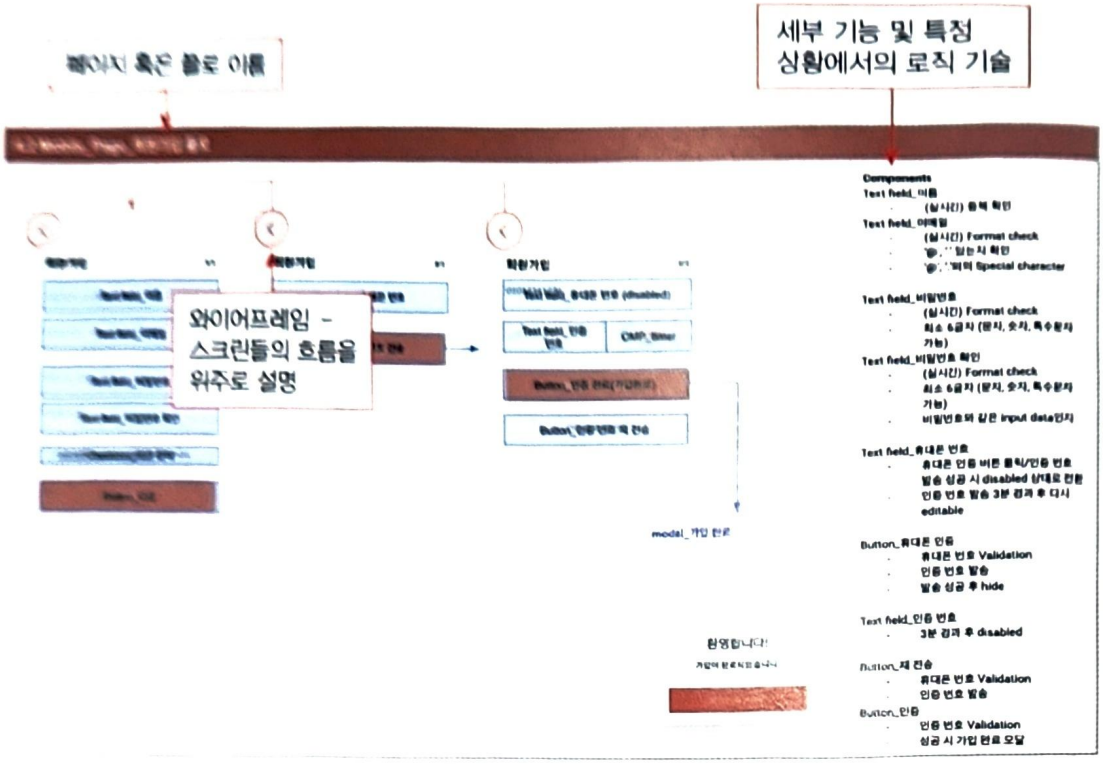
학생이 주어진 질문에 작문하고 피드백을 받는 일련의 플로를 기획서로 작성하면 다음과 같습니다. 먼저 상단에는 해당 플로나 페이지의 이름을 적고 그 아래에 와이어프레임과 기능에 대한 세세한 설명을 작성합니다. 각 요소의 흐름에 따라 화살표로 연결하기도 하고 요소별 세부 사항이 필요할 때는 하단의 여백을 활용합니다. 이때 문서에 포함되는 와이어프레임은 로우 피델리티인 경우가 많습니다.



영작 플랫폼 서비스 기획서의 일부

다음 예시는 같은 서비스에서 사용자가 회원가입 시 휴대폰 인증을 하는 플로를 나타낸 부분입니다. 와이어프레임의 오른쪽에는 각 요소 및 발생 가능한 상황에 대한 상세 설명을 작성해 개발자가 디자인 의도를 파악할 수 있도록 합니다.

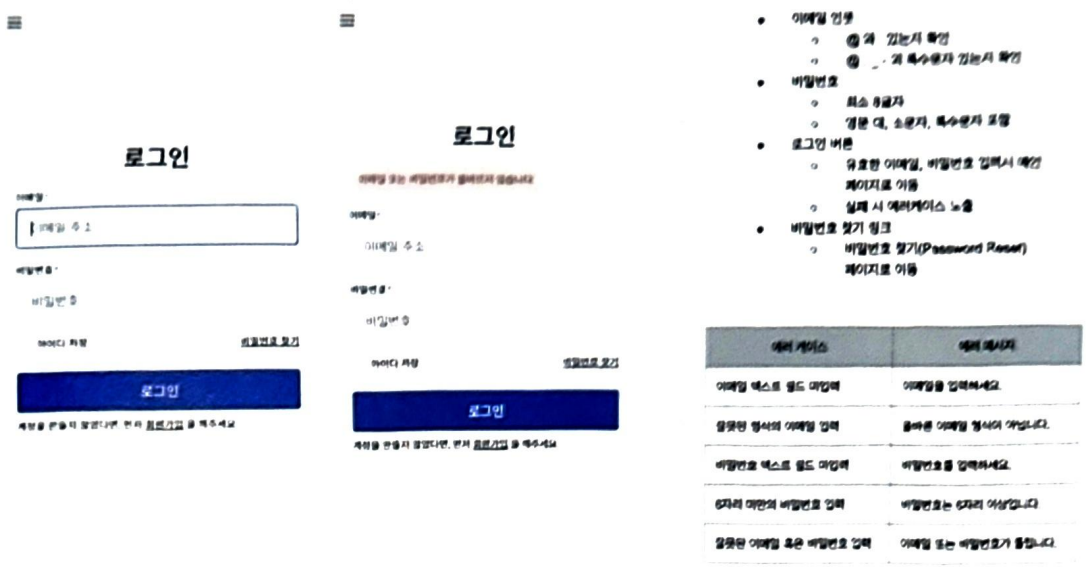
Tip. 전체적인 문서의 구조는 제공하는 노션 템플릿의 '와이어프레임과 서비스 기획서'에서 확인할 수 있습니다.



회원가입 과정에 대한 기획서

에러 케이스

기획서를 작성하는 경험이 쌓이다 보면 자연스럽게 에러 케이스에 대한 훈련이 이루어집니다. **에러 케이스**error case란 특정 플로나 피처에서 발생할 수 있는 오류와 이에 대한 대응을 정리한 문서입니다. 여기서 말하는 오류는 개발 문제로 발생한 오류 외에 사용자가 설계한 의도와 다르게 행동하는 것, 기획자가 미처 고려하지 못한 사용자의 접근 등 무척 다양하게 발생할 수 있습니다. 가령 로그인 화면에서 잘못된 이메일 주소 또는 비밀번호를 입력했을 때 "이메일 또는 비밀번호가 올바르지 않습니다."라는 에러 메시지를 보여 주는 거죠. 에러 케이스의 작성 예시는 다음과 같습니다.



에러 케이스 작성 예

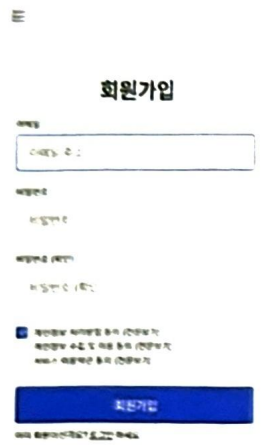
이처럼 한쪽에는 피처의 흐름을 볼 수 있는 와이어프레임 또는 화면 이미지를 두고, 다른 한쪽엔 이 흐름에서 발생할 수 있는 에러 케이스를 모두 작성해 둡니다. 이 에러 케이스는 기획서에 함께 작성해도 좋지만, 개별 문서로 목록화해 두어도 좋습니다. 단순히 에러 케이스만 기술하는 것이 아니라 에러가 발생했을 때 사용자에게 띄울 문구도 함께 정리해야 하므로 에러 케이스와 에러 문구를 함께 정리해 두는 것이 좋습니다.

에러 케이스와 기획서를 함께 작성해 두면 사용자가 오류를 일으킬 때 또는 기술적으로 오류가 날 때 사용자의 다음 행동을 제대로 된 방향으로 유도할 수 있습니다. 또, 기획자나 디자이너가 에러 케이스를 미리 목록화해 두면 상세한 내용까지 점검해 가며 개발할 수 있어 개발자들의 작업 속도도 향상됩니다. 하지만 아무리 많은 에러 케이스를 고려해도 협업 과정에서 보완점이 발견되곤 하기 때문에 개발 단계로 넘어간 이후에도 개발자와의 논의가 이어져야 합니다.

Mission 에러 케이스와 기획서 작성하기

템플릿 : 에러 케이스 & 예시 기획서

에러 케이스를 찾는 연습과 실제 기획서를 구성해 보는 연습을 함께 해보겠습니다. 다음 회원가입 페이지의 에러 케이스를 채워 보세요.



Error Cases	Wording

그리고 여러분이 작성한 와이어프레임을 기반으로 화면별 흐름과 각 요소에 대한 로직을 정리해 보세요. 템플릿 페이지에서 제공하는 '모임 플랫폼'과 '영어 독립' 기획서를 참고해 여러분만의 기획서를 작성해 보세요.

5-4 운영까지 고려한 백 오피스 기획하기

백 오피스back office는 제품 · 서비스를 운영 · 관리하는 것을 목적으로 개발하는 소프트웨어를 뜻합니다. 관리자 페이지라고도 하죠. 콘텐츠, 상품, 사용자 등 운영 · 관리와 관련된 모든 기능이 백 오피스에 포함됩니다.

백 오피스는 사용자가 직접 마주하는 화면이 아니기 때문에 초기에는 적은 자

원만 들여서 제작하기도 합니다. 그래서 디자인이 투박하거나 기능이 섬세하지 못하기도 하죠. 그러나 사용자 수가 증가하고 지표가 성장하면 백 오피스를 고려한 서비스와 그렇지 못한 서비스의 차이는 운영적인 측면에서 크게 벌어지기 시작합니다. 특히 사용자에게 발생할 수 있는 문제나 불편 사항에 적절하게 대처하기가 어려워지죠. 비록 사용자 눈엔 보이지 않지만, 운영 측면에서 백 오피스는 무척 중요한 역할을 합니다.

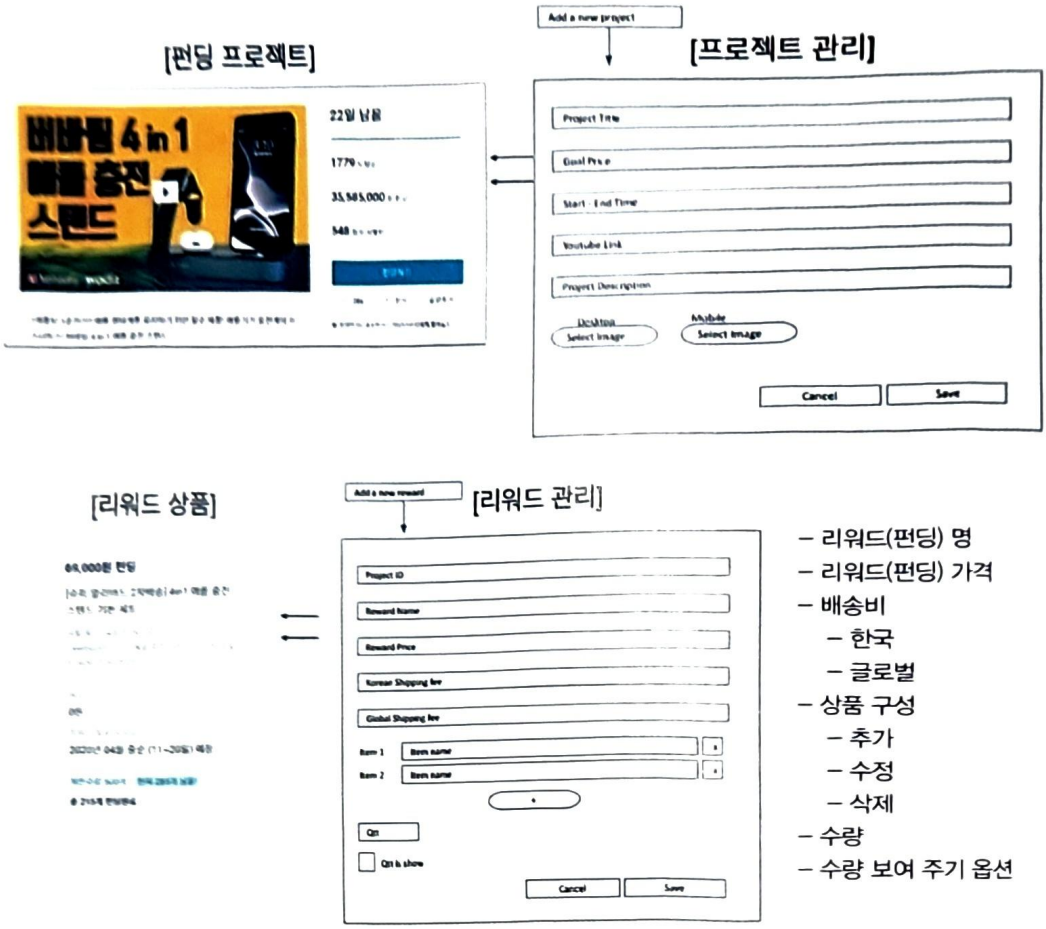
백 오피스를 기획하는 것도 PM의 몫이죠. 이번엔 백 오피스를 기획할 땐 어떤 요소를 고려해야 하는지 크라우드 펀딩 플랫폼과 영어 작문 첨삭 서비스의 백 오피스 사례를 통해 살펴보겠습니다.

크라우드 펀딩 플랫폼의 백 오피스

크라우드 펀딩crowd funding이란 생산자 · 판매자가 제품 개발 등의 목적으로 개인 구매자로부터 자금을 모으는 방식을 뜻합니다. 가령 제품 개발에 천만 원이 필요하다면 개발할 제품을 홍보해 사전 구매자를 모집합니다. 그렇게 기반 비용을 먼저 마련한 다음 제품 개발을 시작하는 방식입니다.

즉, 제품 개발에 필요한 만큼의 사전 판매가 이루어져야 펀딩에 성공하고 제품 개발을 시작할 수 있죠. 이러한 특성 덕분에 크라우드 펀딩 플랫폼에선 시장에 나오지 않은 실험적이고 재미있는 아이디어 상품을 쉽게 접할 수 있습니다. 대표적 크라우드 펀딩 플랫폼으로는 '킥스타터', '와디즈', '텀블벅' 등이 있습니다.

크라우드 펀딩은 펀딩 프로젝트에 참여할 구매자의 페이지도 중요하지만, 판매자가 프로젝트를 관리할 백 오피스도 무척 중요합니다.



구매자가 보는 화면(왼쪽)과 판매자가 보는 백 오피스 화면(오른쪽)

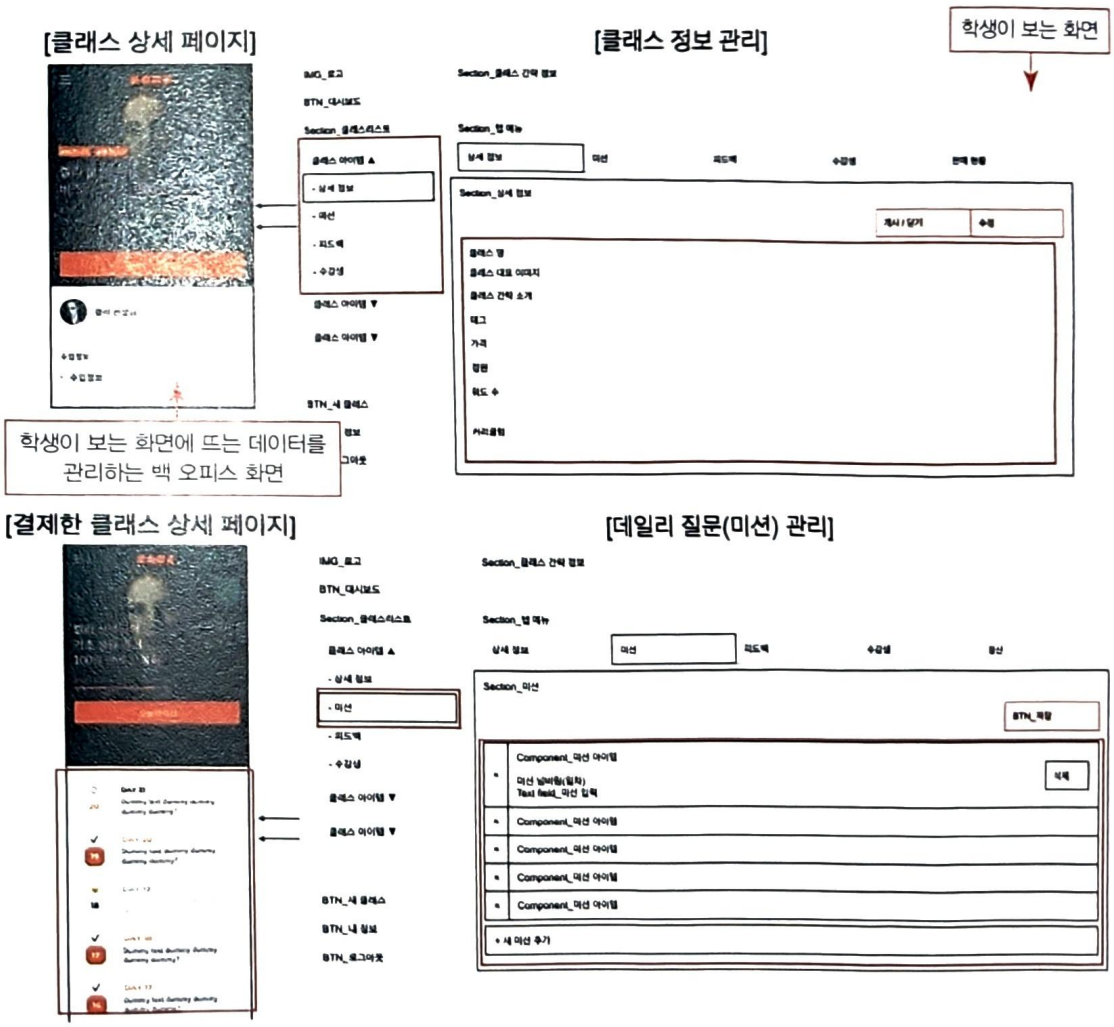
이처럼 구매자 화면은 판매자나 내부 운영자가 백 오피스에서 프로젝트를 생성하고 관리한 덕분에 구성됩니다. 구성 요소와 피처는 플랫폼마다 조금씩 다르지만 대체로 판매자는 프로젝트 제목, 가격, 홍보 영상, 상세 페이지, 잔여 수량 등을 관리합니다. 판매 현황을 볼 수 있는 대시보드도 필요하죠. 즉, 백 오피스는 운영을 위해 설계, 개발하는 것이므로 화려한 디자인보다는 정보 구조나 가독성을 우선으로 고려하고 개발해야 합니다.

백 오피스 설계에 익숙하지 않다면 대부분 데이터가 준비되어 있다는 가정하에 사용자 화면을 먼저 디자인하는데요. 이 경우 다양한 상황에 대응하기가 어렵습니다. 따라서 초기엔 항상 데이터가 없는 시작점부터 설계를 고려하는 것이 좋습니다. 실제 제품 · 서비스가 운영되려면 어떻게 데이터가 생성되고

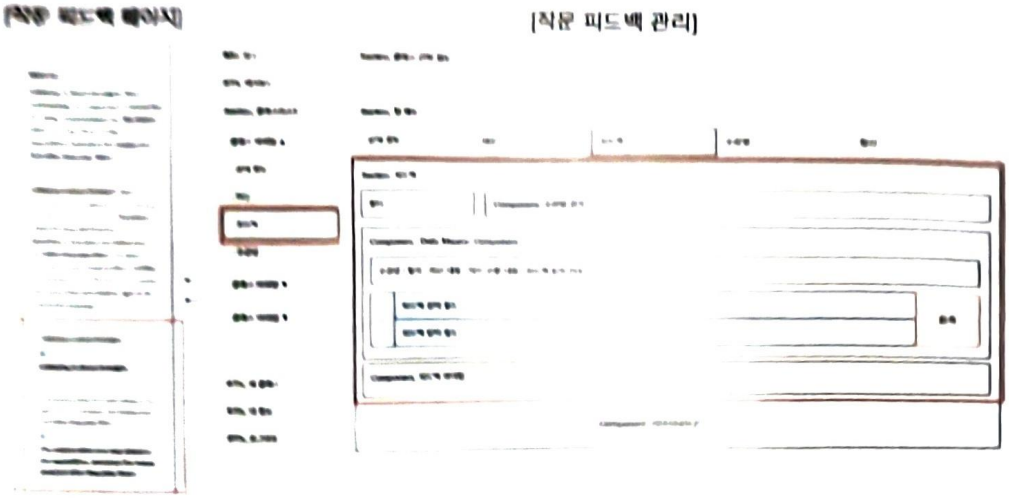
이 데이터가 사용자에게 어떻게 보이는지 그리고 인터랙션하는지 전체를 고려할 수 있어야 합니다.

영어 작문 첨삭 서비스의 백 오피스

다음은 앞서 살펴본 영어 작문 첨삭 플랫폼의 예시입니다. 영어 작문 첨삭 플랫폼은 학생이 클래스를 듣고 과제로 영작문을 제출하면 선생님이 피드백을 주는 방식입니다. 즉, 백 오피스를 이용할 사용자는 선생님이 됩니다. 이때 선생님은 학생들이 제출한 영작문에 피드백을 주는 것뿐만 아니라 클래스를 생성하고 관리하고 참여 학생들에 대한 데이터도 볼 수 있어야 합니다.



학생이 보는 화면(왼쪽)과 선생님이 보는 백 오피스 화면(오른쪽)



작문 피드백 페이지

이처럼 학생들이 보는 클래스, 영작문 과제들은 모두 선생님이 운영 · 관리하는 백 오피스에 입력한 데이터입니다.

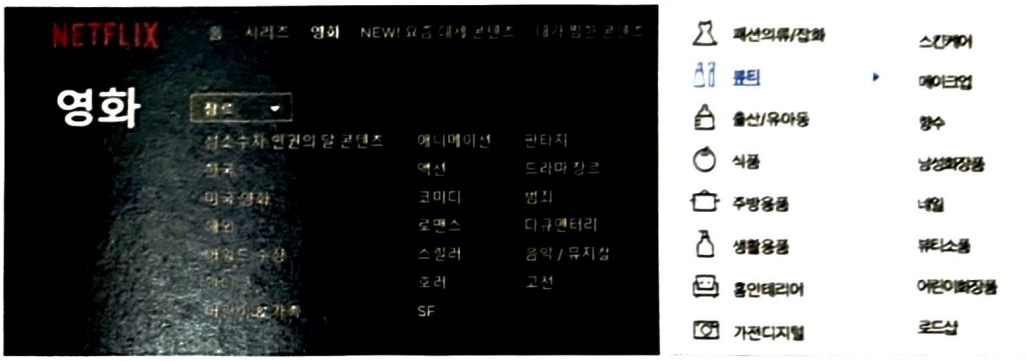
백 오피스는 제품 개발과 운영을 하는 스타트업에서는 특히 필수입니다. 실무에서 백 오피스를 설계할 때는 내부 관리자나 플랫폼 내 생산자를 기준으로 데이터가 전무한 시점부터 데이터가 생성되고 사용자와 인터랙션하는 전체 고객 여정 지도를 작성해 보면 어떤 피처들이 필요한지 꼼꼼히 챙길 수 있습니다. 또, 제품 · 서비스 특성에 따라 백 오피스 사용자의 편의를 위해 검색, 필터 등 추가 기능이 필요한 경우, 개발자와 사전에 협의를 해야 할 수 있으니 초기엔 지나치게 복잡하게 설계하는 것보다 대략적 구조를 잡고 개발자와 피드백을 주고받으면서 작업하는 것이 효율적입니다.

Tip. 영어 첨삭 서비스의 더 자세한 백 오피스와 기획이 실제 UI로 어떻게 바뀌었는지 궁금하다면 노션 템플릿에서 제공하는 '백 오피스 기획' 페이지를 참고하세요.

5-5 사용자가 길을 잃지 않는 '정보 구조' 구성하기

정보 구조information architecture, IA는 정보의 성격이나 순서, 우선순위 등에 따라 구조화하는 것을 의미합니다. 잘 설계한 정보 구조는 사용자의 접근성을 높이고 사용자 스스로 길을 찾을 수 있도록 돕습니다. 많은 사용자가 이용하는 서비스를 관찰하면 빠른 검색을 돕는 카테고리, 직관적인 콘텐츠 배치 등 정보 구조가 간결하고 명확한 것을 볼 수 있습니다.

Tip. 앞서 학습했던 유저 플로, 퍼소나, 와이어프레임 또한 일종의 정보 구조에 해당합니다.



넷플릭스의 콘텐츠(왼쪽) 쿠팡의 제품 카테고리(오른쪽) 정보 구조

반면 잘못 설계된 정보 구조는 사용자가 오류를 경험하게 하고 작업 수행에 드는 시간이 늘어나 이탈률을 높이기도 합니다. 예를 들어 주문한 상품을 환불하기 위해 보통 직관적으로 [마이페이지] 또는 [나의 정보]에서 주문한 기록을 찾기 마련입니다. 이때 이곳에서 주문한 상품 목록을 찾지 못하면 사용자는 곧장 혼란에 빠지게 되죠.

사용자가 유입되고 전환되는 과정은 결코 쉽지 않은 반면 이탈은 아주 쉽고 빠르게 이루어집니다. 이를 방지하기 위해서는 사용자가 진입하는 순간부터 명확한 정보 구조를 제공해 원하는 것을 빠르게 찾을 수 있도록 도와 주어야 합니다. 물론 생각처럼 쉽지 않은 일입니다. 심지어 정보 구조를 개선하는 작업이 하나의 UX 프로젝트가 되는 경우도 있습니다.

웹/앱의 정보 구조

간결하고 명확한 정보 구조는 사용자가 생각하는 우선순위를 그대로 구현하는 것입니다. 즉, 사용자의 **멘탈 모델**mental model을 고려해야 하죠. 가령 '비밀번호 변경은 [내 정보]에 있을 것이다.', '제품 정보는 [제품 상세페이지]에 있을 것이다.'라는 직관적 흐름을 그대로 따라갈 수 있도록 구성된 것이 잘된 정보 구조입니다.

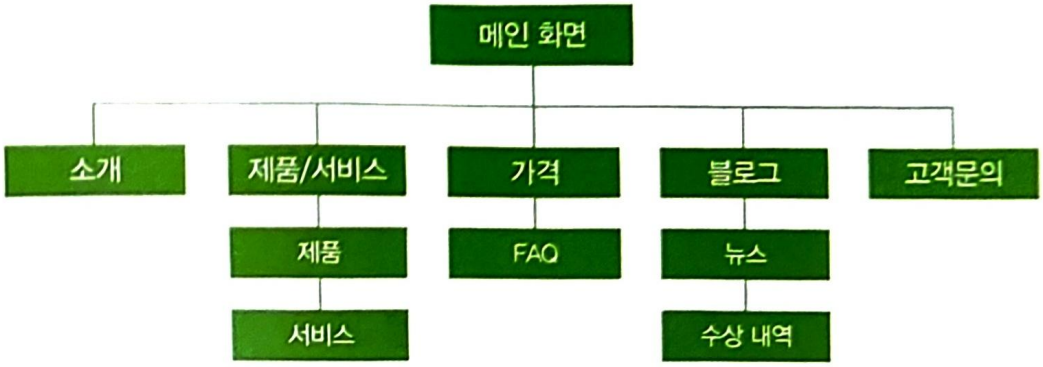
용어 사전 멘탈 모델

멘탈 모델mental model이란, 심성 모형이라고도 하며, 도널드 노먼은 '사람들이 자기 자신, 다른 사람, 환경, 자신이 상호 작용하는 사물들에 대해 갖는 인지적 모형'이라고 정의합니다.

정보 구조를 구성하는 것은 간단해 보이지만 제품 · 서비스의 전체 흐름과 구조, 각 유저 플로를 머리에 그릴 수 있어야 합니다. 또, 사용자 멘탈 모델에서 크게 벗어나지 않도록 흔히 볼 수 있는 정보 구조도 파악하고 있어야 하죠. 따라서 자주 방문하는 웹/앱은 어떻게 정보 구조를 구성했는지 평소에 유심히 보고 분석해 보는 것도 좋은 훈련이 됩니다. 또 웹과 앱의 정보 구조는 어떻게 다른지 봐두는 것도 좋은데요. 어떤 점이 다른지 간단하게 살펴보겠습니다.

웹의 정보 구조

웹의 정보 구조는 사용자 특성과 서비스 목적에 따라 무척 다양하지만 가장 흔하게 접할 수 있는 웹사이트의 구조는 홈 화면에서 출발해 하위 계층에 메뉴, 메뉴 안에 콘텐츠 등이 있습니다. 가령 홈 화면이 있고 회사 소개, 연락처, 블로그 등의 메뉴를 배치한 다음 그 하위 콘텐츠 페이지나 상세 페이지 등이 있는 식이죠. 이러한 정보 구조를 시각화한 것이 사이트 맵입니다.



웹사이트의 정보 구조(사이트 맵)

앱의 정보 구조

앱의 정보 구조 예시로 모바일 앱 '인스타그램'을 살펴보겠습니다. 앱의 경우 하단 탭을 활용하는 내비게이션 구조로 되어 있는 경우가 많습니다. 그래서 일반적인 웹사이트와는 달리 정보 구조의 최상단에 하단 탭의 메뉴들이 위치하고 각 탭의 하위 계층에는 해당 탭 내부의 메뉴나 피처들이 배치됩니다.

앱 서비스의 정보 구조 (출처: 인스타그램)

앱 화면은 정보의 접근성을 높이기 위해 하단 탭을 활용하는 것이 일반적입니다. 하지만 최근에는 웹과 앱의 경계가 점점 사라지고 있습니다. 예를 들어 펫 스타트업인 '펫프렌즈'의 경우 데스크톱 버전의 웹사이트에 들어가도 앱과 동일한 형태의 UI를 확인할 수 있습니다. 이를 통해서 회사의 비즈니스 모델과 자원에 따라 선택하는 기술과 정보 구조도 다양해짐을 알 수 있습니다.

정보 구조는 정보를 그룹화하고 시각화하는 작업으로, 정해진 방법도 툴도 따로 없습니다. 따라서 작업자의 성향 또는 제품 · 서비스의 특성에 따라 가장 효율적인 방법을 찾는 것이 좋습니다. 예를 들어 정보 구조의 단계를 한눈에 파악하기 쉬운 구글 스프레드시트를 활용하는 것도 좋은 방법입니다.

				Type			비고
[illegible]				Landing, Main, Search	비회원, 수강생, 강사	비회원, 수강생, 강사	강사 모드에서 레이아웃 변경
[illegible]	[illegible]	[illegible]		View, Link	비회원, 수강생, 강사	비회원, 수강생, 강사	
[illegible]		[illegible]		View	비회원, 수강생, 강사	비회원, 수강생, 강사	
[illegible]		[illegible]		View	비회원, 수강생, 강사	비회원, 수강생, 강사	
[illegible]	[illegible]	[illegible]		List	비회원, 수강생	비회원, 수강생	카드 뷰 형태 리스트 수업 예약 및 결제
[illegible]	[illegible]	[illegible]		List	비회원, 수강생, 강사	비회원, 수강생, 강사	
[illegible]	[illegible]			View, File Upload	비회원, 수강생	수강생	
[illegible]	[illegible]			Write, Link	비회원, 수강생, 강사	비회원, 수강생, 강사	
[illegible]	[illegible]			Write, Session	비회원	비회원	KCP 본인인증 모듈
[illegible]	[illegible]			Session	비회원	비회원	
[illegible]	[illegible]	[illegible]		Footer, View	비회원, 수강생, 강사	비회원, 수강생, 강사	
[illegible]	[illegible]	[illegible]		Footer, View	비회원, 수강생, 강사	비회원, 수강생, 강사	
[illegible]		[illegible]		View	비회원, 수강생, 강사	비회원, 수강생, 강사	
[illegible]		[illegible]		View	비회원, 수강생, 강사	비회원, 수강생, 강사	
[illegible]		[illegible]		View	비회원, 수강생, 강사	비회원, 수강생, 강사	
[illegible]	[illegible]	[illegible]		Footer, View	비회원, 수강생, 강사	비회원, 수강생, 강사	
[illegible]	[illegible]			Footer, Link	비회원, 수강생, 강사	비회원, 수강생, 강사	네이버 블로그, 인스타그램, 페이스북 링크 제공
[illegible]	[illegible]	[illegible]		List	수강생	수강생	
[illegible]	[illegible]			List	수강생, 강사	수강생, 강사	
[illegible]	[illegible]			View, Write	수강생, 강사	수강생, 강사	
[illegible]	[illegible]	[illegible]		View, Link, Authority	수강생, 강사	수강생, 강사	
[illegible]		[illegible]		Update	수강생, 강사	수강생, 강사	
[illegible]		[illegible]		List	수강생	수강생	
[illegible]		[illegible]		Session	수강생, 강사	수강생, 강사	
[illegible]	[illegible]			Write, Update, File Upload	강사	강사	

스프레드시트를 사용한 필라테스 예약 플랫폼 '필러스'의 정보 구조

스프레드시트로 정보 구조를 표현할 때는 시트의 열을 이용해 간단하게 계층을 나타낼 수 있어 실무에서 자주 사용하는 도구이기도 합니다. 이처럼 정보 구조가 익숙하지 않은 초기에는 복잡하지 않은 웹사이트나 모바일 앱의 정보 구조를 참고하며 연습한 다음 손에 익기 시작하면 점점 복잡한 정보 구조를 그려 보기를 추천합니다.

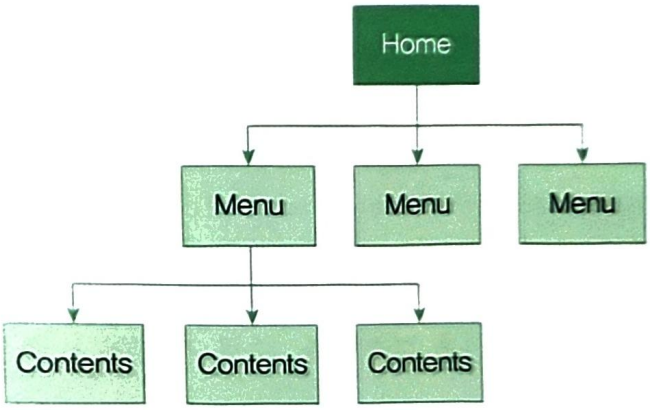
Tip. 노션 템플릿에서 제공하는 '스프레드시트형 정보 구조' 템플릿을 참고하세요.

대표적인 3가지 내비게이션 유형

정보 구조 중 웹/앱에서 사용하는 대표적인 3가지 내비게이션 유형을 함께 살펴보겠습니다.

계층적 내비게이션

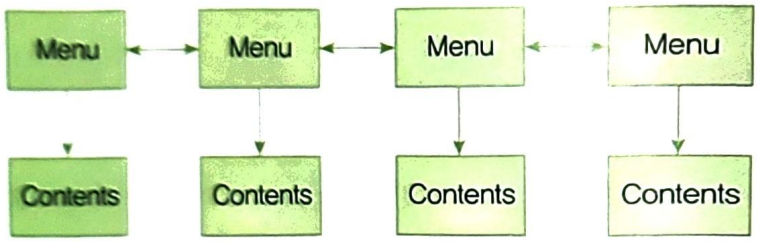
계층적 내비게이션hierarchical navigation은 홈 메뉴에서 시작해 한 계층씩 아래로 내려가면서 메뉴(정보)를 표현합니다. 따라서 서로 연결되어 있지 않은 페이지로 이동하려면 처음 시작한 화면으로 돌아가야 합니다. 이런 형태의 내비게이션은 주로 이메일 서비스, 기업 페이지 그리고 폴더 구조에서도 볼 수 있습니다.



계층적 내비게이션

플랫 내비게이션

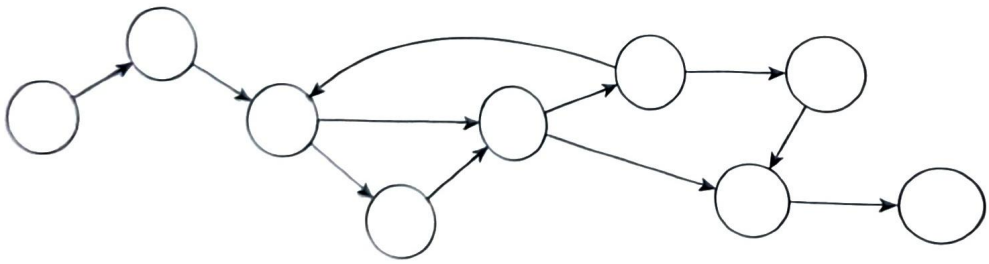
플랫 내비게이션flat navigation은 최상위 메뉴가 2개 이상이고 서로 이동할 수 있는 구조입니다. 모바일 앱에서 흔히 볼 수 있는 구조로, 안드로이드에서는 **보텀 내비게이션**bottom navigation, iOS에서는 **탭 바**tab bar라고 합니다. 앱에서는 보통 하단 탭이 정보 구조의 최상위 레벨로, 하나의 최상위 메뉴에서 아무리 깊게 들어가도 다른 최상위 메뉴로 옮겨 가면 완전히 새로운 페이지로 이동할 수 있죠. 인스타그램과 같이 보텀 내비게이션이나 탭 바가 존재하는 대부분 앱 서비스에서 플랫 내비게이션을 볼 수 있습니다.



플랫 내비게이션의 구조

콘텐츠 드리븐 내비게이션

콘텐츠 드리븐 내비게이션content-driven navigation은 콘텐츠에서 콘텐츠로 이동할 수 있어 다른 내비게이션보다 메뉴와 메뉴 사이 이동이 유연합니다. 이 구조에서는 콘텐츠 자체가 내비게이션을 결정하죠. 주로 게임이나 전자책에서 자주 볼 수 있습니다.



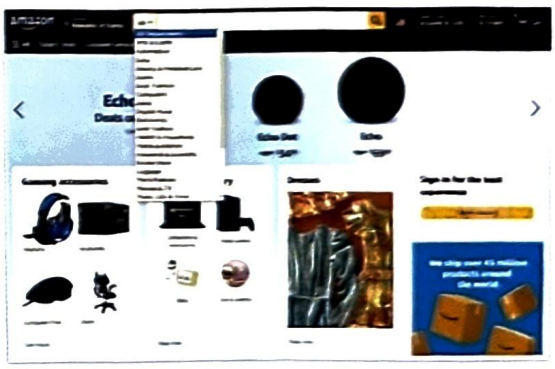
콘텐츠 드리븐 내비게이션 구조 도식

궁금해요 메뉴를 어떻게 묶고 배치해야 하나요?

정보 구조의 메뉴를 구성하는 데도 사용자 리서치는 무척 유용합니다. 이때 가장 널리 사용하는 방법이 카드 소팅card sorting이라는 정성적 사용자 리서치입니다. 카드 소팅은 앞서 어피니티 다이어그램처럼 상위 정보와 하위 정보가 적힌 카드를 나열한 다음 우선순위에 맞게 그룹화하는 방식입니다. 어피니티 다이어그램과 차이가 있다면 피처가 아니라 사용자가 직접 우선순위를 정리한다는 것입니다. 메뉴가 많은 커머스 서비스나 복잡한 금융 서비스 등에서 우선순위를 정할 때 무척 유용합니다.

카드 소팅 방법은 간단합니다. 상위 정보와 하위 정보가 적힌 카드들을 나열하고 리서치 참가자는 자신이 생각하는 기준으로 카드들의 우선순위를 세우고 이에 맞게 그룹화합니다. 여러 참가자와 카드 소팅을 진행한 다음 공통적으로 나타나는 패턴을 기준으로 우선순위를 재조정합니다.

카드 소팅은 콘텐츠의 우선순위를 정할 때도 사용하지만 홈 화면의 피드 구성이나 상세 페이지의 정보 우선순위를 정할 때도 유용합니다.



카드 소팅 리서치(왼쪽) 우선순위에 따라 나열된 메뉴(오른쪽)

5-6 피처의 완성도를 높이는 QA와 테스트 케이스

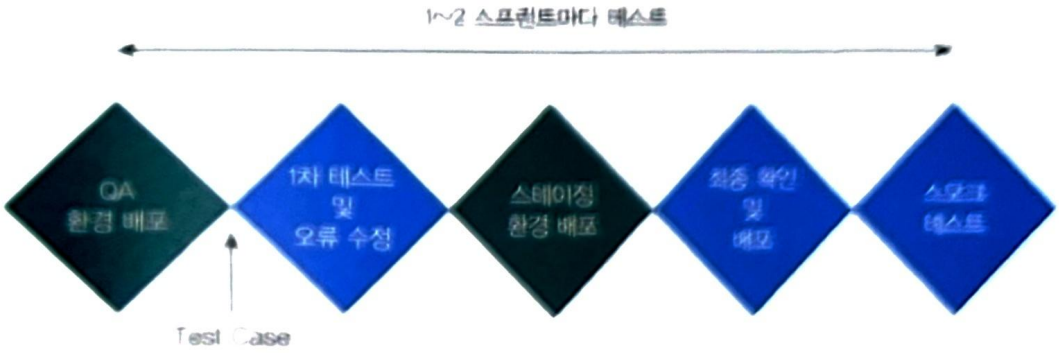
품질을 높이는 QA

[illegible] 서비스에서 피처를 구현 [illegible] 개발 작업을 모두 [illegible] 피처가 잘 동작하는지, [illegible] 지 등을 확인하는 [illegible] QA [illegible] 가 필요합니다.

QA [illegible] 위해 테스트 및 관리의 단계로, 프로젝트 [illegible] 전은 물론 사용자에게 [illegible]의 품질과 가치 [illegible] QA에는 품질 향상, 유지, 개선 등 여러 업무가 [illegible] 품질 관리로 정리할 수 있습니다(소규모 기업 [illegible] 경우도 있습니다).

[illegible] 프로세스 과정들

QA 담당자는 기획부터 개발까지 모든 과정에 참여하면서 QA 과정을 최적화합니다. 또, 회원가입이나 결제와 같은 중요한 단계에서 오류가 발생하지 않도록 개발자가 프로그래밍한 코드를 시뮬레이션하는 테크니컬 QA(technical quality assurance)를 진행하기도 합니다. 애자일 환경에서 스프린트를 거쳐 제품을 개발하는 조직의 QA는 대체로 다음과 같이 진행됩니다.



QA 프로세스

조직에 따라 다를 수 있지만, 보통 한 스프린트는 2주에서 4주 정도로 소요되며 스프린트마다 QA와 배포를 하는 곳도 있고 2~3회 정도 스프린트를 진행하고 한번에 모아 QA와 배포를 하기도 합니다.

QA가 자리 잡힌 개발 조직에서는 개발 단계부터 테스트를 염두에 둡니다. 테스트할 수 있는 수준으로 개발이 완료되면 QA 담당자는 테스트 케이스(test case)라는 문서를 참고해 피처가 작동하는지 테스트합니다. 테스트 과정에서 버그가 발생했을 때, 일반적으로 사용자가 상호 작용하는 화면에 문제가 있으면 프런트엔드 개발자에게, 데이터를 주고받는 과정에 문제가 있으면 백엔드 개발자에게 전달해 이슈를 확인하고 해결할 수 있도록 합니다.

개발 서버에서 1차 테스트가 완료되면 사용자가 실제로 사용하는 것과 동일한 데이터를 가진 환경인 **스테이징 버전**(staging version)으로 제품 · 서비스를 배포합니다. 스테이징 버전에서도 2차 테스트를 거치고 문제가 없다면 실제 사용자가

[illegible] 프로덕션 배포 [illegible]

[illegible] 중요한 비즈

[illegible] 스모크 테스트 [illegible]

Point 스테이징 서버 & 스모크 테스트

스테이징 서버 [illegible] 실제와 유사한 환경 [illegible]

스모크 테스트 [illegible] 테스트하는 작업 [illegible]

[illegible]

테스트 케이스 [illegible] 및 수행 절차를 [illegible] 테스트 케이스를 작성할 때는 대분류에서 중분류, [illegible] 가령 웹사이 [illegible] 회원가입, 로그인 화면으로 좁히는 [illegible] 사이트와 같은 툴을 이용해 작성할 수 [illegible] 다음과 같습니다.

Tip [illegible] 이를 상세히 적어 두었다면 테스트 케이스를 작성하는 데 많은 시간을 들이지 않아도 됩니다.

No	테스트 시나리오	테스트 조건	실행 순서	Pass Result	테스트 유형	P/F	비고
	이메일로 회원가입 이메일/비밀번호 입력 오류 실패	이메일 계정 있음 회원가입 미진행 상태	1. 앱 진입 2. 메일 계정으로 계속하기 탭 3. 회원가입 탭 4. 이메일에 잘못 입력 하여 에러 메시지 발생하도록 입력 5. 다른 입력 창 탭	에러 메시지 발생 이메일이 올바르지 않음을 알려 주는 팝업 창이 나타나도록 구성해 주세요	CS	Pass	
	이메일로 회원가입 비밀번호 입력 실패	이메일 계정 있음 회원가입 미진행 상태	1. 앱 진입 2. 메일 계정으로 계속하기 탭 3. 회원가입 탭 4. 비밀번호에 대문자 소문자 특수문자 중 한 가지만 작성 5. 비밀번호를 5자 이만으로 작성 6. 다른 입력 창 탭	에러 메시지 발생 비밀번호는 영문 8문자 소문자 특수문자 중 2가지 이상을 사용해 5자리 이상으로 설정해 주세요	CS	N/A	
3	이메일로 회원가입 비밀번호 재입력 불일치	이메일 계정 있음 회원가입 미진행 상태	1. 앱 진입 2. 메일 계정으로 계속하기 탭 3. 회원가입 탭 4. 비밀번호 바르게 입력 5. 새 비밀번호와 다른 정보 입력 6. 다른 입력 창 탭	에러 메시지 발생 비밀번호는 영문 8문자 소문자 특수문자 중 2가지 이상을 사용해 5자리 이상으로 설정해 주세요	CS	N/A	
4	이메일로 회원가입 이메일/비밀번호 설정 성공	이메일 계정 있음 회원가입 미진행 상태	1. 앱 진입 2. 메일 계정으로 계속하기 탭 3. 회원가입 탭 4. 이메일 바르게 입력 5. 비밀번호 바르게 입력 6. 비밀번호와 같은 정보 입력 7. 다음 탭	1. 다른 정보 입력 창으로 이동	CS	N/A	
5	비밀번호 재설정				CS	N/T	

테스트 케이스를 정리한 스프레드시트 예시

먼저 '테스트 시나리오'에는 테스트할 피처 또는 작업을 적습니다. 유저 스토리에서 '~을 할 수 있다.'라고 적었던 부분이 여기에 해당합니다. 유저 스토리의 제목을 그대로 써도 무방합니다. 이어서 '테스트 조건'에는 테스트하는 계정의 상태를 작성합니다. 그리고 '실행 순서', '실행 결과'에는 유저 스토리에서 작성한 '구현에 대한 기준'을 반영합니다. 예를 들어 '이메일의 형식을 잘못 입력했을 때는 유효하지 않은 이메일이라는 에러 메시지가 뜬다.'라는 식으로 조건과 순서, 통과 기준을 작성합니다.

'테스트 환경'은 말 그대로 제품 · 서비스를 테스트한 환경을 기재합니다. 웹이라면 모바일/데스크톱 환경 또는 실행한 브라우저, 앱이라면 운영체제나 사용 기기, 해상도가 테스트 환경이 될 수 있습니다. 'P/F'는 통과 또는 실패 여부를 나타내는 항목으로, P는 Pass(통과), F는 Fail(비정상 작동, 이슈)를 뜻합니다. 이외에도 Block(보류), N/A(개발 지연 또는 취소), N/T(테스트 진행 안 함)가 있으며 이를 표로 정리하면 다음과 같습니다.

Pass	정상 작동
Fail	비정상 작동으로 이슈 발생
Block	여러 원인으로 테스트 진행이 보류
N/A(Not Available)	개발이 지연되거나 개발이 취소되는 기능 사업자 및 개발 담당자와 협의해서 테스트를 진행하지 않는 경우
N/T(Not Tested)	아직 테스트를 진행하지 않은 경우

테스트 케이스는 100% 통과를 목표로 하지 않습니다. 부족한 부분이 있을 가능성을 염두에 두고 실무자들의 협업으로 보완해 나가는 것이 바람직합니다. 웹 브라우저나 모바일 기기를 테스트할 때도 내부 자원에 따라 사용자 비중이 큰 주요 환경을 중심으로 테스트하는 것이 효율적입니다.

소규모 개발 조직이나 애자일 조직에서는 테스트 케이스 작성에 드는 시간을

줄이기 위해 테스트 케이스를 미리 작성하지 않고 바로 실행해 보면서 테스트하는 **탐색적 테스팅**exploratory testing으로 대체하기도 합니다. 탐색적 테스팅은 보통 1~2시간 정도 제한된 시간time-boxing 안에 스프린트 목표에 따라 테스트를 수행하고 요약 보고debriefing하는 방식입니다.

이외에도 이전에 발생한 이슈를 다시 테스트하는 **리그레션 테스트**regression test 등이 있으니 조직의 자원을 고려하되 피처의 완성도를 최대한 높일 수 있도록 다양한 테스트를 진행해 보세요.

Tip. 노션 템플릿에 테스트 케이스 템플릿과 실무에서 타 부서와 함께 발생한 버그를 보고받고 관리하는 버그 보드 템플릿도 있으니 참고하세요.

궁금해요 **피처 검증 테스트에는 어떤 것이 있나요?**

피처 검증 테스트, 탐색적 테스팅 외에도 앞서 잠시 언급했던 스모크 테스트를 비롯해 해피 패스 테스트, 리그레션 테스트 등 다양한 종류의 테스트가 있습니다. 이 테스트들은 각기 목표와 방식이 다르기 때문에 조직의 자원이나 상황을 고려해 종류나 순서를 달리 할 수도 있고 여러 테스트를 한 번에 할 수도 있습니다. 그렇다면 각 테스트의 특성을 알아보겠습니다.

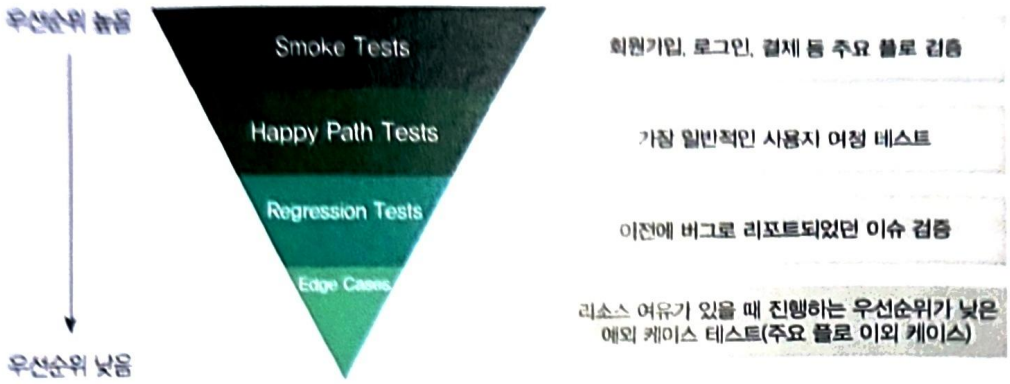
먼저 수시로 실행할 수 있는 테스트로는 스모크 테스트가 있습니다. 회원가입, 로그인, 결제 등 주요 플로가 정상 작동하는지 확인할 수 있죠. PM은 혹시 모를 오류가 없을지 수시로 주요 플로에 스모크 테스트를 진행하는 것이 좋습니다.

해피 패스 테스트는 사용자가 가장 많이 경험하는 진행 경로를 테스트하는 것입니다. 고객 여정 지도를 테스트하는 유형이라고 볼 수 있습니다.

리그레션 테스트는 이전에 보고했던 이슈들을 다시 테스트하는 것입니다. 배포 과정에서 과거 이슈들이 다시 발생하는 경우가 종종 있어 QA 환경에서 발생한 이슈를 스테이징에서 다시 확인하는 것도 리그레션 테스트의 일종입니다.

엣지 케이스는 주요한 유저 플로 외에 창의성을 발휘하여 일어날 수 있는 다양한 케이스를 테스트하는 것입니다. 다른 테스트에 비해 상대적으로 우선순위가 낮아 자원이 부족한 조직에서는 생략하는 경우가 많습니다.

이 테스트들의 중요도를 그림으로 나타내면 다음과 같습니다.

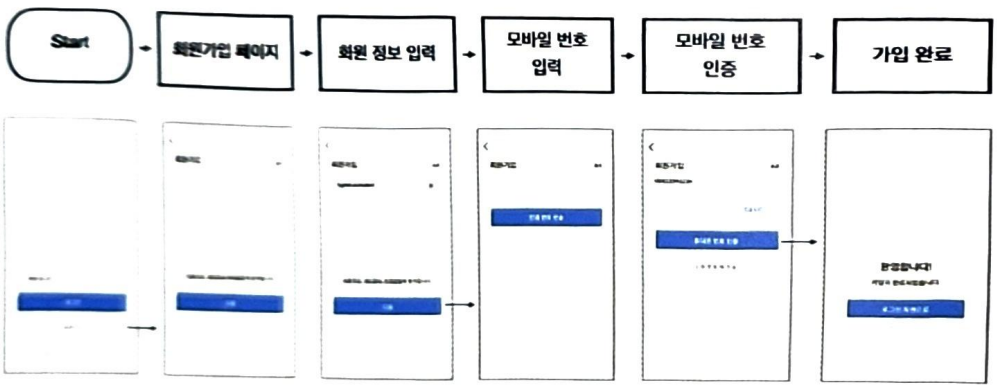


중요도로 보는 테스트 유형

Mission **테스트 케이스 작성하기**

템플릿 : 테스트 케이스

다음 회원가입 화면의 유저 플로를 확인하고 테스트 케이스를 작성해 보세요. 전체를 한번에 작성하는 것보다 큰 기능부터 대분류에서 하분류로 범위를 좁히면서 작성하는 것이 좋습니다.



No	테스트 시나리오	테스트 조건	실행 순서	Pass Result	테스트 환경	P/F	실제 결과	비고
0								
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								