
OWNING THE WORK

AI - NATIVE DESIGN

OLIVE YOUNG

Taking ownership of the problems I find
using AI to spend more time
on what matters — what to build, and why.

AI와 함께 일합니다.

AI를 활용한 Skill, Plugin을 만들어 반복업무를 줄이고 효율적으로 작업합니다.
단순 작업은 AI에게 맡기고 무엇을, 왜, 어떻게 만들지에 집중합니다.

효율적으로 일할 수 있는 시스템을 만듭니다.

Toss Place 포스 / 가비아 하이웍스 / 신한은행 AI Agent 플랫폼의 디자인 시스템을
구축·운영하며, 팀원들이 효율적으로 일할 수 있는 환경을 만들어 왔습니다.

주도적으로 일합니다.

2번의 서비스 릴리즈를 경험하며, 문제를 주도적으로 찾아 해결하는 태도로 일해왔습니다.

5 PROJECTS

AI • UX/UI • Design

Kang seohyun
010 7669 6903
rkdtjs1103@naver.com

OLIVE YOUNG

CAREER.

EDUCATION

2021.03 ~ 2025.08

숙명여자대학교 전체평점: 4.23/4.5 전공평점: 4.33/4.5

WORK

2026.06 ~

Toss Place, Platform Design Assistant

- Toss Place 디자인 시스템 마이그레이션
- Codemod 마이그레이션을 위한 정책 작성 및 자동화
- Toss POS Color Strong(고 대비 모드)리서치 · 도입

2026.01 ~ 2026.04

Gabia, UX UI 디자이너 인턴

- 하이웍스 앱 디자인 가이드 · 시스템 마이그레이션
- 하이웍스 홈페이지 개선 제안 작업 및 AI 바이브 코딩

2025.04 ~ 2025.12

R-JAEOB(UX UI 디자인 에이전시)

- 신한은행 AI Agent 플랫폼 UX/UI 디자인
- 신한은행 AI Agent 플랫폼 가이드 · 시스템 구축

ENGLISH

2025 English OPIC: IH (Intermediate High)

AWARDS

- 2025 대한민국 어린이 제품 디자인 공모전 입선
- 2025 숙명여자대학교 졸업성적우등상
- 2024 숙명여자대학교 캡스톤 경진대회 최우수상
- 2024 대한민국 패키징 대전 한국포장기술사회장상
- 2024 UX Design Award New Talent 2024
- 2024 ASIA DESIGN PRIZE 2024, Finalist
- 2023 KIDP 대한민국 디자인 전람회 특선
- 2023 KIDP 한글 원천 아이디어 콘텐츠 공모전 금상
- 2023 LG U+ 고객아이디어 공모전 대상
- 2022 숙명여자대학교 성적 학업우수상
- 2022 한국 장애인 인권포럼 유니버설 공모전 입선

ACTIVITIES

- 2024 34기 SOPT IT 동아리 활동
- 2023 fwd 숙명여자대학교 산업디자인과 리서치 학회
- 2023 체코 오스트라바대학교 시각디자인과 교환학생

PROJECTS

Morib: SaaS형 생산성 서비스 릴리즈

<https://www.morib.in/>

Trabble: 나와의 채팅으로 남기는 여행기록' 앱 릴리즈

<https://apps.apple.com/app/id6745779775>

OLIVE YOUNG CONTENTS.

5 PROJECTS

Design • AI • UX/UI

01 **Toss Place, POS 디자인시스템 구축**
AI - Native / 디자인시스템 / 자동화

02 **가비아 하이웍스, 디자인시스템 구축**
디자인시스템 / B2B / 모바일

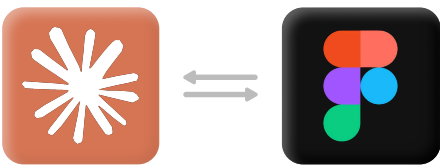
03 **Morib: SaaS형 생산성 서비스 릴리즈**
SaaS / 생산성 / 데스크톱

04 **SOPT: IT 동아리 앱, 모임 개설 UX 개선**
운영 / UX 리서치 / 모바일

05 **Trabble: 채팅 기반 여행 기록 앱 릴리즈**
릴리즈 / UX 리서치 / 모바일

Claude AI 디자인 시스템 자동화

Toss Place, POS 디자인 시스템 구축



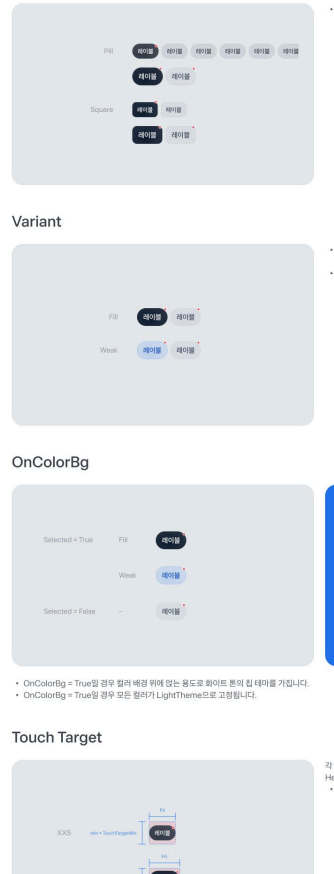
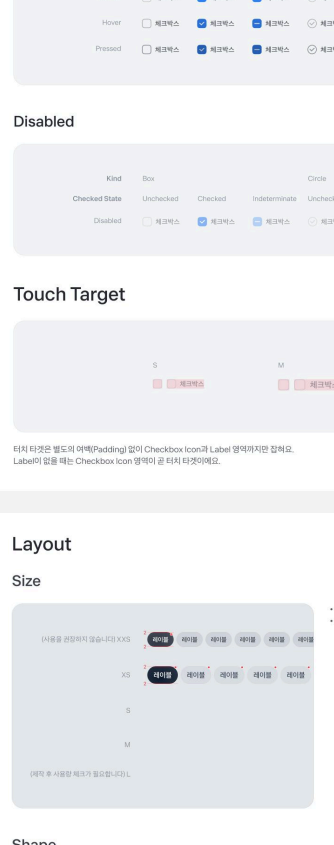
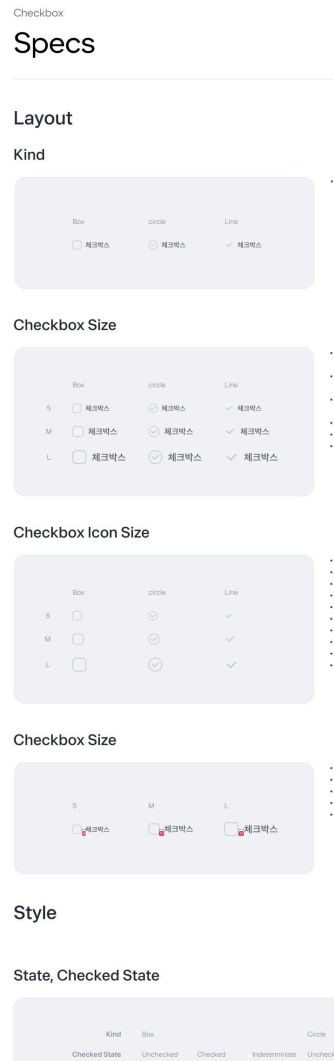
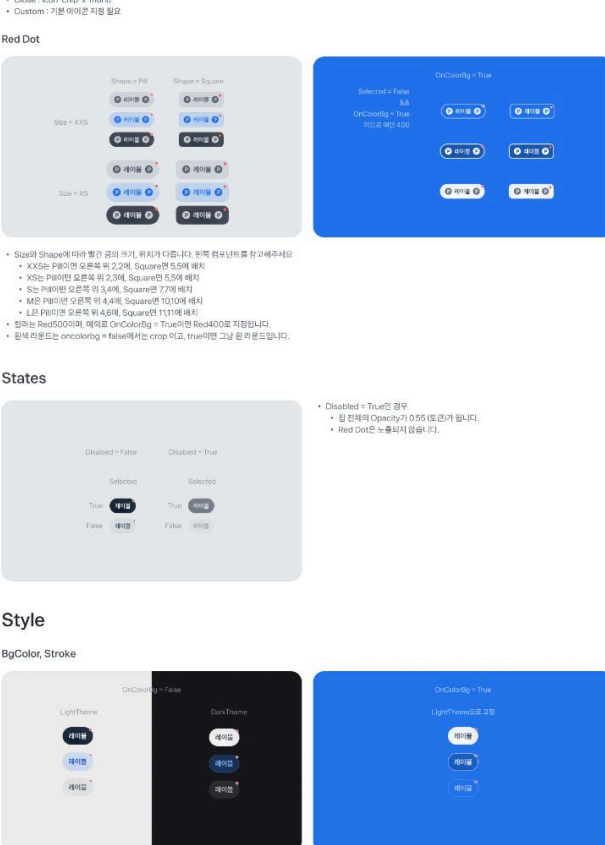
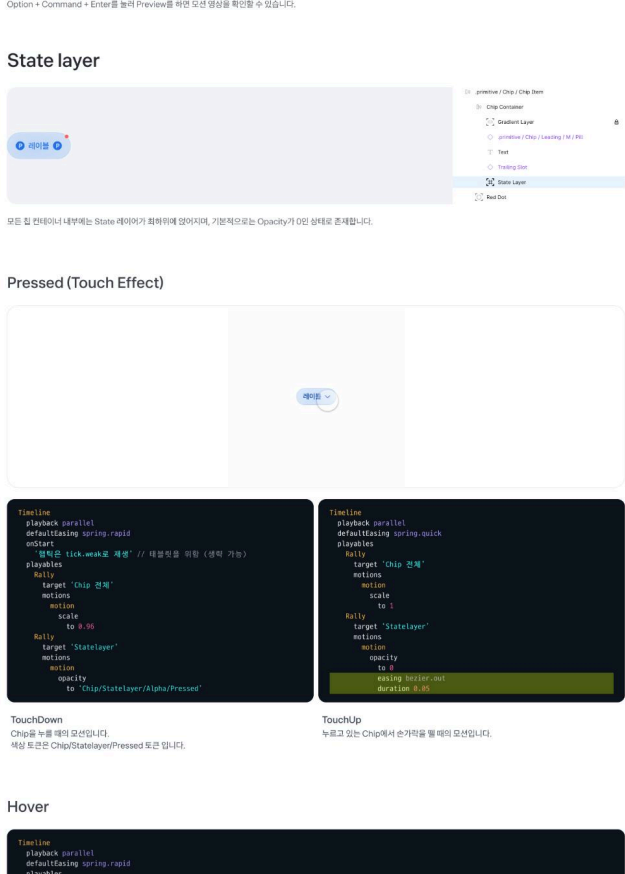
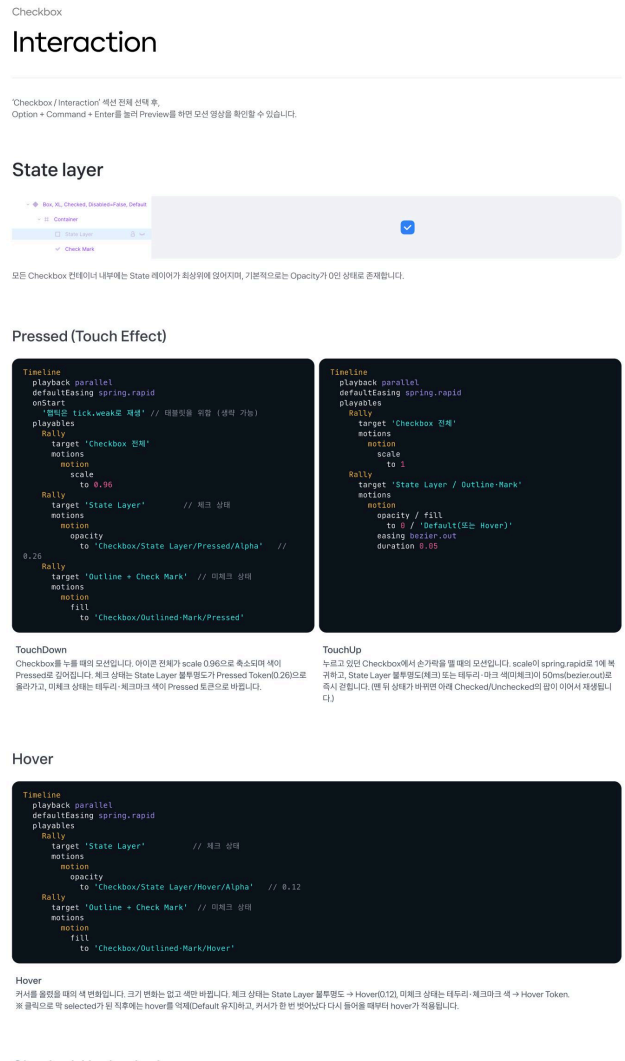
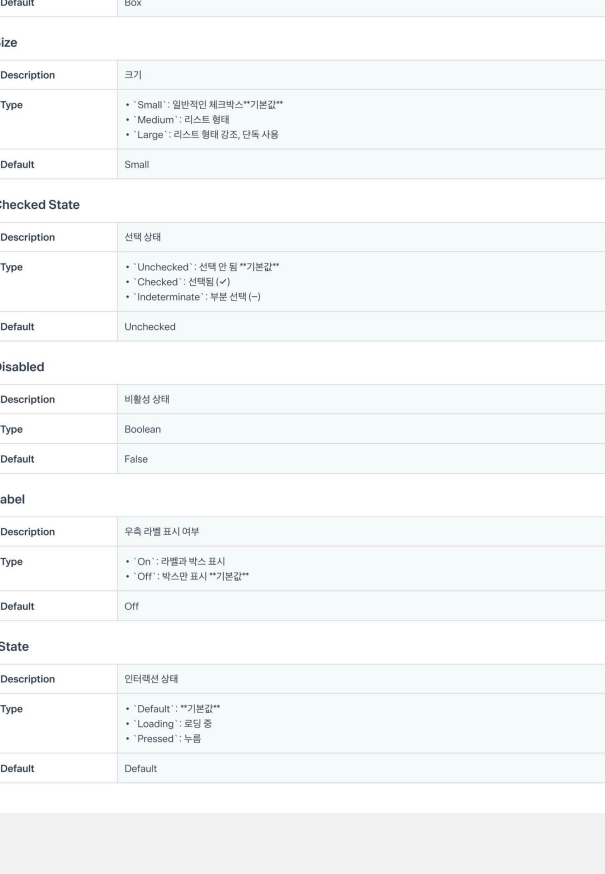
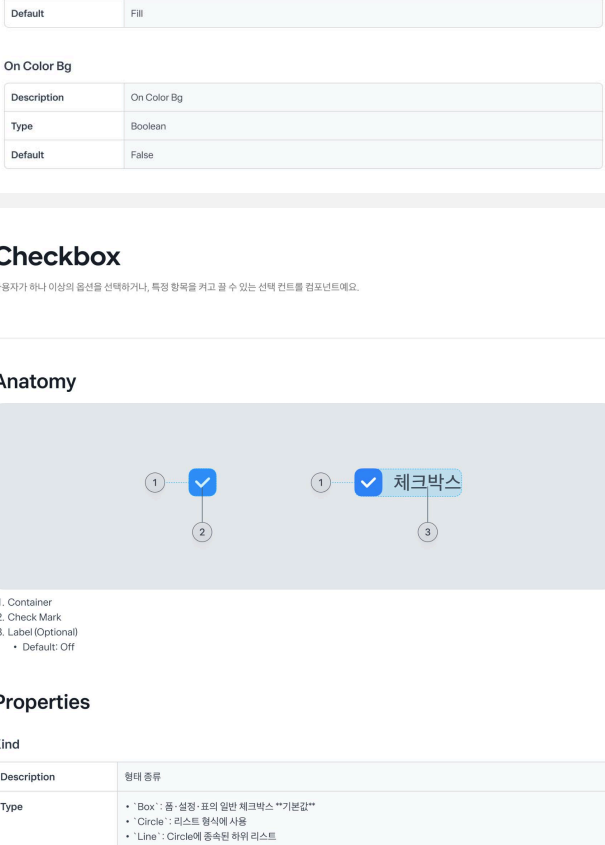
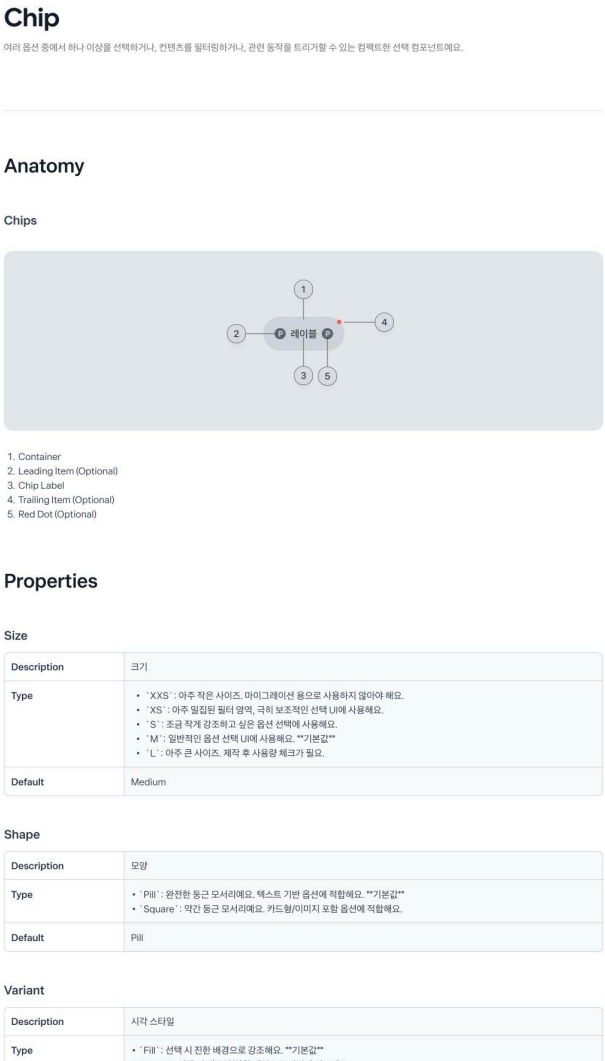
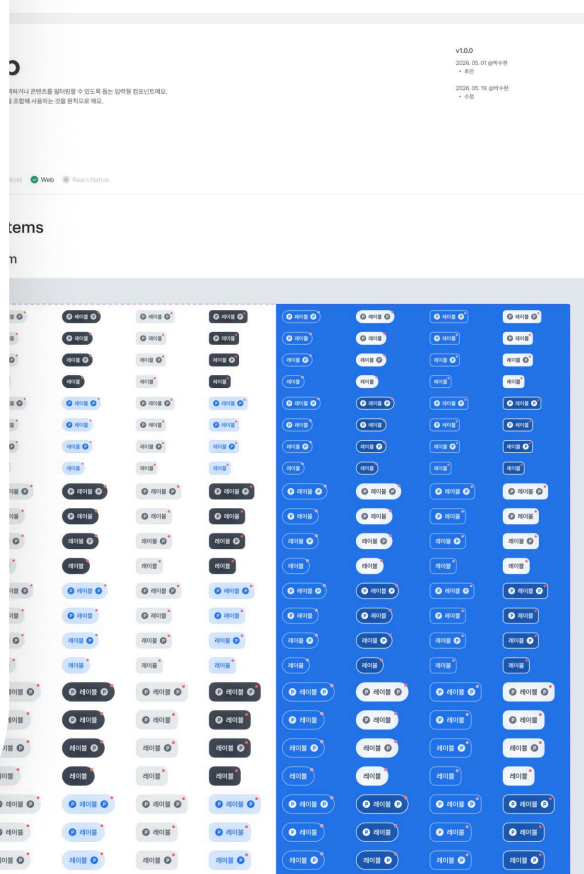
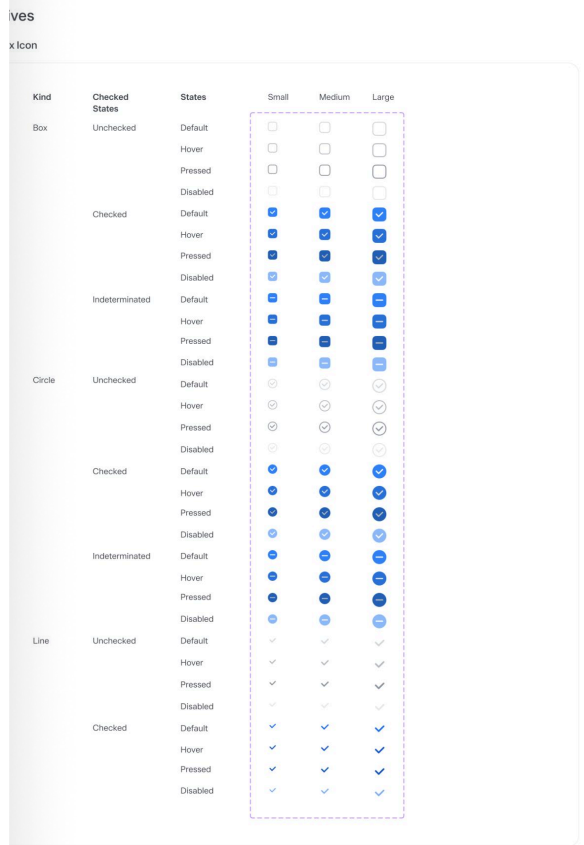
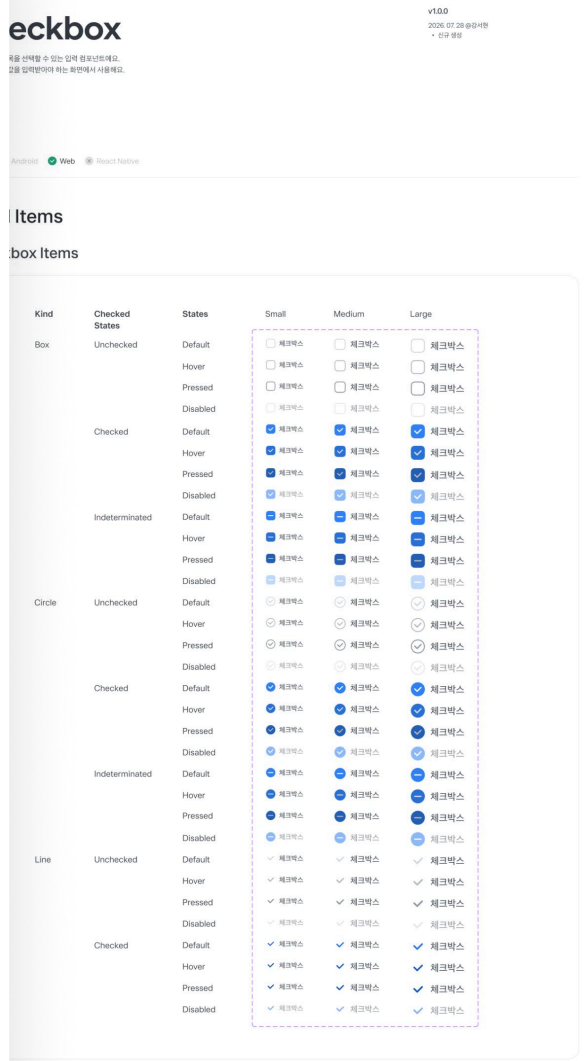
Toss Place POS UI에는 3개의 디자인 시스템이 섞여 있었습니다. 이를 1개의 디자인시스템으로 통합하며 **마이그레이션**을 진행했습니다.

이 과정에서 반복되는 변수 등록과 문서화 작업은 **Claude의 MCP 기반 Skill**로 자동화하며, 컴포넌트 하나당 8시간 걸리던 정리 작업을 1시간으로 줄일 수 있었습니다.

기간 | 2026.06 - 진행중

프로젝트 참여 인원 | 디자인 2명

디자인 시스템 구축 기여도 | 70%



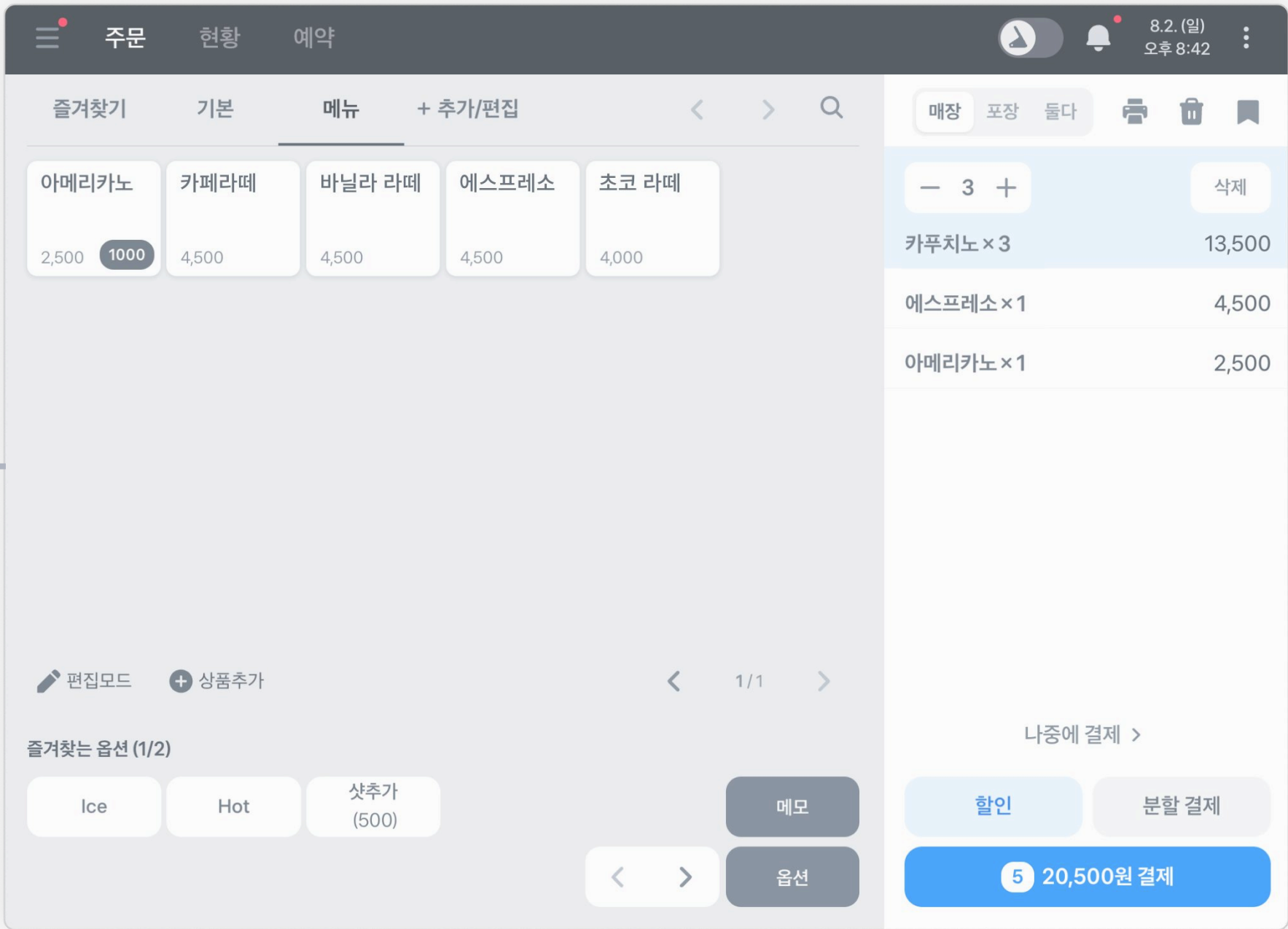
왜 toss place, POS만의 시스템이 필요했을까?

하나의 화면에 3개의 디자인시스템이 섞여 있었어요. 기준이 될 수 있는 'toss place POS만의 디자인시스템' 1개가 필요했어요.



! 1개의 화면을 구성한 3개의 패키지(디자인시스템)

- @toss/mobile-design system (Mobile 디자인시스템)
- @toss/desktop-trading-design system (증권 디자인시스템)
- @toss/desktop-design system (PC 디자인시스템)



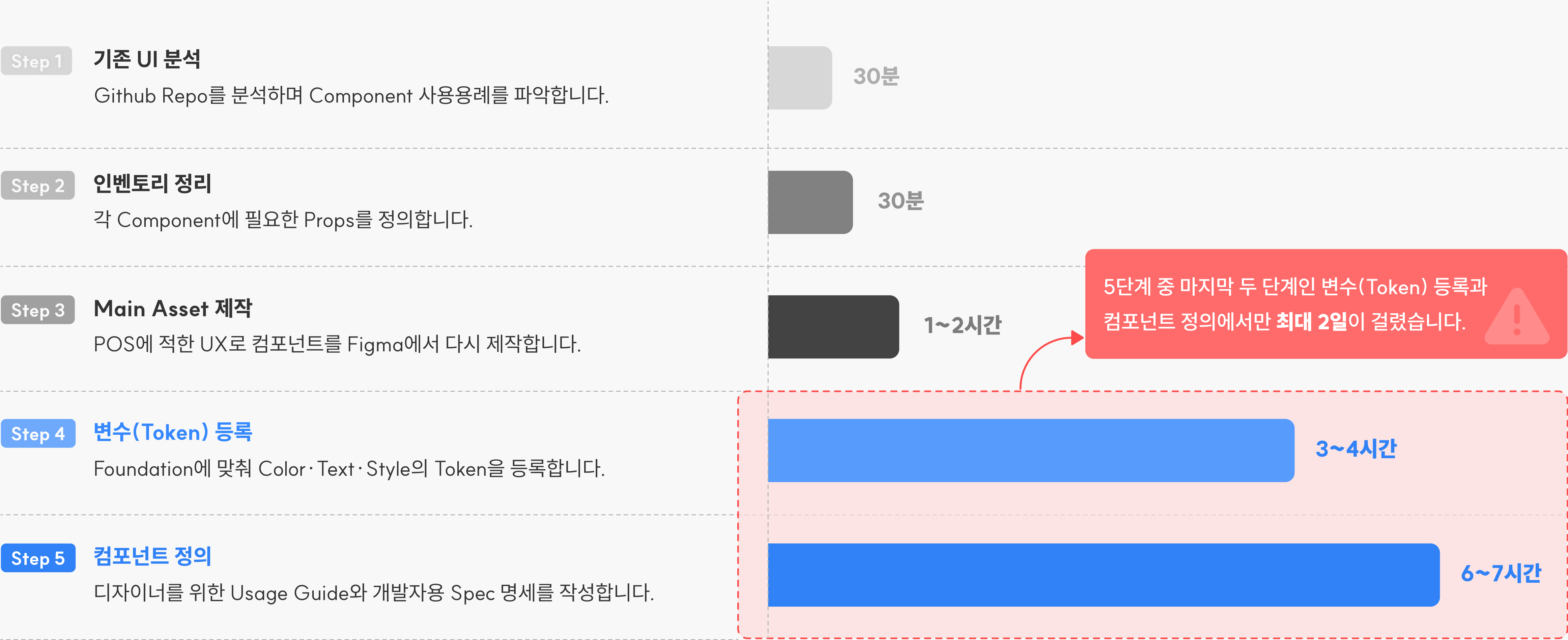
왜 POS만의 시스템이 필요했을까?

3개의 패키지(디자인시스템)를 1개의 패키지로 이동시키는 ‘디자인시스템 마이그레이션’을 진행했어요.



이런 순서로 컴포넌트를 만들었어요!

기존 UI 분석부터 컴포넌트 정의까지, 5단계로 마이그레이션을 진행했어요.
그런데 에셋을 제작한 후에 '변수(Token) 등록'과 '컴포넌트 정의' 두 단계에 많은 시간이 걸렸어요.



시간이 오래 걸리는 문제가 있었어요!

즉 컴포넌트를 만드는 시간보다, 변수를 등록하고 문서를 작성하는데에 더 많은 시간이 걸렸어요.
그래서 Claude Skill을 활용하여 이 단계를 자동화시켰어요.

AS IS Problem

끝나지 않는 변수 등록

😞 “피그마에서 변수 등록만 3시간째 하고 있어요...”

정의된 규칙에 따라 컴포넌트의 속성마다 변수(Variables)를 하나씩 연결해야 했습니다. 컴포넌트 하나에 많게는 40~50개를 손수 연결하다 보니, 끝내는 데만 3~4시간이 걸렸습니다.



UX Solution

변수 등록과 검수 자동화 Skill.md

AS IS Problem

컴포넌트 제작보다 오래 걸리는 문서 작성

😞 “컴포넌트는 빨리 제작하는데, 가이드 쓰기가 너무 어려워요”

컴포넌트를 만들고 나면 사용 가이드와 스펙을 문서로 작성했습니다. 가이드를 쓸 때마다 토스의 문체와 형식이 맞는지 하나하나 확인해야 했습니다. 작은 컴포넌트도 4~5시간, 상태가 많은 컴포넌트는 2일이 걸렸습니다.



UX Solution

Guide·Spec 문서 작성 자동화 Skill.md

[문제] 컴포넌트 하나에 변수만 수십 개, 하나씩 등록하다 보면 빠지거나 잘못 연결되는 문제가 있었어요.

Claude에 Figma MCP를 연결해, 하나하나 직접 등록하던 변수(Variables)를 자동으로 등록하고, 검수 리포트까지 보여주는 Skill을 만들었어요.



[Claude]

프롬프트 Checkbox 변수(Variable) 등록해줘.

<https://www.figma.com/design/gwuiFIJgVxx6g2...>

[/tds-pos-component-tokens.skill](#)

← 스킬

tds-pos-component-tokens

사용자 제공

TDS-POS Figma 파일(2. TDS - POS 및 그 브랜치)에서 마스터 컴포넌트(Main asset)의 Variables Style을 규칙에 맞게 생성/적용/점검하는 스킬. 사용자가 Figma 컴포넌트에 "토론 적용", "변수 정리", "스타일 등록", "raw 값/hex 정리", "Semantic 매핑", "규칙에 맞는지 점검/검수"를 요청하... [더 보기](#)

SKILL.md

3 개 파일

TDS-POS 컴포넌트 토큰화 & 점검

TDS-POS 디자인시스템에서 마스터 컴포넌트의 색·크기·타이포 값을 Base → Semantic → Component 3층 토큰 체계로 정리하고, 스타일로 묶어 컴포넌트에 적용한 뒤 검증하는 작업 절차와 규칙.

두 가지 모드

- 토큰화(tokenize): 사용자가 만든 컴포넌트의 raw/hex/원격 값을 규칙에 맞는 로컬 변수·스타일로 교체하고 마무리한다.
- 점검(audit): 읽기 전용으로 규칙 위반을 스캔해 리포트만 낸다.

상세 규칙은 `references/rules.md`, Figma Plugin API 스크립트 템플릿은 `references/scripts.md`를 읽는다. 아래는 절대 어기면 안 되는 핵심만 요약.

왜 이 구조인가 (배경)

- 최종 목표는 다크모드 원클릭 전환. Figma alias는 특정 모드를 고정 참조(mode pinning)할 수 없고, 모드별 해석은 같은 컬렉션 안에서만 동작한다. 그래서 원격 라이브러리(0. TDS POS Foundation) 변수를 직접 참조하면 항상 기본 모드(Light)로 풀린다 → 모든 참조는 로컬 변수로 한다 (로컬 복제 원칙).
- Base=순수 수치/색 팔레트(이름=값), Semantic=용도가 이름에 있는 토큰, Component=특정 컴포넌트 전용. 값 수정 파급을 Semantic 한 곳에서 제어하기 위함.

핵심 규칙 (요약)

- 참조 체인: 레이아웃/스타일 → Component 변수 → Semantic 변수 → Base 변수

MCP, 변수 연결
← 검수 리포트 제출



[Figma]

Step 1 기존 컴포넌트에 등록된 raw/hex·외부 변수를 읽어오.

Step 2 자동으로 Base > Semantic > Component 3단계 규칙으로 변수를 생성하고 매핑해요.

Step 3 검수 리포트에서 변수 매핑 결과와 규칙에 어긋난 항목 등을 보여줘요.

Variables

Collections

Color

Effect

Scale

Screen

118N

Groups

All

Base

Semantic

Component

Scroll Area

Button

Numeric Spinner

Text Button

State Layer

Fill

Pressed

Grey

Blue

Hover

523

65

716

8

12

523

138

53

331

6

6

25

10

6

6

2

1

1

2

Name

Light

Dark

Component / Checkbox / Fill

Default

Semantic/Fill/Brand

Semantic/Fill/Brand

Component / Checkbox / State Layer / Hover

Fill

Alpha

...mantic/State Layer/Default

...mantic/State Layer/Default

...c/State Layer/Alpha/Hover

...c/State Layer/Alpha/Hover

Component / Checkbox / State Layer / Pressed

Fill

Alpha

...mantic/State Layer/Default

...mantic/State Layer/Default

...State Layer/Alpha/Pressed

...State Layer/Alpha/Pressed

Component / Checkbox / Mark

Default

Hover

Pressed

Semantic/Icon/Unselected

Semantic/Icon/Unselected

... Opacity/Grey Opacity 400

... Opacity/Grey Opacity 400

... Opacity/Grey Opacity 500

... Opacity/Grey Opacity 500

Component / Checkbox / Mark / Brand

Default

Hover

Pressed

Semantic/Icon/Brand

Semantic/Icon/Brand

Base/Blue/Blue 800

Base/Blue/Blue 800

0F54B7

Base/Blue/Blue 900

Create variable

Selection colors

Component/Checkbox/Text/Text

Component/Checkbox/Fill/Default

Component/Checkbox/Mark/On Fill B...

_Dummy

Component/Checkbox/State Layer/Pr...

Component/Checkbox/State Layer/H...

Component/Checkbox/Mark/Default

Component/Checkbox/Outlined/Defa...

Component/Checkbox/Outlined/Pres...

Component/Checkbox/Mark/Pressed

Component/Checkbox/Mark/Hover

Component/Checkbox/Outlined/Hover

Component/Checkbox/Mark/Brand/D...

Component/Checkbox/Mark/Brand/Pr...

Component/Checkbox/Mark/Brand/H...

[문제] 컴포넌트를 만들고 가이드와 스펙 작성하는데에 너무 과도하게 많은 시간이 걸렸어요.

기존 컴포넌트의 문서를 가지고, Guide·Spec 공통 구조를 정리했어요.

컴포넌트를 Claude에 넘기면, Skill이 Guide·Spec 문서를 .md 파일로 만들어요.



[Claude]

프롬프트 Checkbox Guide·Spec 문서 작성해줘.

<https://www.figma.com/design/gwuiFIJgVxx6g2...>

</tds-component-docs.skill>

- 1 Figma에서 컴포넌트 구조·Variant를 읽어요.
- 2 Skill에 정의된 공통 구조에 맞춰 .md 문서를 작성해요.
- 3 완성된 .md를 Figma 문서 페이지로 옮겨 배포해요.

Checkbox docs

Checkbox

여러 항목을 동시에 선택하거나, 한 항목을 켜고 끌 수 있게 하는 컴포넌트예요. 선택지가 서로 배타적이지 않고 0개부터 전체까지 자유롭게 고를 수 있을 때 사용해요.

When to use

Use Checkbox:

- 목록에서 여러 항목을 동시에 선택해야 할 때 사용해요. (예: 배달 결제수단 다중 선택, 매출리포트 필터)
- 선택 결과가 바로 적용되지 않고 저장·확인 같은 후속 액션을 거칠 때 사용해요.
- 약관 동의처럼 하나의 항목에 동의/비동의를 표시해야 할 때 사용해요.
- 상위 항목이 하위 항목 전체를 켜고 끄는 전체 선택(부모-자식) 구조를 만들 때 사용해요.
- 테이블-리스트에서 여러 행을 골라 일괄 처리(batch action)할 때 사용해요.

Don't use Checkbox:

- 옵션 중 하나만 선택해야 하는 경우 사용하지 마세요. 대신 **Radio**를 사용해요.
- 선택 즉시 기능이 켜지고 꺼지는 단일 설정에는 사용하지 마세요. 대신 **Switch(Toggle)****를 사용해요.
- 선택 즉시 화면이 전환되거나 팝업이 뜨는 경우 사용하지 마세요. 그런 선택이 아니라 액션이므로 **Button**을 사용해요.
- 여러 항목을 고르되 선택지가 가로로 나열되는 태그 형태라면 사용하지 마세요. 대신 **Chip**을 검토해요.

Anatomy

1. **Icon (Container)**: 선택 상태를 표시하는 영역이에요. Kind에 따라 사각형(Box)-원형(Circle)이 되고, Line은 컨테이너 없이 마크만 표시돼요.
2. **Mark**: Container 안에 표시되는 마크예요. Checked는 체크(✓), Indeterminate는 가로선(-)이에요.
3. **State Layer**: Hover/Pressed 시 Container 위에 겹쳐지는 오버레이예요. Box-Circle에만 적용돼요.
4. Label (Optional): 항목의 이름을 표시하는 텍스트예요. Label 프로퍼티로 숨길 수 있어요.
5. **Focus ring**: 키보드 포커스를 표시하는 테두리예요. Variant가 아니라 Effect Style 토크로 적용해요.

Button

Usage

사용 규칙

Use Checkbox

- 목록에서 여러 항목을 동시에 선택해야 할 때 사용해요. (예: 배달 결제수단 다중 선택, 매출리포트 필터)
- 선택 결과가 바로 적용되지 않고 저장·확인 같은 후속 액션을 거칠 때 사용해요.
- 약관 동의처럼 하나의 항목에 동의/비동의를 표시해야 할 때 사용해요.
- 상위 항목이 하위 항목 전체를 켜고 끄는 전체 선택(부모-자식) 구조를 만들 때 사용해요.
- 테이블-리스트에서 여러 행을 골라 일괄 처리(batch action)할 때 사용해요.

Kind



- Box: 기본값이에요. 폼·레이블 등 일반적인 선택에 사용해요.
- Circle: 리스트 항목의 선택 표시(indicator)에 사용해요.
- Line: 배경이 있는 영역 위나, 상위 항목에 종속된 하위 목록의 선택 표시에 사용해요.

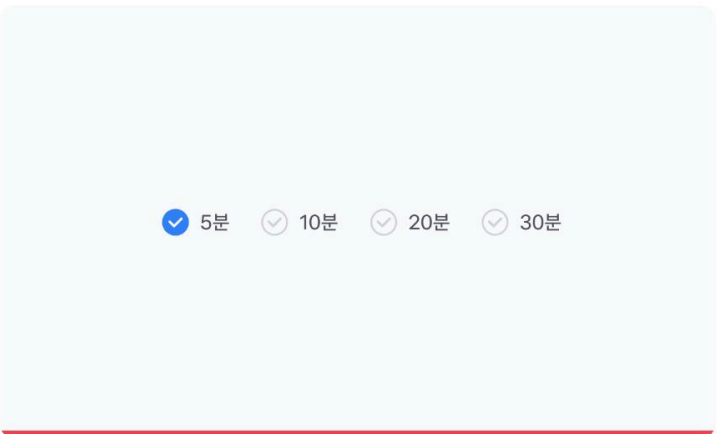
Label



- Checkbox를 단독으로 사용할 때는 Label을 항상 노출해요. 무엇을 선택하는지 Checkbox 스스로 설명해야 해요.
- List·Table 내부의 선택 표시로 쓸 때만 Label 없이(Label=False) 사용해요. 이때 항목의 이름은 행(List)이 제공해요.
- 체크 박스 단독 사용시에는, 'primitive / Checkbox / Icon'을 사용하는 것이 아닌 'Checkbox'의 Label을 False두고 사용해요.

사용시 주의할 점

Checkbox 단독 사용



Don't

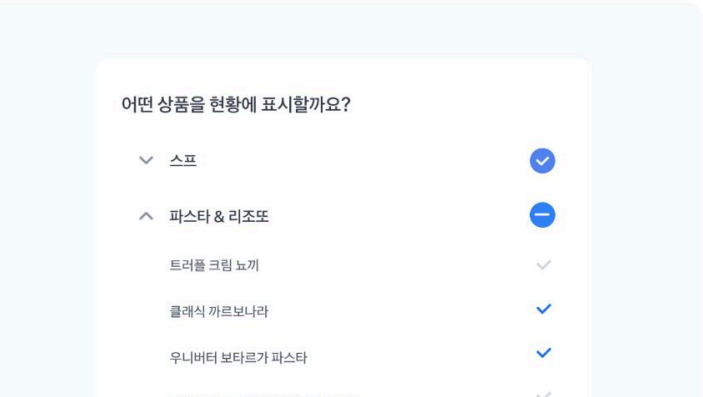
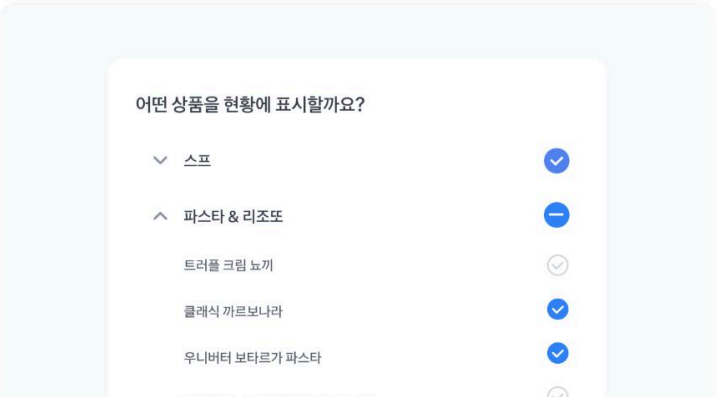
Checkbox를 단독으로 배치해 사용하는 것을 지양해요. 터치 타겟이 icon과 Label 영역뿐이라 최소 터치 타겟(48px)에 미치지 못해요.



Do

단독 사용은 지양하고, 행 전체가 클릭 영역인 List(우측 Checkbox 형태)나 Chip, Toggle으로 대체해서 사용해요.

Line Checkbox



문서 작성 자동화

이렇게 달라졌어요!

Skill 2개로 컴포넌트 제작과 문서화를 자동화하며, 반복 작업을 줄일 수 있었어요.

컴포넌트 개수

재정의 컴포넌트, 51개

- POS 기준으로 재정의한 컴포넌트 51개

패키지

패키지, 3개 > 1개

- POS만의 1개의 디자인시스템 패키지 제작

커버리지

신규 화면의 84%가 POS DS로 구성

- 신규 화면 12개 기준 · 잔여 17%는 예외 케이스

스킬 제작 후, 소요 시간

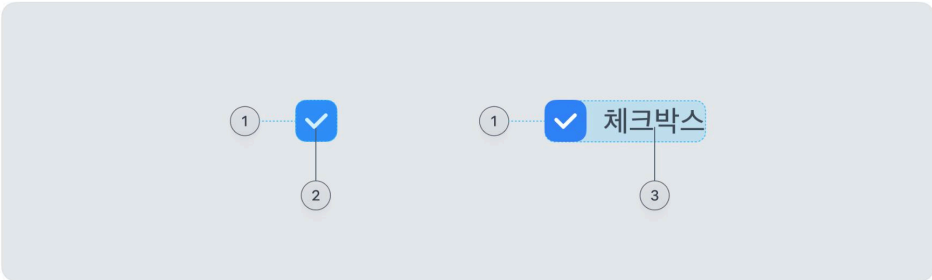
컴포넌트 1개 제작, 13시간 > 2시간

- 구조 정리·변수 등록·문서화 등, 컴포넌트 제작 시간

Checkbox

사용자가 하나 이상의 옵션을 선택하거나, 특정 항목을 켜고 끌 수 있는 선택 컨트롤 컴포넌트예요.

Anatomy



1. Container
2. Check Mark
3. Label (Optional)
 - Default: Off

Properties

Kind	
Description	형태 종류
Type	• 'Box': 폼·실장·표의 일반 체크박스""기본값"" • 'Circle': 리스트 형식에 사용 • 'Line': Circle에 종속된 하위 리스트
Default	Box

Size	
Description	크기
Type	• 'Small': 일반적인 체크박스""기본값"" • 'Medium': 리스트 형태 • 'Large': 리스트 항목 강조, 단독 사용
Default	Small

Checked State	
Description	선택 상태
Type	• 'Unchecked': 선택 안 됨""기본값"" • 'Checked': 선택됨 (✓) • 'Indeterminate': 부분 선택 (-)
Default	Unchecked

Disabled	
Description	비활성 상태
Type	Boolean
Default	False

Label	
Description	우측 라벨 표시 여부

Checkbox

하나 이상의 항목을 선택할 수 있는 입력 컨트롤예요.
사용자의 선택 값을 입력받아야 하는 화면에서 사용해요.

iOS Android Web React Native

Child Items

Checkbox Items

		v1.0.0 2026.07.28 송광서현 · 신규 변경				
Kind	Checked States	States	Small	Medium	Large	
Box	Unchecked	Default	☐	☐	☐	☐
		Hover	☐	☐	☐	☐
		Pressed	☐	☐	☐	☐
		Disabled	☐	☐	☐	☐
	Checked	Default	☒	☒	☒	☒
		Hover	☒	☒	☒	☒
		Pressed	☒	☒	☒	☒
		Disabled	☒	☒	☒	☒
	Indeterminate	Default	☐	☐	☐	☐
		Hover	☐	☐	☐	☐
		Pressed	☐	☐	☐	☐
		Disabled	☐	☐	☐	☐
Circle	Unchecked	Default	☐	☐	☐	☐
		Hover	☐	☐	☐	☐
		Pressed	☐	☐	☐	☐
		Disabled	☐	☐	☐	☐
	Checked	Default	☒	☒	☒	☒
		Hover	☒	☒	☒	☒
		Pressed	☒	☒	☒	☒
		Disabled	☒	☒	☒	☒
	Indeterminate	Default	☐	☐	☐	☐
		Hover	☐	☐	☐	☐
		Pressed	☐	☐	☐	☐
		Disabled	☐	☐	☐	☐
Line	Unchecked	Default	☐	☐	☐	☐
		Hover	☐	☐	☐	☐
		Pressed	☐	☐	☐	☐
		Disabled	☐	☐	☐	☐
	Checked	Default	☒	☒	☒	☒
		Hover	☒	☒	☒	☒
		Pressed	☒	☒	☒	☒
		Disabled	☒	☒	☒	☒
	Indeterminate	Default	☐	☐	☐	☐
		Hover	☐	☐	☐	☐
		Pressed	☐	☐	☐	☐
		Disabled	☐	☐	☐	☐

Primitives

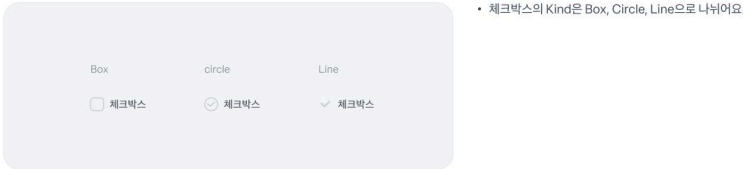
Checkbox Icon

Checkbox

Specs

Layout

Kind



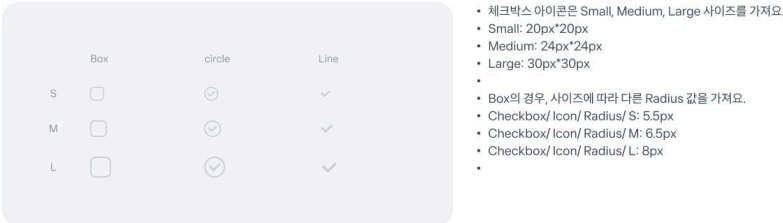
- 체크박스의 Kind은 Box, Circle, Line으로 나뉘어요.

Checkbox Size



- 체크박스는 Small · Medium · Large 사이즈를 가져요.
- Label을 노출할 때는 Checkbox의 Label을 Hug × Hug로 감싸요.
- Label 없이 사용할 때는 내부 Checkbox icon의 크기를 최소값(Min Width × Min Height)으로 가져요.
- Checkbox/Min Width · Height/ S: 30px*20px
- Checkbox/Min Width · Height/ M: 24px*24px
- Checkbox/Min Width · Height/ L: 30px*30px

Checkbox Icon Size



- 체크박스 아이콘은 Small, Medium, Large 사이즈를 가져요.
- Small: 20px*20px
- Medium: 24px*24px
- Large: 30px*30px
- Box의 경우 사이즈에 따라 다른 Radius 값을 가져요.
- Checkbox/Icon/Radius/ S: 5.5px
- Checkbox/Icon/Radius/ M: 6.5px
- Checkbox/Icon/Radius/ L: 8px

Checkbox Size



- Checkbox icon과 Label 사이의 Gap은 사이즈에 따라 다른 값을 가져요.
- Small Gap: 8px
- Medium Gap: 10px
- Large Gap: 12px

Style

State, Checked State

		Kind							
		Box	Circle		Line				
Checked State	Unchecked	Checked	Indeterminate	Unchecked	Checked	Indeterminate	Unchecked	Checked	
Default	☐	☒	☐	☐	☒	☐	☐	☒	☐
Hover	☐	☒	☐	☐	☒	☐	☐	☒	☐
Pressed	☐	☒	☐	☐	☒	☐	☐	☒	☐

Checkbox

Interaction

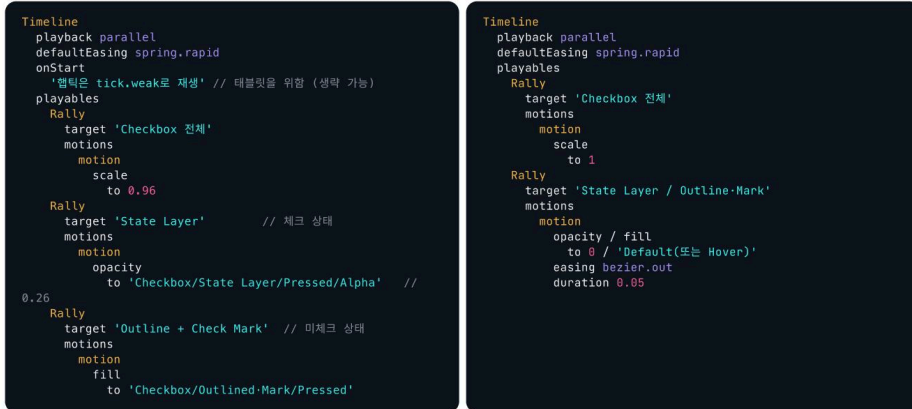
'Checkbox/Interaction' 세션 전체 선택 후,
Option + Command + Enter를 눌러 Preview를 하면 모션 영상을 확인할 수 있습니다.

State layer



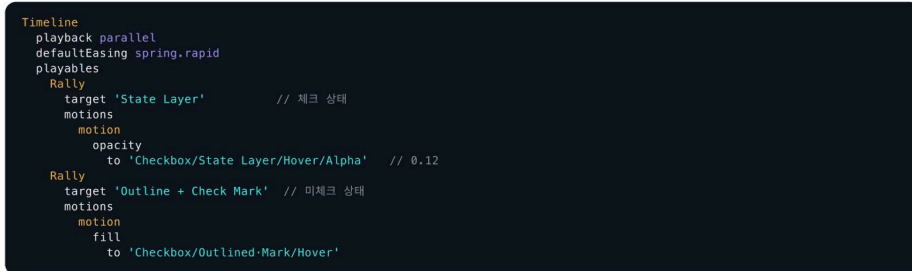
모든 Checkbox 컨테이너 내부에는 State 레이어가 최상위에 위치하며, 기본적으로는 Opacity가 0인 상태로 존재합니다.

Pressed (Touch Effect)



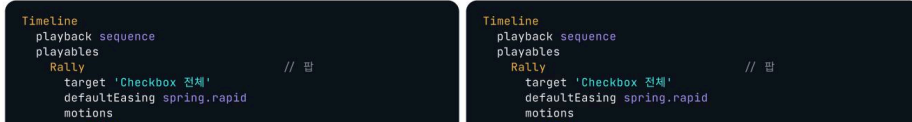
TouchDown
Checkbox을 누를 때와 모션입니다. 아이콘 전체가 scale 0.95으로 축소되며 색이 바뀌고, State Layer 불투명도(오프) 또는 테두리·마크 색이 체크(50msbezier.out)로 올라가고, 마크 상태는 테두리·체크마크 색이 Pressed 토큰(0.26)으로 바뀝니다.

Hover



Hover
커서를 움직일 때와 색 변화입니다. 크기 변화는 없고 색만 바뀝니다. 체크 상태는 State Layer 불투명도 → Hover(0.12), 마크 상태는 테두리·체크마크 색 → Hover Token.
※ 클릭으로 즉 selected가 된 직후에는 hover를 지켜Default 유지하고, 커서가 한 번 벗어났다 다시 들어올 때부터 hover가 적용됩니다.

Checked, Unchecked

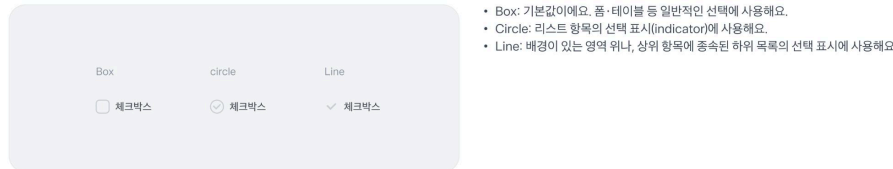


Button

Usage

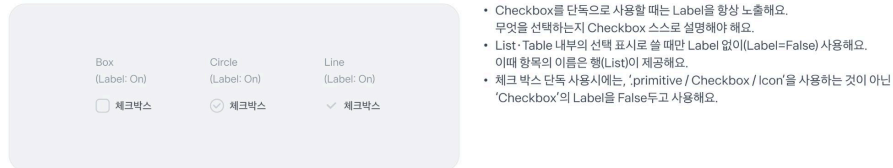
사용 규칙

Kind



- Box: 기본값이예요. 폼·타이틀 등 일반적인 선택에 사용해요.
- Circle: 리스트 항목의 선택 표시(Indicator)에 사용해요.
- Line: 배경이 있는 영역 뒤나, 상위 항목에 종속된 하위 항목의 선택 표시에 사용해요.

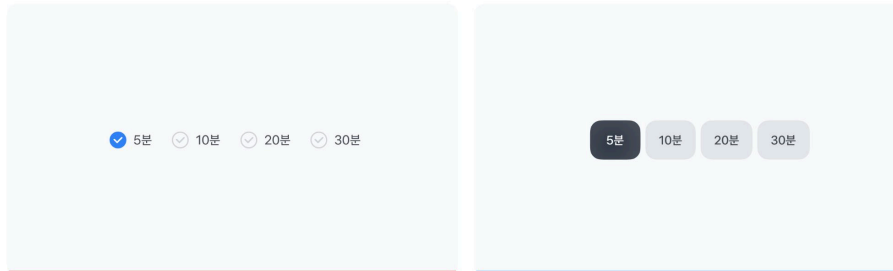
Label



- Checkbox을 단독으로 사용할 때는 Label을 항상 노출해요.
위치를 선택하는지 Checkbox 스스로 설명해야 해요.
- List·Table 내부의 선택 표시로 쓸 때만 Label 없이(Label=False) 사용해요.
이때 항목의 이름은 행(Last)이 제공해요.
- 체크 박스 단독 사용시에는 'primitive / Checkbox / Icon'을 사용하는 것이 아닌 'Checkbox'의 Label을 False로 사용해요.

사용시 주의할 점

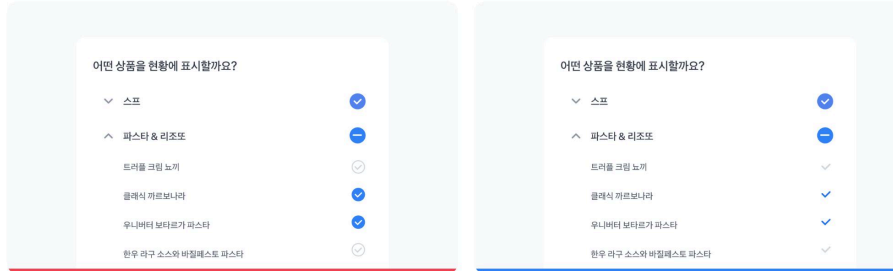
Checkbox 단독 사용



Don't
Checkbox을 단독으로 배치해 사용하는 것을 지양해요.
타치 타겟이 Icon과 Label 명칭뿐이라 최소 터치 타겟(48px)에 미치지 못해요.

Do
Line을 사용은 지양하고, 행 전체가 클릭 영역인 List우측 Checkbox 형태나
Chip, Toggle으로 대체해서 사용해요.

Line Checkbox



Don't
Line을 단독으로 배치하지 마세요.
배경 면이 없이 선택 요소로 인지되기 어렵고, 선택 여부가 세로로만 구분돼요.

Do
Line은 List 행 내부의 선택 표시나 상위 항목에 종속된 하위 항목에서만,
주변 맥락이 선택 대상을 설명해 주는 곳에서 사용해요. 단독 사용에는 Box를 사용해요.

Result

아래와 같이 51개의 컴포넌트를 제작하고 문서화해 팀 내에 배포하여 사용하고 있어요.

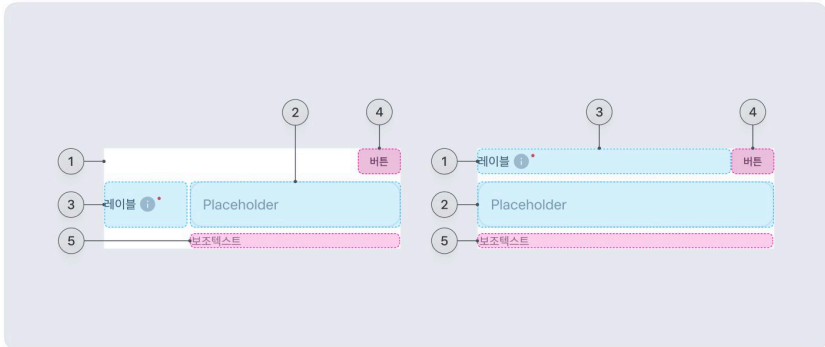
컴포넌트의 Anatomy, Props, Spec, Interaction, Usage를 체계적으로 나누어 문서화했어요.

Text Field

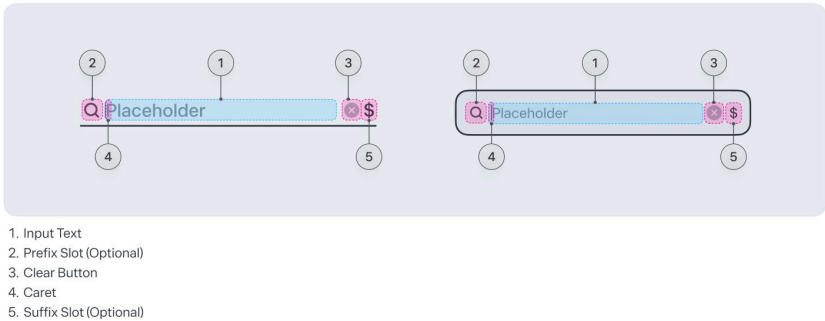
사용자가 텍스트를 직접 입력하거나 수정하는 컴포넌트예요. 레이블·입력 박스·보조 텍스트를 하나의 품 항목으로 묶어서 제공해요.

Anatomy

Text Field



Input



Properties

Size	
Description	크기
Type	- XS: xSmall 사용하지 않음 (미리그레이션 용) - XS: xSmall 목록 내 인라인 입력 - S: Small - M: Medium, "기본값" - L: Large, 제목 입력 - XL: xLarge, 히어로 입력
Default	M

Variant

Description	외형 스타일
Type	- Box: 모든 방향으로 테두리가 있으며 흰색 배경인 스타일, "기본값" - Line: 하단에만 테두리가 있고 왼쪽에 값을 강조하는 스타일
Default	Box

State

Description	Text Field 상태
Type	- Default: "기본값" - Error: 값이 유효하지 않음을 알림 - Disabled: 입력 자체가 불가능 - Read Only: 값을 조회만 할 때 사용
Default	Box

Direction

Description	방향
Type	- Column: 레이블이 인풋 위에 위치하는 수직 형태 "기본값" - Row: 레이블이 인풋 왼쪽에 있는 수평 형태
Default	Column

Fill

Description	Text Field Box의 Background Fill
Type	- False: 흰 배경 + 테두리, 기본값 "기본값" - True: 회색 배경, 흰 배경 위에서 입력 영역을 구분해야 할 때

Text Field

사용자가 텍스트를 직접 입력하는 컴포넌트

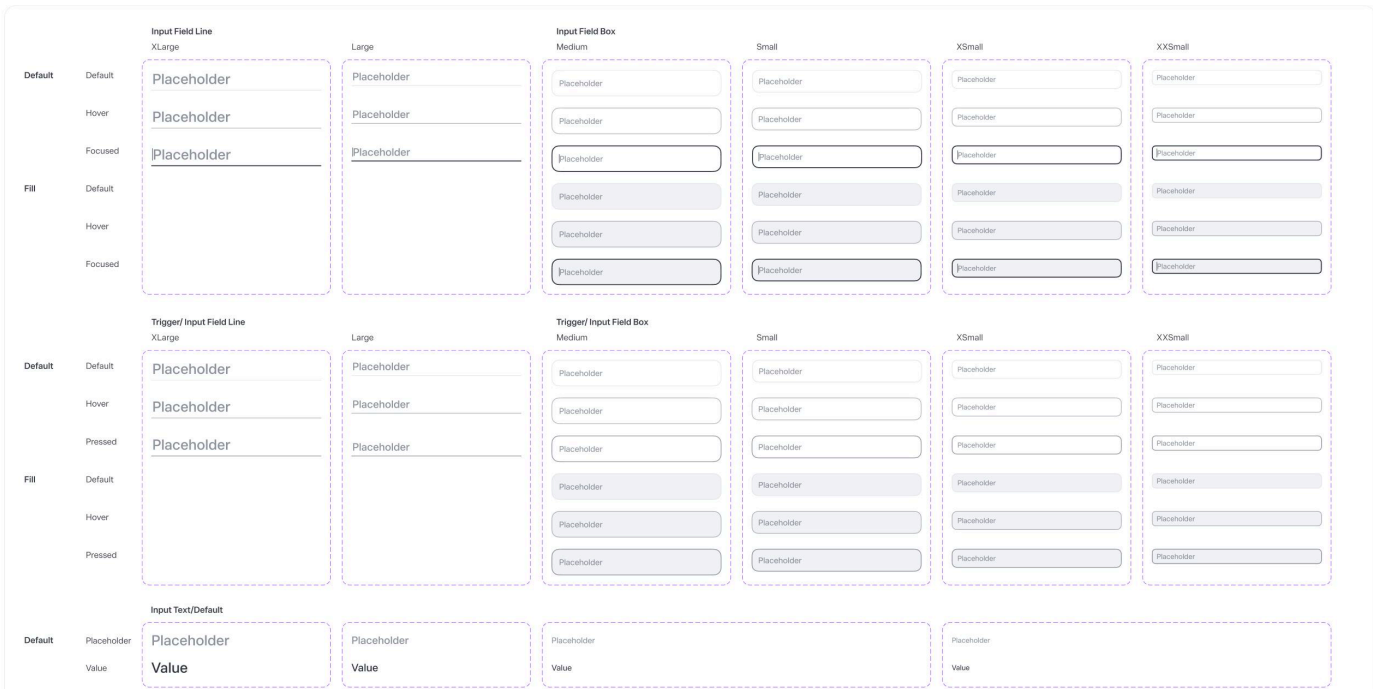
Master Component

Text Field



Child Items

Primitive Items



Text Field

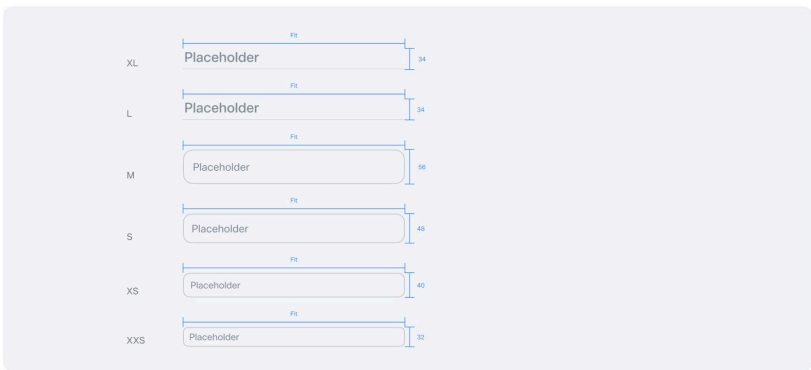
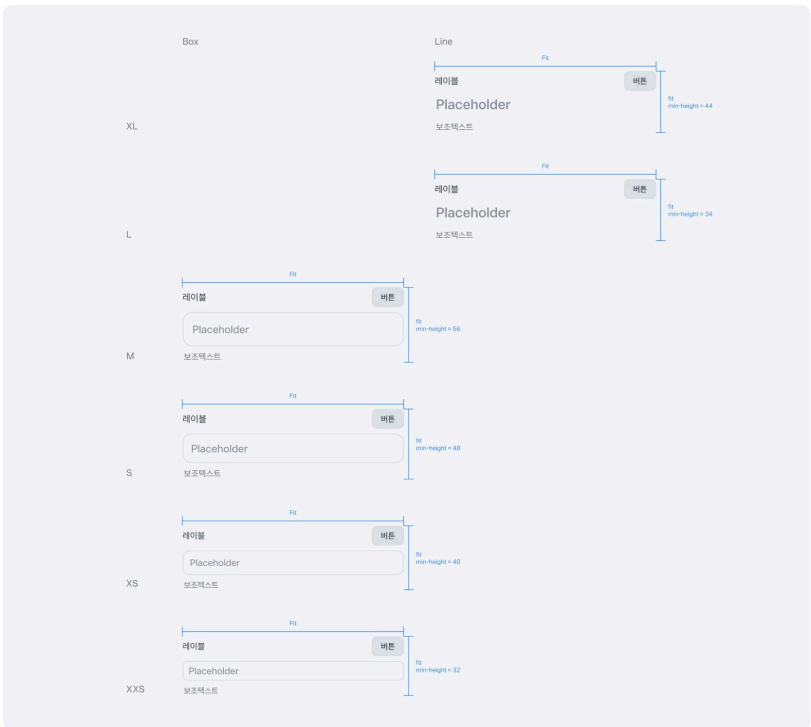
Spec

Variant

- Box: 작은 텍스트를 가진 기본형이에요. M ~ XS까지 사이즈를 지원하고, 패딩을 커진 테두리 대신 배경이 있어요.
- Line: 커진 테두리 대신 배경이 없어요. L, XL 사이즈를 지원하고, Background가 있어요.
- variant: 지원하는 Size 범위 (값이 없는 경우)는 Default/Disabled/ReadOnly만 공통이에요.
- 테두리: 라인은 box-shadow: inset 2px 2px 0px; 상단과 좌측이 수직과 수평이 2px씩 들어, 테두리 안쪽만 들어가지 않아요.
- Default: Hover: 1px, Error: Focus: 2px

Size

- Text Field 높이는 고정값이 아니에요. Input의 높이(min-height)를 기준으로, 내용이 따라 늘어날 수 있어요.
- Input의 높이(min-height):
 - Line XL: 44px
 - Line L: 36px
 - Box M: 56px
 - Box S: 48px
 - Box XS: 40px
 - Box XXS: 32px
- 레이블과 입력 텍스트의 폭은 늘어나지 않아요. 값이 없으면 호퍼가 커지는 항상 일정, 플러스를 제외하지 않아요.
- XS는 레이아웃에 따른 전용 사이즈예요. 신규 화면에서는 사용하지 않아요.



Direction

- Column(레이블 위): Header(레이블)를 기본형 FieldInput+Supporting Text가 바로 들어요.
- Row(레이블 왼쪽): Header(레이블)는 오른쪽으로, 레이블과 Field가 가로로 배치돼요. 레이블은 Input과 서로 동일 방향이에요.
- 레이블 위치는 방향에 관계없이 항상 Input과 붙어 있어요.
- 레이블이 커진 뒤 있는 경우: 레이블은 Label Position + None을 사용해요.

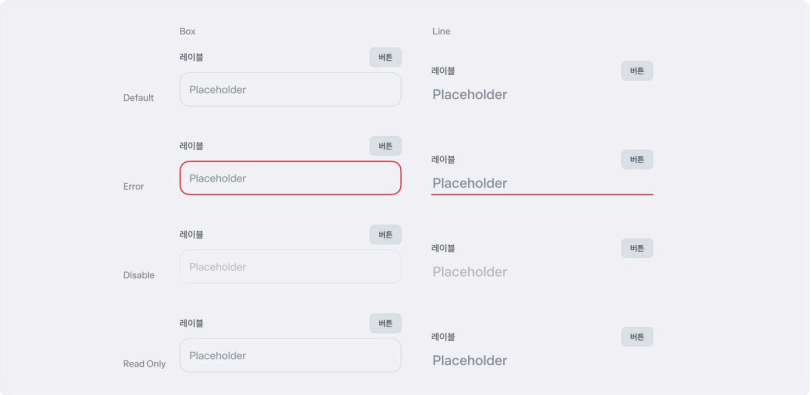
- Label Row Height: 레이블 영역의 높이는 Input 높이에 같은 것으로 고정해요. 레이블 텍스트는 영역의 좌측 상단 정렬이에요. 높이가 Input과 떨어지면 Supporting Text가 생기거나 높이도 레이블이 뒤는 변하지 않아요.
 - XL: 34px
 - L: 36px
 - M: 56px
 - S: 48px
 - XS: 40px
 - XXS: 32px
- Label Column Height: Header가 놓이는 기본 높이값으로 고정해요. 레이블은 그 안에서 좌측, 세로 중앙에 정렬해요. 레이블이 있는 Header 높이와 레이블 높이를 같고 레이블이 시작 위치가 달라지지 않아요.
 - XL: L/M/S/XXS: 32px

Text Field

State

Text Field State

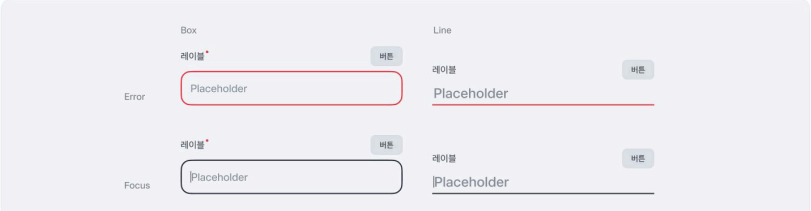
Default, Error, Disabled, Read only 4가지 상태가 있어요.



Error

유효하지 않은 값을 표창하는 상태예요.

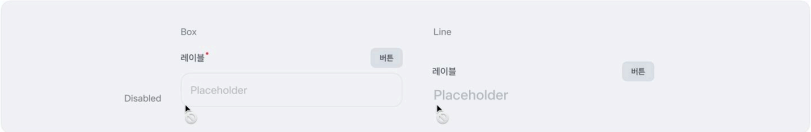
유효한 focus가 된다면 focus 표현을 더 우선시 해요.



Disabled

입력 자체를 차단하고, 값을 제출하는 경우 없이 바로로 전달되지 않아요.

내부 입력 요소가 아닌 등 인터랙티브 요소가 함께 제공될 수 있어요.



Read Only

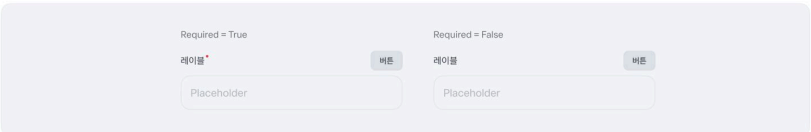
수정은 불가능하지만 텍스트를 입력할 수 있는 상태로 제공될 때 (readonly) 값이 변경되지 않고 제출되므로 전달 가능한 필수 있는 상태예요.

기본적으로 box를 통한 focus 적용에 포함되지 않아요. 사용자에서 box를 통한 focus 접근을 제한할 수 있어요.



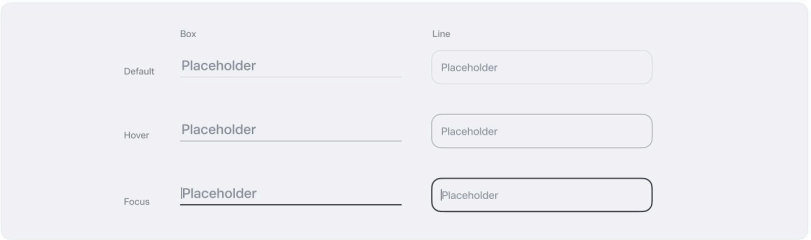
Required

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.



Input Interaction State / Default

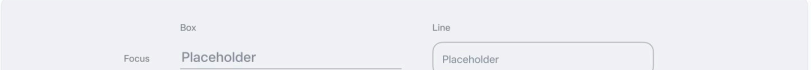
Default, Hover, Focus 3가지 상태가 있어요.



Hover

마우스 커서가 텍스트 위에 올랐을 때예요. 테두리 색과 조작성을 주는 영역임을 알리고요.

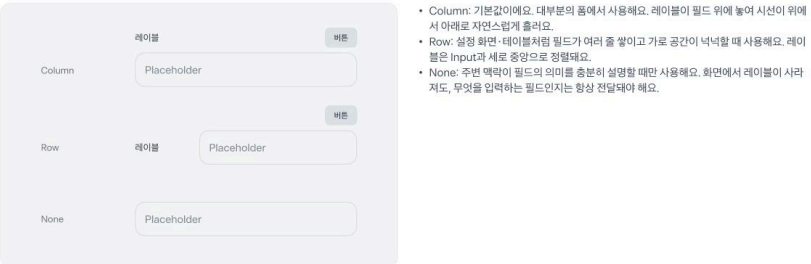
마우스가 있는 지점 한쪽에서는 나타나지 않아요.



Text Field

Usage

Direction



Row

Row 사용 규칙



Label+Button

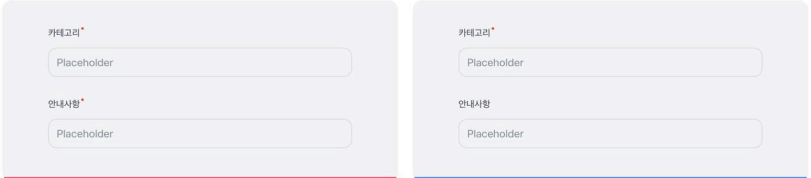
Label Slot



Top Slot



Required



Don't

모든 필드가 필수인 폼에 Required를 전부 붙이지 마세요. 구분할 필요가 없어요.

마우스가 있는 지점 한쪽에서는 나타나지 않아요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

필수 입력 항목임을 알리는 표시예요. 레이블 바로 뒤에 붙여요.

폼 전체에 필수 항목이 있을 때만 Required를 붙여요. 어떤 필드가 필수인지는 표시하지 않아요. — 폼을 볼려면 모두 필드도 주지 마세요.

엔터프라이즈 B2B 디자인 시스템

가비아 하이웁스, 디자인 시스템 구축

하이웁스 디자인 시스템의 Adobe XD와 Figma 파일이 혼재하며 컴포넌트 중복과 컬러 파편화 문제가 누적되고 있었습니다.

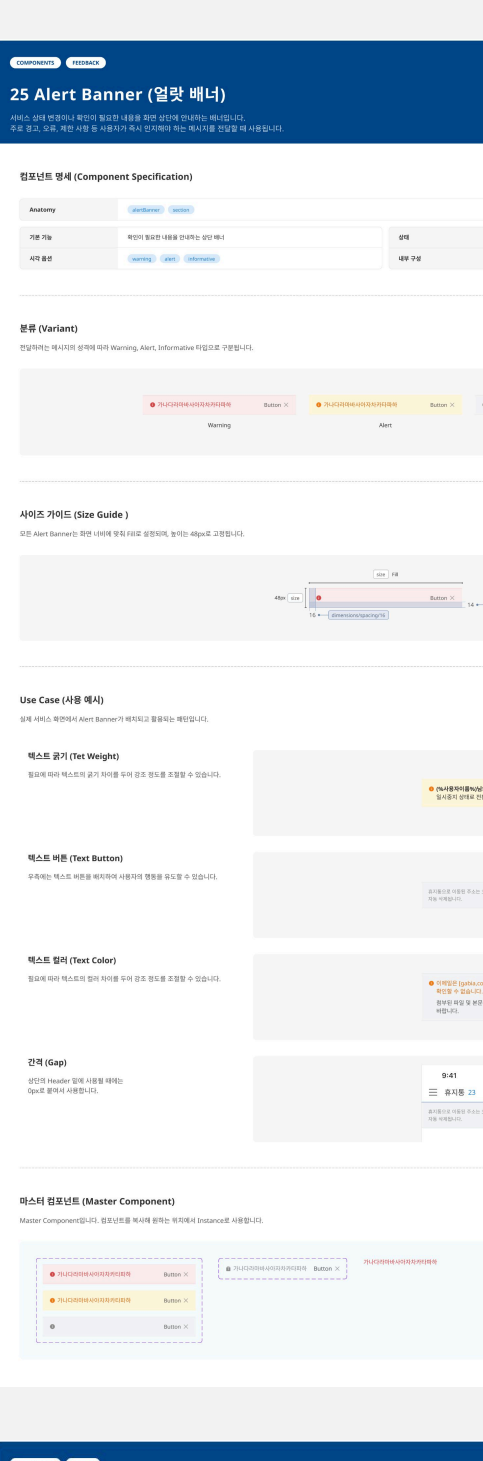
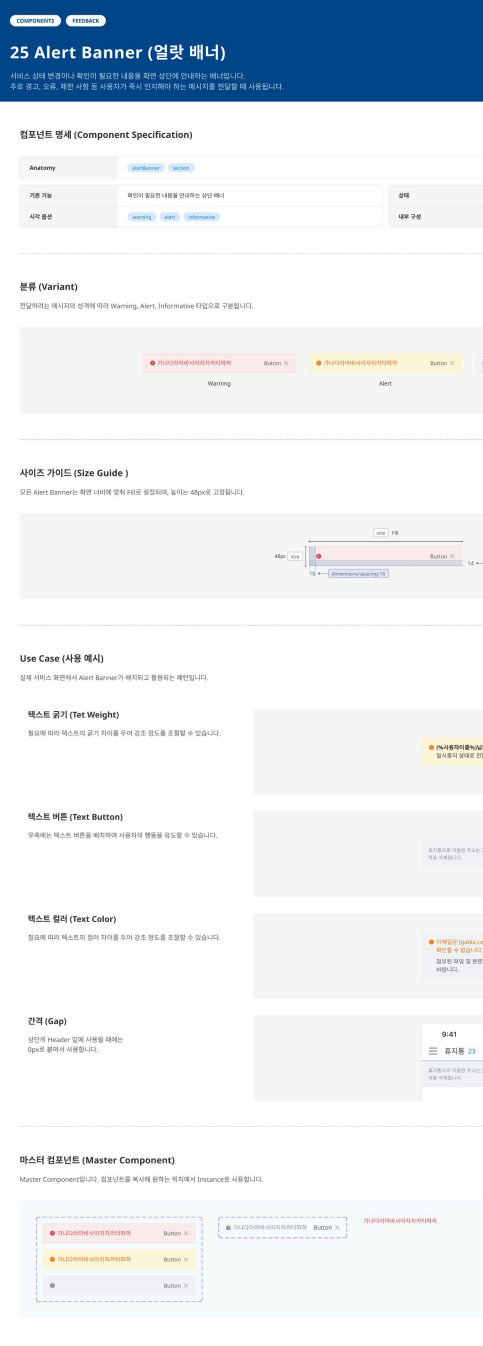
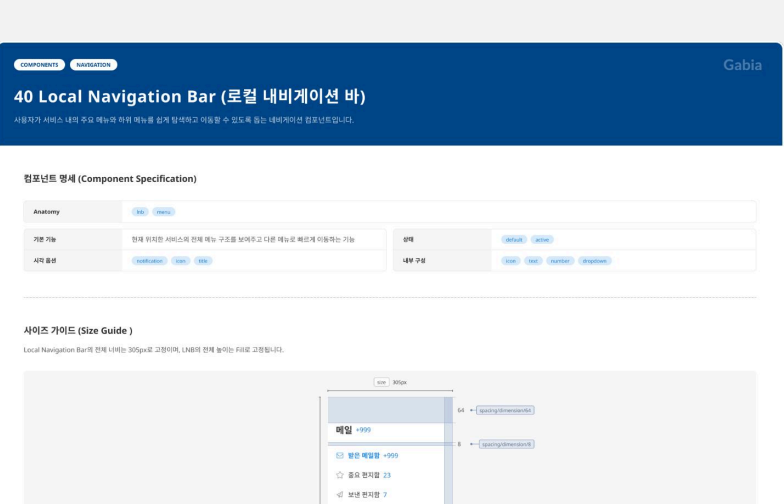
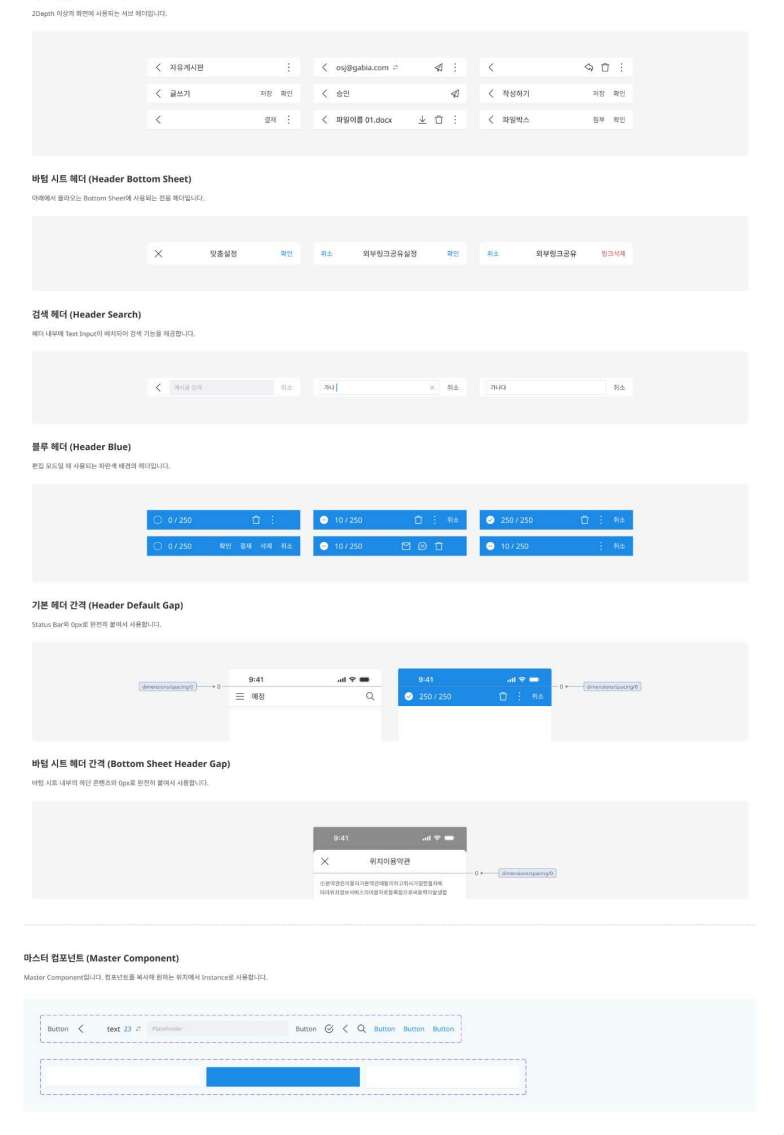
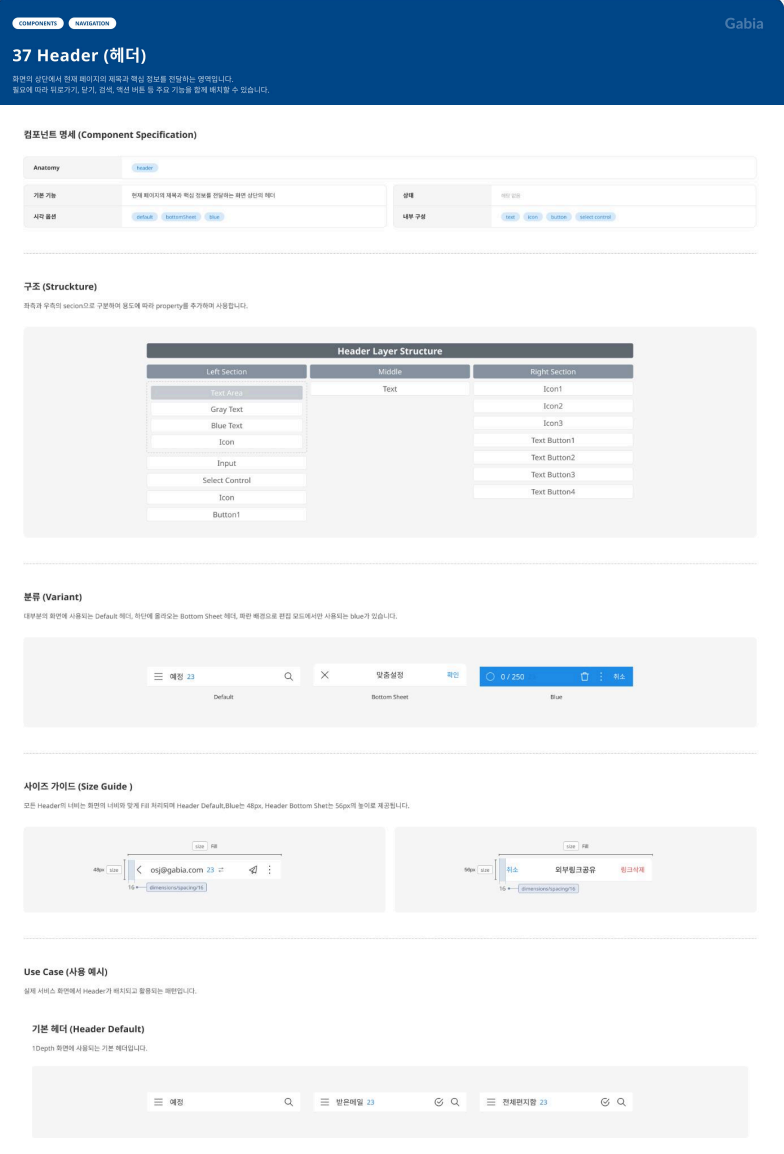
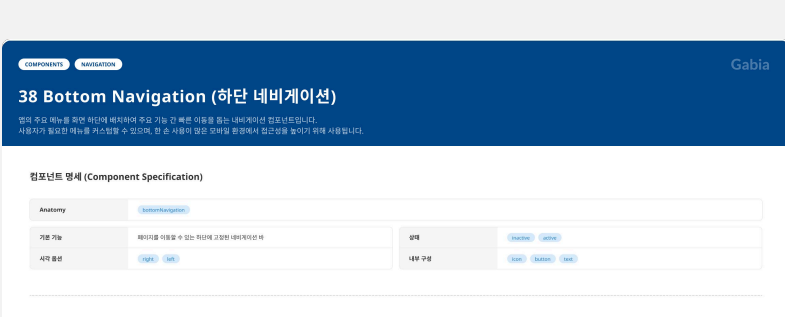
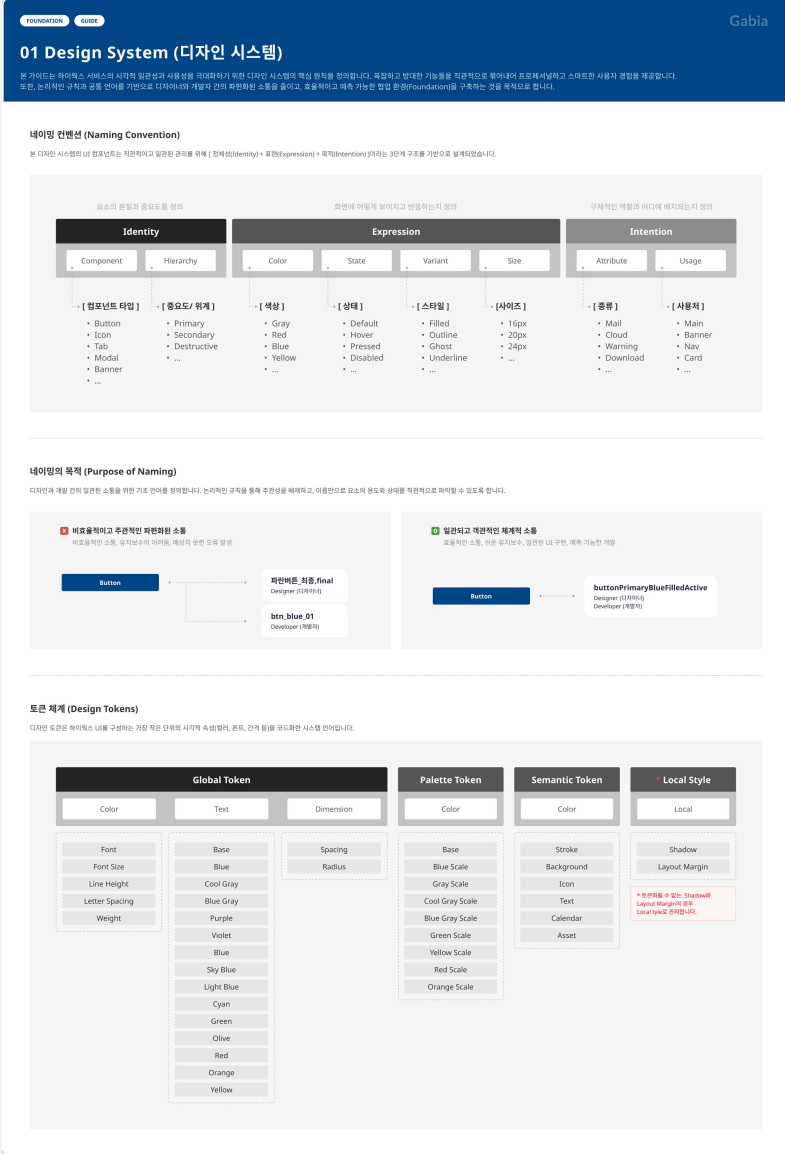
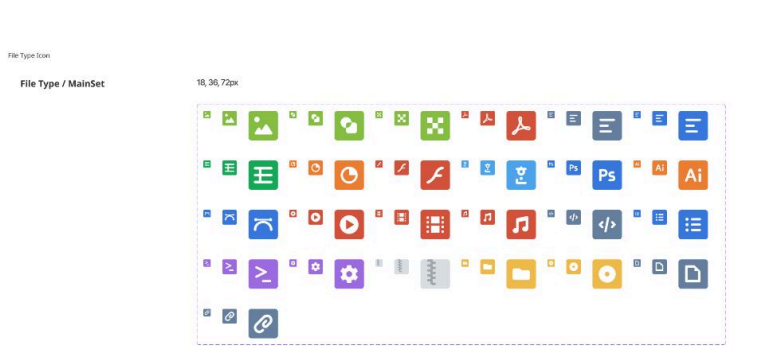
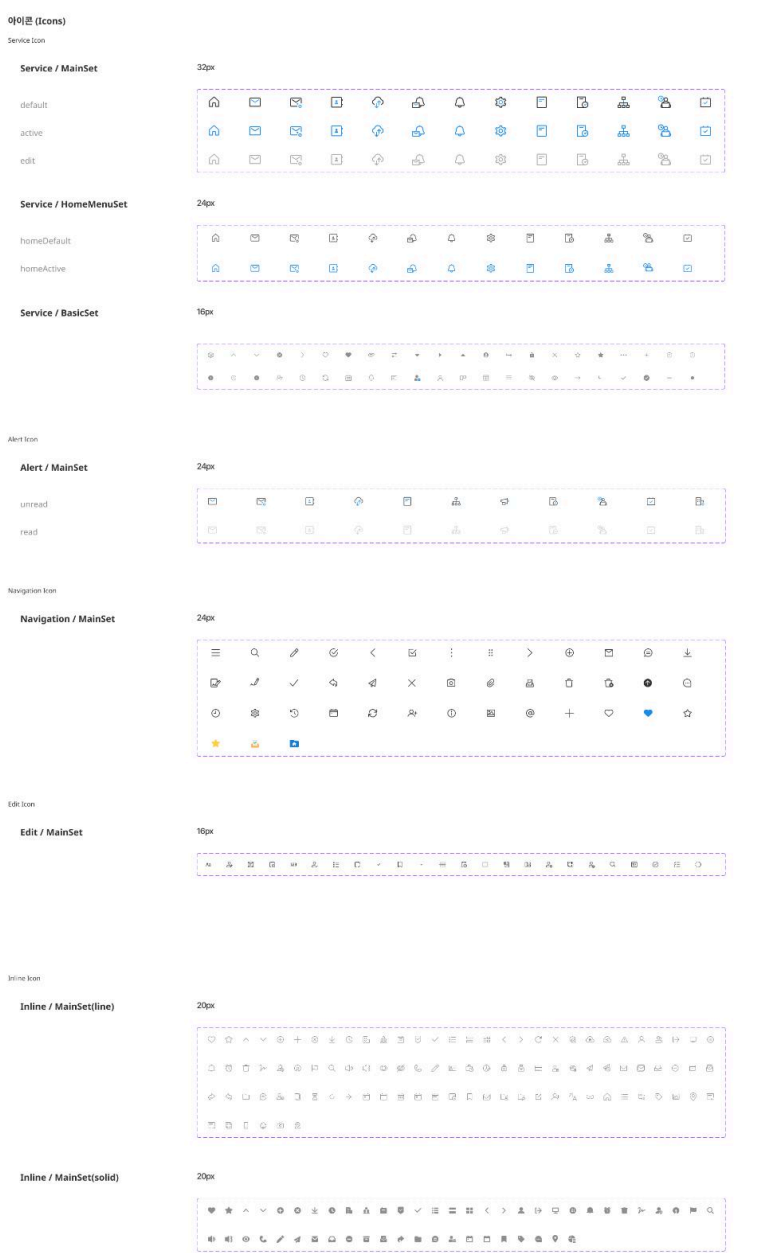
전사 서비스에서 공통으로 사용할 수 있는 **아토믹 기반의 컴포넌트 체계**와 **3단계 컬러 토큰 시스템**을 설계해, 유지보수가 가능한 디자인 시스템으로 개선했습니다.

디자인 시스템 링크 | <https://www.kangseohyun.com/pf/ds/ds>

기간 | 2026.01-2026.03

프로젝트 참여 인원 | 디자인 2명

디자인 시스템 구축 기여도 | 70%

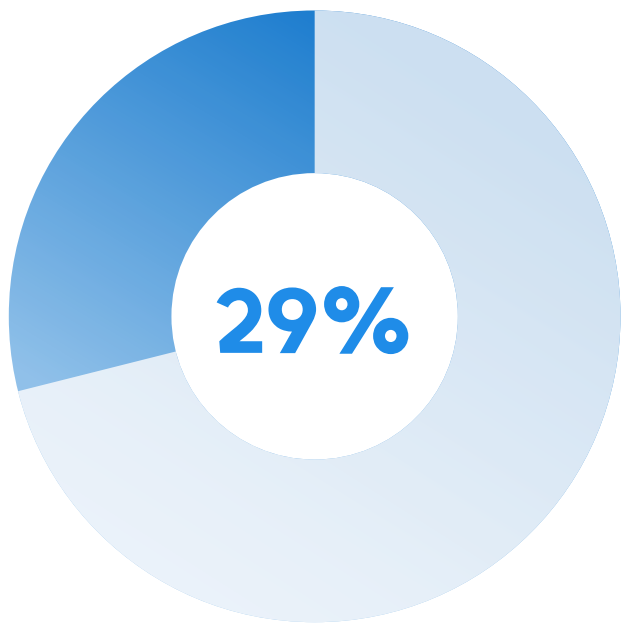


이러한 문제가 있었어요!

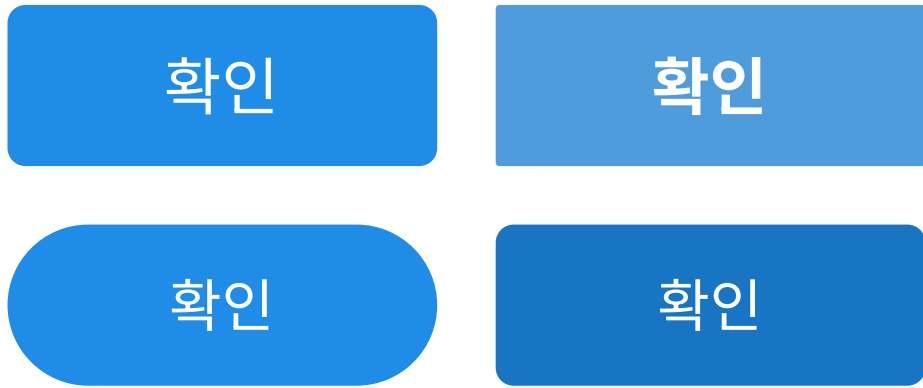
9년간 가비아의 서비스(Gabia Hiworks / Domain / Cloud)는 확장되었지만, 디자인 시스템은 관리되지 않고 있었습니다. 다음과 같은 문제가 발견되어 2가지 목표를 가지고 Token 기반의 Atomic 디자인 시스템을 세우고자 했습니다.

문제1

비슷한 컴포넌트가 하나로 정리되지 않고 여러 개로 나뉘어 관리되고 있었습니다.



[컴포넌트 중복율]
전체 72개 중 21개 중복



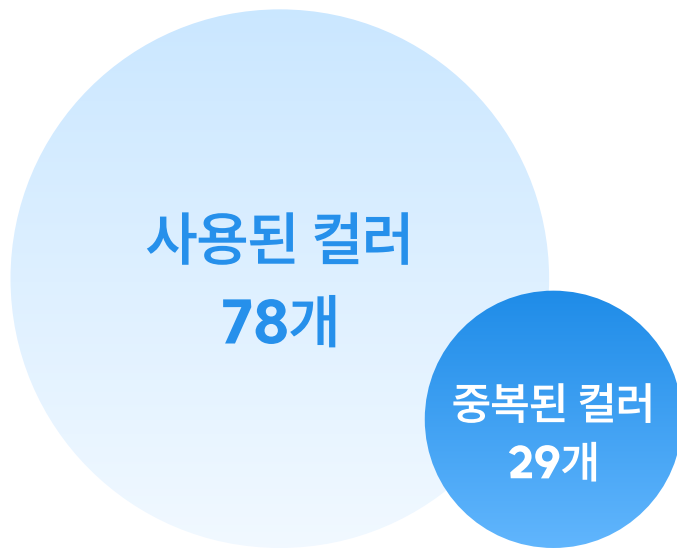
[중복되는 컴포넌트]
4개의 중복되는 CTA 버튼

목표1

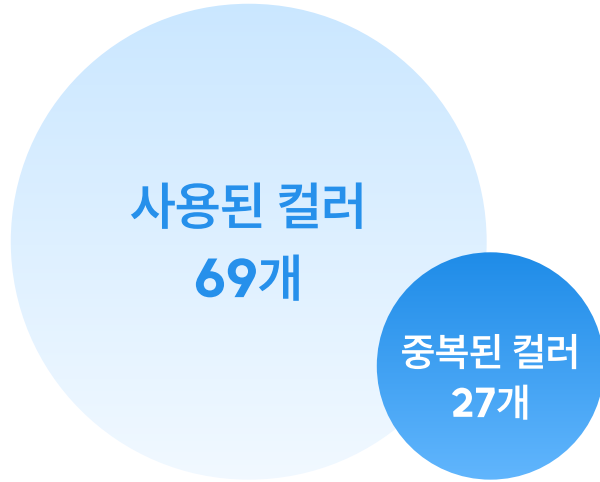
Atomic 디자인 시스템 기반의 효율적인 컴포넌트 체계를 구축한다.

문제2

가비아 전사의 각서비스 마다 컬러의 명도·채도가 달랐고, 유사한 색이 중복 사용되었습니다.



[Cloud, Domain 컬러]
전체 78개 중 29개 중복



[Hiworks 컬러]
전체 69개 중 27개 중복

목표2

컬러를 단순화하고, Token 기반의 전사 공통 컬러 시스템을 구축한다.

이렇게 토큰 체계를 세웠어요!

가비아 전사 토큰(Global Token)-팔레트 토큰(Palette Token)-역할별 토큰(Semantic Token)으로 토큰 체계를 세웠습니다.

Global Token			Palette Token	Semantic Token	Local Style	
Color	Text	Dimension	Color	Color	Text Style	Effect & Layout
Font	Base	Spacing	Base	Stroke	Headline	Shadow
Font Size	Blue		Blue Scale	Background		
Line Height	Cool Gray		Radius	Gray Scale	Icon	Body
Letter Spacing	Blue Gray	Cool Gray Scale		Text		
Weight	Purple	Blue Gray Scale		Calendar	Caption	
	Violet... +9	Green Scale... +3		Asset		

컴포넌트의 분류(Identity)-설명(Expression)-사용의도(Intention)에 따라서 네이밍 컨벤션을 세웠습니다.

Identity		Expression				Intention	
Component	Hierarchy	Color	State	Variant	Size	Attribute	Usage
[컴포넌트 타입] - Button - Icon - Tab - Modal ...	[중요도/ 위계] - Primary - Secondary - Teritiary - Destructive ...	[색상] - Gray - Red - Blue - Yellow ...	[상태] - Default - Hover - Pressed - Disabled ...	[스타일] - Filled - Outline - Ghost - Underline ...	[사이즈] 16px 20px 24px 28px ...	[종류] - Mail - Cloud - Warning - Download ...	[사용처] - Main - Banner - Nav - Card ...

이렇게 토큰 체계를 세웠어요!

4단계[Hex 코드-Global Token-Palette Token-SemanticToken] 컬러 토큰 체계를 세웠습니다.

1 HEX 코드 값

색상의 실제 값(HEX, RGB 등)으로,
시스템에 등록된 기본적인 색상 데이터

#1f8ce8
rgb(31, 140, 232)

Raw Value

#1f8ce8

13.46 AAA AA

1.56

2 가비아 전사 컬러 토큰

가비아의 전사 모든 서비스 (HiWorks, 도메인,
클라우드) 에서 공유되어 사용되는 컬러 토큰

#1f8ce8
rgb(31, 140, 232)

Gabia Global Token

v3 color/skyBlue/50

13.46 AAA AA

1.56

3 서비스별 컬러 팔레트 토큰

Gabia Global Token으로 매핑한
Hiworks에서만 사용되는 팔레트 컬러 토큰

#1f8ce8
rgb(31, 140, 232)

Hiworks Palette Token

blue/skyBlue/50

13.46 AAA AA

1.56

4 역할별 컬러 토큰

Hiworks Palette Toekn으로 매핑한
역할에 따라 분류된 컬러 토큰

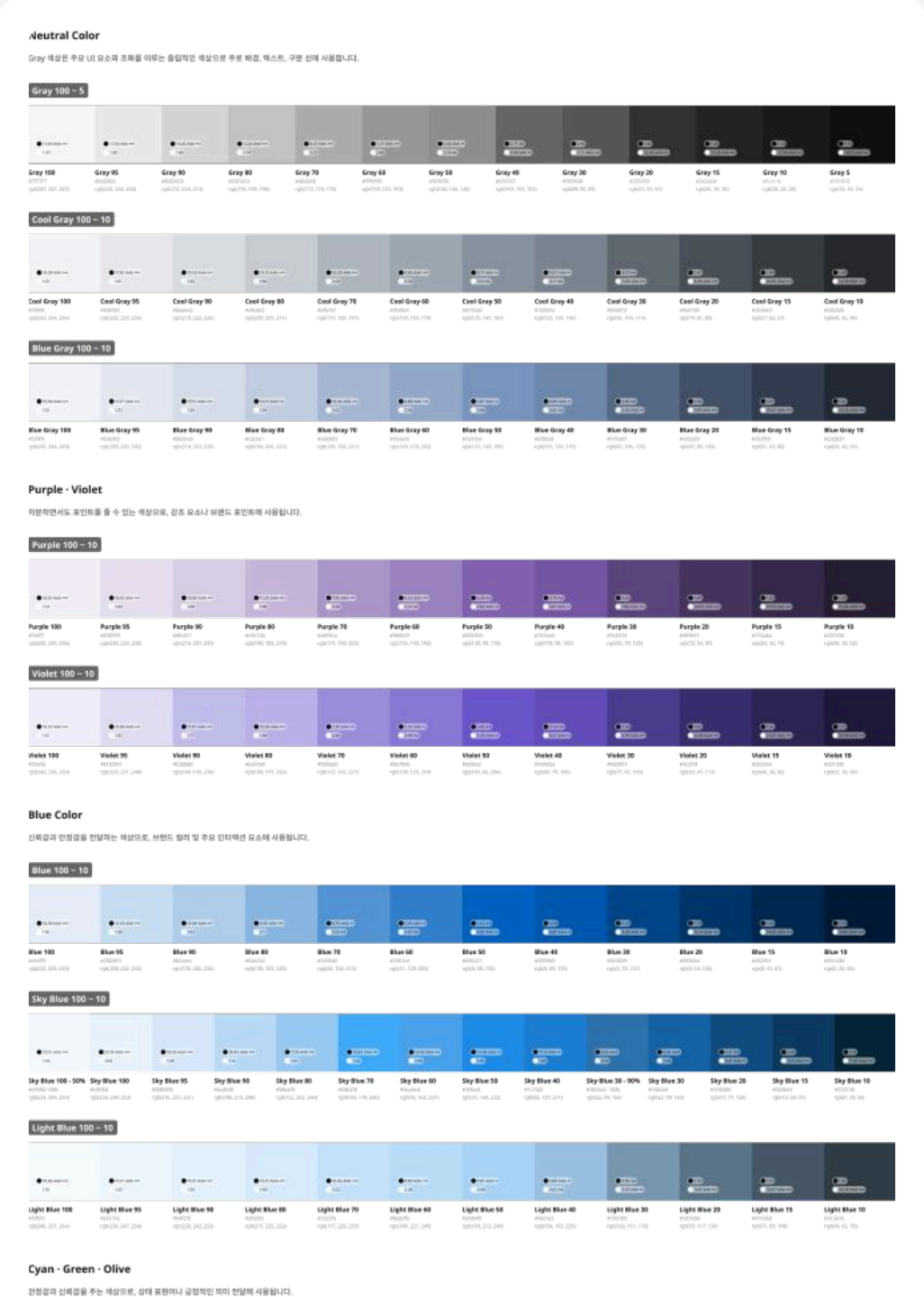
#1f8ce8
rgb(31, 140, 232)

Sementlc Token

bg/default/brand/Primary

13.46 AAA AA

1.56



Variables		🔍 Search			↗	✕
Collections	+	Name	Value	+		
global	223	v3 color / base				
palette	33	🎨 white	☐ FFFFFFFF			
semantic	136	🎨 black	☒ 000000			
Groups	☰	v3 color / gray				
All	223	🎨 100	☐ F7F7F7			
text	22	🎨 95	☐ EAEAEA			
font	1	🎨 90	☐ D6D6D6			
size	9	🎨 80	☐ C4C4C4			
weight	2	🎨 70	☐ AEAEAE			
lineHeight	8	🎨 60	☐ 999999			
lineSpacing	2	🎨 50	☐ 909090			
v3 color	173	🎨 40	☐ 676767			
base	2	🎨 30	☐ 585858			
gray	13	🎨 20	☐ 333333			
coolGray	12	🎨 15	☐ 242424			
blueGray	12					
purple	12					
violet	12					
blue	12					
skyBlue	14					

palette		Search			
Name	Light	Dark			
base					
white	Hiworks v3 color/base/white	Hiworks v3 color/base/black			
black	Hiworks v3 color/base/black	Hiworks v3 color/base/white			
blackOpacity	2B2B2B 33 %	272727 45 %			
blue					
lightBlue100	...orks v3 color/lightBlue/100	...orks v3 color/lightBlue/100			
lightBlue95	...orks v3 color/lightBlue/95	...orks v3 color/lightBlue/100			
lightBlue80	...orks v3 color/lightBlue/80	...orks v3 color/lightBlue/20			
lightBlue60	...orks v3 color/lightBlue/60	...orks v3 color/lightBlue/30			
skyBlue80	...works v3 color/skyBlue/80	...works v3 color/skyBlue/40			
skyBlue70	Hiworks v3 color/skyBlue/70	Hiworks v3 color/skyBlue/70			
skyBlue50	...works v3 color/skyBlue/50	...works v3 color/skyBlue/90			
skyBlue40	...works v3 color/skyBlue/40	...works v3 color/skyBlue/90			
blue30	Hiworks v3 color/blue/30	Hiworks v3 color/blue/100			
gray					
gray90	...orks v3 color/gray/90	Hiworks v3 color/gray/10			

semantic		Search			
Name	Light	Dark			
dim	base/blackOpacity	base/blackOpacity			
Icon / default / neutral					
strong	gray/gray20	coolGray/coolGray100			
normal	gray/gray30	coolGray/coolGray95			
subtle	gray/gray50	coolGray/coolGray100			
muted	gray/gray70	gray/gray70			
faint	base/white	base/black			
Icon / default / brand					
strong	blue/skyBlue40	blue/skyBlue80			
normal	blue/skyBlue50	blue/skyBlue70			
subtle	blue/skyBlue70	blue/skyBlue40			
Icon / default / positive					
normal	blue/skyBlue50	blue/skyBlue70			
Icon / default / warning					
normal	red/red50	FD6766			

이렇게 토큰 체계를 세웠어요!

일관성과 유지보수를 위해 데스크탑에 최적화된 텍스트, R값, 여백값을 토큰으로 등록했습니다.

T 텍스트 토큰 등록

서체 크기, 행간, 자간 값을 토큰 등록 후, Headline·Body·Caption으로 분류해 토큰을 속성값으로 참조했습니다.

Name	Font size	Line height	Weight
Headline/Display	22pt	32px	
Headline/Display	22pt	32px	
Headline/Title	20pt	28px	
Headline/Title	20pt	28px	
Headline/Default	18pt	26px	
Headline/Default	18pt	26px	0px Regular
Body/Default	16pt	24px	0px Bold
Body/Default	16pt	24px	0px Regular
Body/Emphasis	15pt	22px	0px Bold
Body/Emphasis	15pt	22px	0px Regular
Body/Supporting	14pt	20px	0px Bold
Body/Supporting	14pt	20px	0px Regular
Caption/Default	13pt	20px	0px Bold
Caption/Default	13pt	20px	0px Regular

Edit text style

Rag 123

Name

bold

Description

가장 높은 위계의 메시지나 화면의 핵심 제목을 표현할 때 사용하는 헤드라인 스타일입니다.

Properties

T text/font/Noto Sans

T text/weight/Bold

22

▼

A 32

|A| 0

...

Radius 값, Spacing 값 토큰 등록

컴포넌트의 Radius·Spacing 값을 토큰으로 등록해, 형태와 여백 기준을 통일했습니다.

Token	Rem	Size(px)	Example
Radius 0	0rem	0	
Radius 2	0.0625rem	2	
Radius 4	0.125rem	4	
Radius 6	0.25rem	6	
Radius 8	0.375rem	8	
Radius 10	0.5rem	10	
Radius 12	0.625rem	12	
Radius 14	0.75rem	14	
Radius 16	0.875rem	16	
max 100	1rem	100	

Token	Rem	Size(px)	Example
Spacing 0	0rem	0px	
Spacing 1	0.0625rem	1px	
Spacing 2	0.125rem	2px	
Spacing 4	0.25rem	4px	
Spacing 6	0.375rem	6px	
Spacing 8	0.5rem	8px	
Spacing 10	0.625rem	10px	
Spacing 12	0.75rem	12px	
Spacing 14	0.875rem	14px	
Spacing 16	1rem	16px	
Spacing 20	1.25rem	20px	
Spacing 24	1.5rem	24px	
Spacing 32	2rem	32px	
Spacing 36	2.25rem	36px	
Spacing 40	2.5rem	40px	
Spacing 64	4rem	64px	

각 상태별로 다양한 Variant의 컴포넌트를 등록했어요!

Input을 크기와 상태별로 분류해 Asset 등록을 했습니다.

Type	Empty	Default	Typing	Activated		
Solid / Small	Placeholder	Placeholder	Placeholder	Placeholder		
L / Medium /Button	Placeholder ⬆	Placeholder ⬆	Placeholder ⬆	Placeholder ⬆		
L / Medium /Icon	Placeholder ✕	Placeholder ✕	Placeholder ✕	Placeholder ✕	Placeholder ✕	Placeholder error text
L / Medium /Icon	Placeholder	Placeholder	Placeholder	Placeholder	Placeholder	Placeholder error text
L / Large	Placeholder	Placeholder	Placeholder	Placeholder	Placeholder	Placeholder error text
Line / Dropdown	text ▼	text ▼ Placeholder	text ▼ Placeholder	text ▼ Placeholder	text ▼ Placeholder	text ▼ Placeholder error text
L / Title / Label	title 0 Placeholder	title 0 Placeholder	title 0 Placeholder	title 0 Placeholder	title 0 Placeholder	title 0 Placeholder
L / Title / Bold	title Placeholder	title Placeholder	title Placeholder	title Placeholder	title Placeholder	title Placeholder
L / Title / Reg	title Placeholder	title Placeholder	title Placeholder	title Placeholder	title Placeholder	title Placeholder
L / Title / Bold / Icon	title Placeholder ▼	title Placeholder ▼	title Placeholder ▼	title Placeholder ▼	title Placeholder ▼	title Placeholder ▼
L / Title / Reg /Select	title Placeholder ○ Text	title Placeholder ○ Text	title Placeholder ○ Text	title Placeholder ○ Text	title Placeholder ○ Text	title Placeholder ○ Text

Input label ▾

From this file ◇

labelCount4 ▾

stateerror ▾

leftSide

leftSide/subtitle

leftSide/dropd...

rightSide

rightSide/icon

errorText

icon/inline/20px/MainSet(line) ▾

typechevron... ▾

Input solid ▾

From this file ◇

heightlarge ▾

stateempty ▾

errorText

button

caret

icon

scroll

text

Placeholder

icon/service/16px/BasicSet

typebigClose ▾

Input large ▾

From this file ◇

weightbold ▾

statedefault ▾

leftSide

leftSide/icon

leftSide/dropd...

leftSide/label

rightSide

rightSide/icon

title

title/asta

errorText

rightSide/select

rightSide/unit

각 상태별로 다양한 Variant의 컴포넌트를 등록했어요!

Button과 Label의 크기, 상태별로 구분해서 Asset 등록을 했습니다.

Button			
Type	Default	Press	Disabled
Blue / Large	Button	Button	Button
Blue / Medium	Button	Button	Button
Blue / Small	Button	Button	Button
Gray / Large	Button	Button	Button
Gray / Medium	Button	Button	Button
Gray / Small	Button	Button	Button
Line / Large	Button	Button	Button
Line / Medium	Button	Button	Button
Line / Small	Button	Button	Button
Text / Gray / Large	Button	Button	Button
Text / Gray / Medium	Button	Button	Button
Text / Gray / Small	Button	Button	Button
Text / Blue / Large	Button	Button	Button
Text / Blue / Medium	Button	Button	Button
Text / Blue / Small	Button	Button	Button

Variants

Button

Button

Button

Brand Solid

Neutral Solid

Outline

Button

Button

Button

Neutral Solid

Neutral Ghost

Outline

32px

32px

size

14

6

spacing/dimension/14

spacing/dimension/6

36px

36px

size

24

8

spacing/dimension/24

spacing/dimension/8

42px

42px

size

24

10

spacing/dimension/32

spacing/dimension/12

Label				
BG, Text Color	Gray	Balck	Red	Blue
White	text	text	text	text
Gray	text	text	text	text
No BG	text	text	text	text
Blue	text	text	text	text
Gray / Round	text	text	text	text
Gray / Round	⊗ text	⊗ text	⊗ text	⊗ text
BG, Size	M/Round	M/Edge	S/Round	S/Edge
Green	text	text	text	text
Red	text	text	text	text
Yellow	text	text	text	text
Blue	text	text	text	text
Gray	text	text	text	text
Line / Gray	text	text	text	text
Color	Gray	Blue	Navy	Navy 2
Number	0	0	0	00

Label small

From this file

radiuspill

bgColoryellow

text

text2text

Label medium

From this file

bgColorblue

stroke

textColorblue

radiusrounded

text

text2text

Button

From this file

SizeLarge

StyleBrand S...

Disabled

Icon Left

Icon Right

Swap Icon 24pxMain...

TextButton

icon/navigation/24px/MainSet

typeinfo

icon/navigation/24px/MainSet

typeinfo

26px

spacing/dimension/8

8

26px

size

3

3px

22px

spacing/dimension/8

8

22px

size

2

spacing/dimension/2

16px

5px

5

16px

size

0

0

0px

각 Section별로 구분해서 컴포넌트를 등록했어요!

List처럼 복잡한 컴포넌트는 각 섹션별로 나누어, 아토믹 기반으로 Asset 등록을 했습니다.

아토믹 기반의 에셋 등록

Head·Body·Caption 각각의 섹션으로 나누어 에셋을 등록했습니다.
필요한 속성만 조합해 다양한 게시판 유형에 유연하게 사용합니다.

Head

Body

Caption

자유게시판

첨부파일 받아주세요.
요청하신 자료를 첨부파일로 전달드립니다. 파일 확인 후 이
상이 있거나 추가로 필요한 자료가 있으면 말씀해 주세요....

자료1.pdf (000KB)

김길동 20:00

조회 120

00

- Head

Body

Caption
- 타이틀과 아이콘으로 게시판 유형을 표시합니다.
[내용] title / icon

제목·본문·투표·첨부파일 속성을 조합해 게시글 내용을
구성합니다.
[속성값] title / mainBody / vote / file

좌측에 프로필·작성시간, 우측에 조회수·좋아요를
배치해 부가 정보를 표시합니다.
[속성값] left(profile / time) / right(time / view / like)

List board

From this file

stateimportant

caption

body

head

head

typehead

body

typebody

title

mainBody

vote

file

caption

typecaption

left

right

profile

time

view

like

Default

Important

자유게시판

첨부파일 받아주세요.
요청하신 자료를 첨부파일로 전달드립니다. 파일 확인 후 이
상이 있거나 추가로 필요한 자료가 있으면 말씀해 주세요....

자료1.pdf (000KB)

김길동 20:00

조회 120

00

자유게시판

첨부파일 받아주세요.
요청하신 자료를 첨부파일로 전달드립니다. 파일 확인 후 이
상이 있거나 추가로 필요한 자료가 있으면 말씀해 주세요....

자료1.pdf (000KB)

김길동 20:00

조회 120

00

spacing/dimension/16

16

size

Fill

spacing/dimension/16

16

spacing/dimension/10

10

spacing/dimension/8

8

spacing/dimension/8

8

spacing/dimension/10

10

Hug

size

spacing/dimension/8

8

8

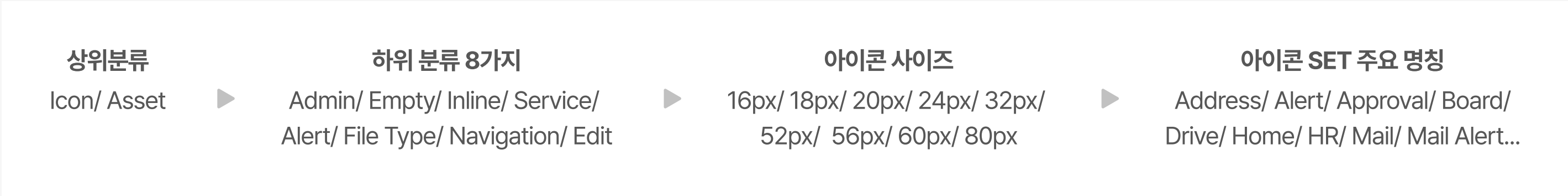
8

아이콘과 에셋도 분류해서 에셋화했어요!

체계적인 분류 체계로 효율적으로 아이콘과 에셋을 찾아서 사용할 수 있습니다.

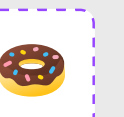
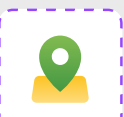
체계적인 분류 체계

다음과 같은 네이밍컨벤션과 폴더트리를 가지고 아이콘과 에셋을 분류했습니다.
사용자는 네이밍만 보고 좌측 Aeet패널에서 필요한 아이콘과 에셋을 찾아 사용할 수 있습니다.



Asset/ Admin

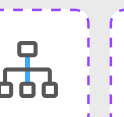
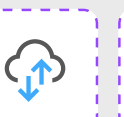
e.g.



다중 컬러와 그라데이션이 적용되어 업무 기능 및 상태를 강조하는 에셋

Icon/ Service

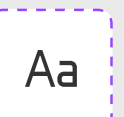
e.g.



하이웍스의 주요 서비스 진입 및 핵심 기능 활성화를 위한 대표 아이콘

Icon/ Edit

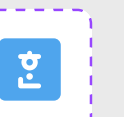
e.g.



문서 편집기 및 다양한 입력 폼 제어에 사용되는 기능성 아이콘

Icon/ File Type

e.g.



첨부파일의 포맷과 확장자 속성을 식별할 수 있도록 돕는 아이콘

Asset/ Empty

e.g.



Empty State를 직관적이고 친근하게 안내하는 일러스트 에셋

Icon/ Inline

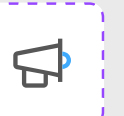
e.g.



본문 내 텍스트 혹은 좁은 공간에서 사용되는 아이콘

Icon/ Alert

e.g.



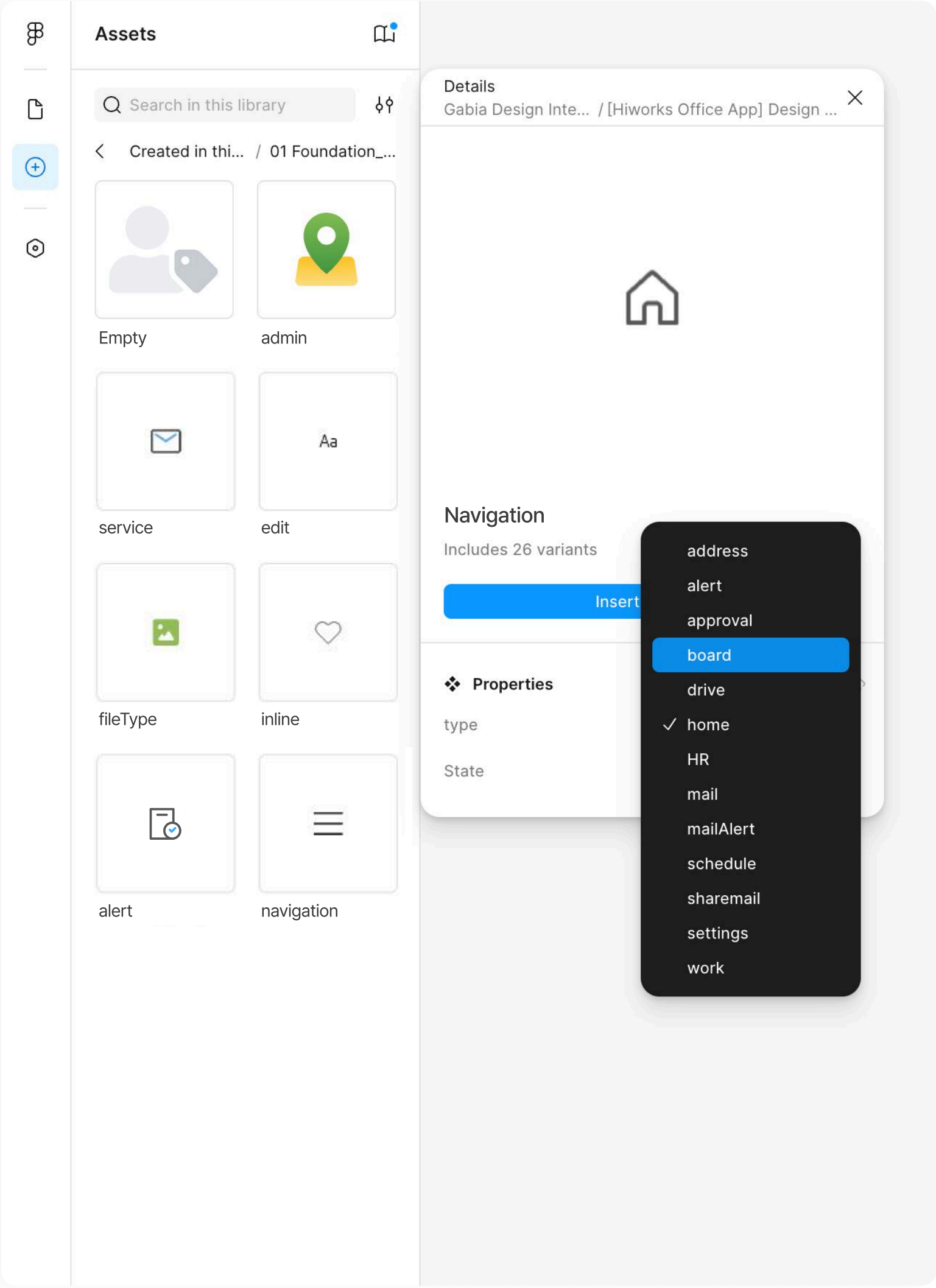
상태 변화 및 알림 피드백을 나타내는 아이콘

Icon/ Navigation

e.g.

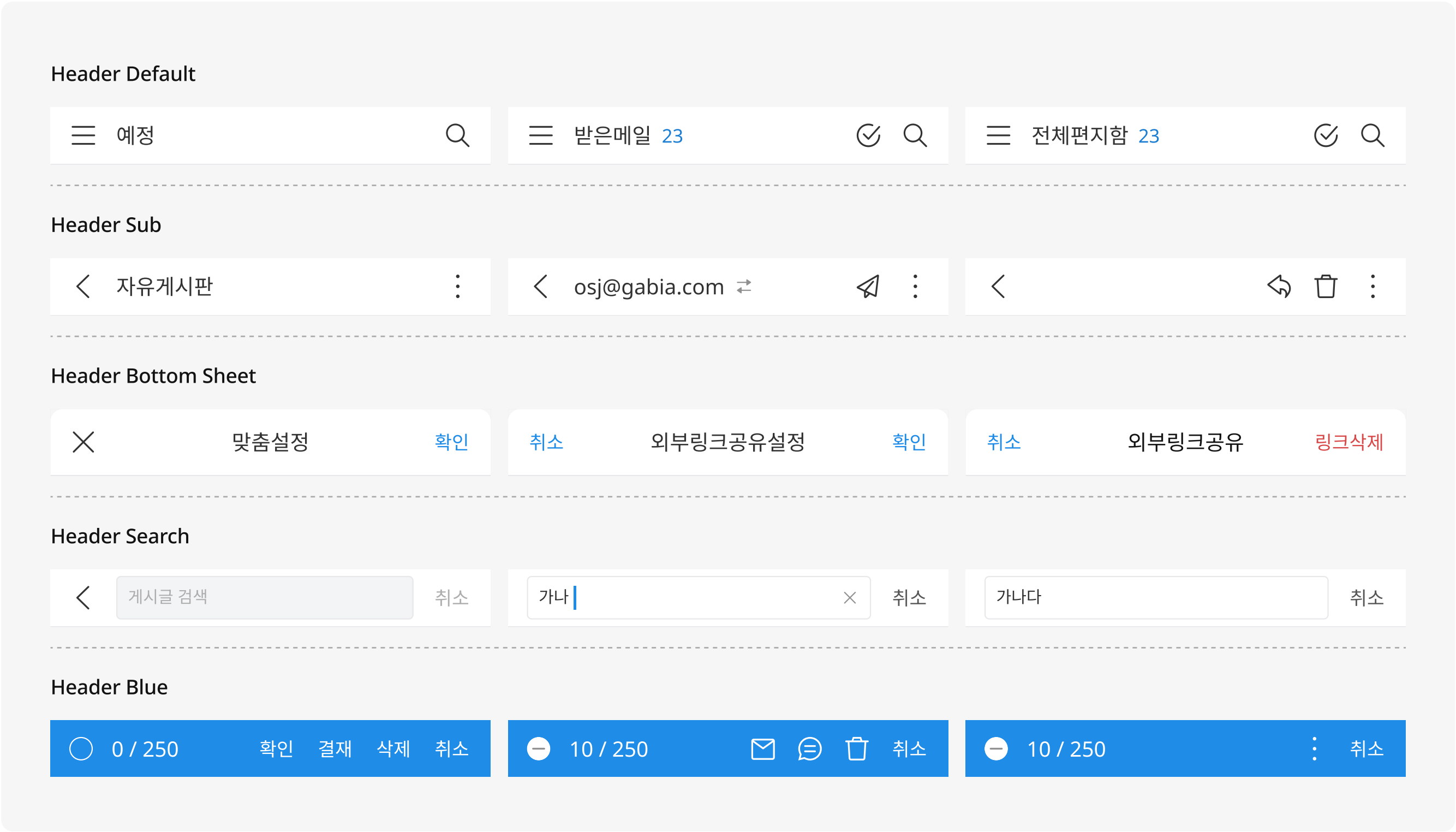
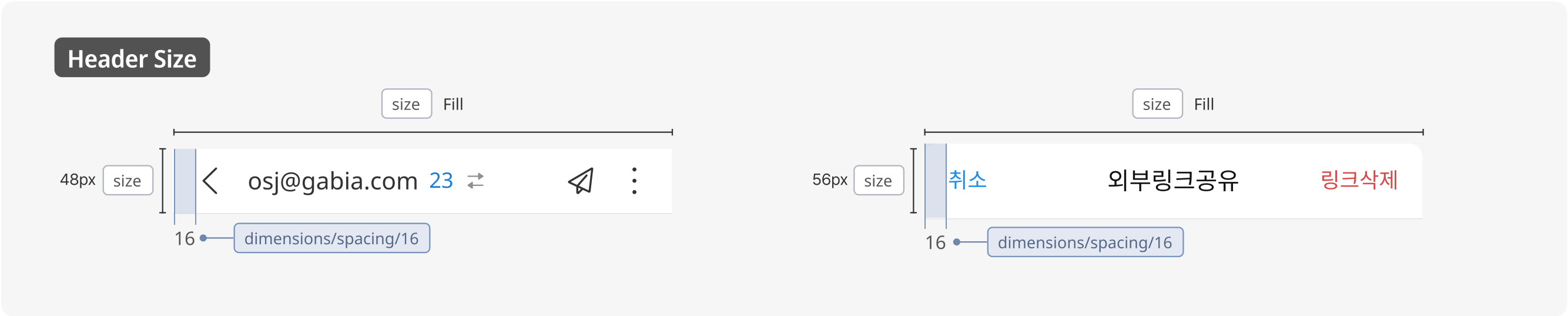


화면 탐색 및 공통 UI 제어를 위한 방향성 및 액션 아이콘



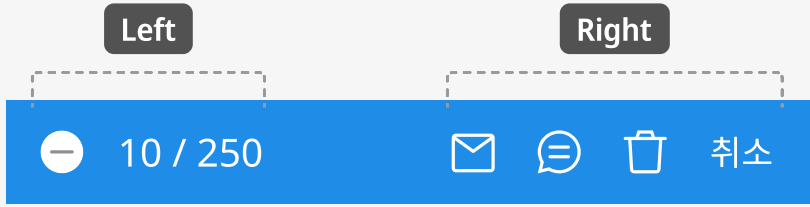
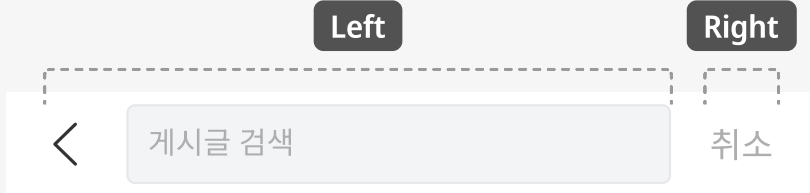
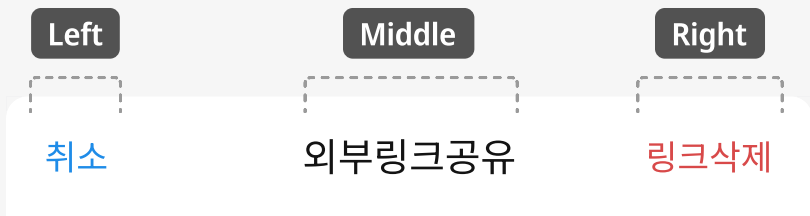
각 Section별로 구분해서 컴포넌트를 등록했어요!

Header처럼 복잡한 컴포넌트는 각 섹션별로 나누어, 아토믹 기반으로 Asset 등록을 했습니다.



아토믹 기반의 에셋 등록

Left·Middle·Right으로 레이어를 나누어 에셋을 등록했습니다.
필요한 속성을 조합해 헤더에 유연하게 사용합니다.



Left Side

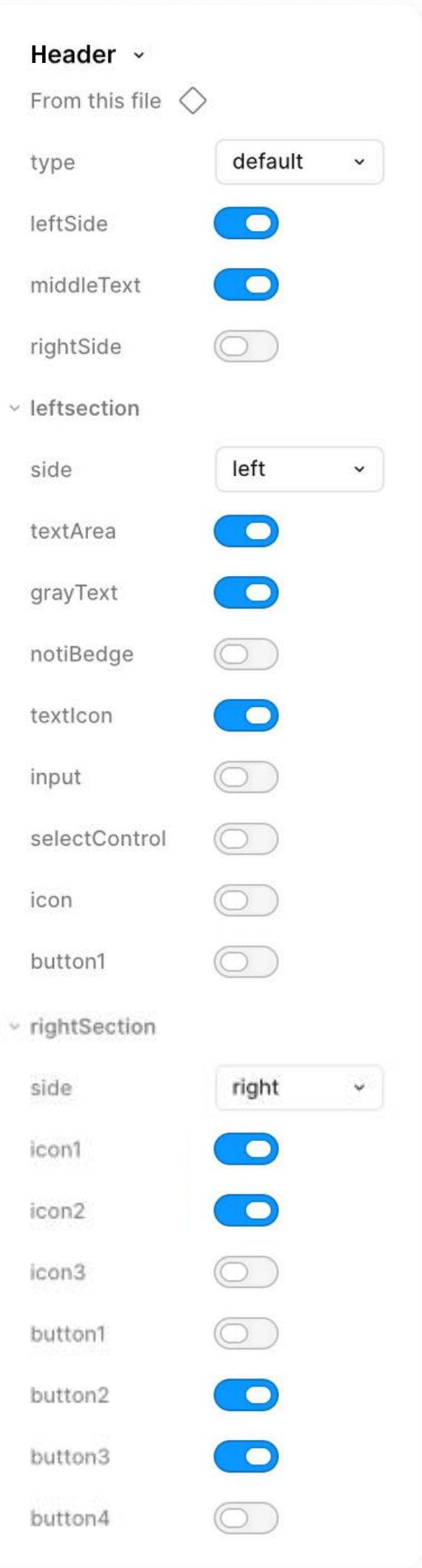
아이콘, 텍스트, 상태 요소 등을 배치해 화면의 시작 정보를 전달하는 영역입니다.

Middle Side

화면의 제목이나 핵심 정보를 중앙에 표시하는 영역입니다.

Right Side

추가 액션(아이콘, 버튼 등)을 배치해 사용자의 빠른 기능 실행을 돕는 영역입니다.

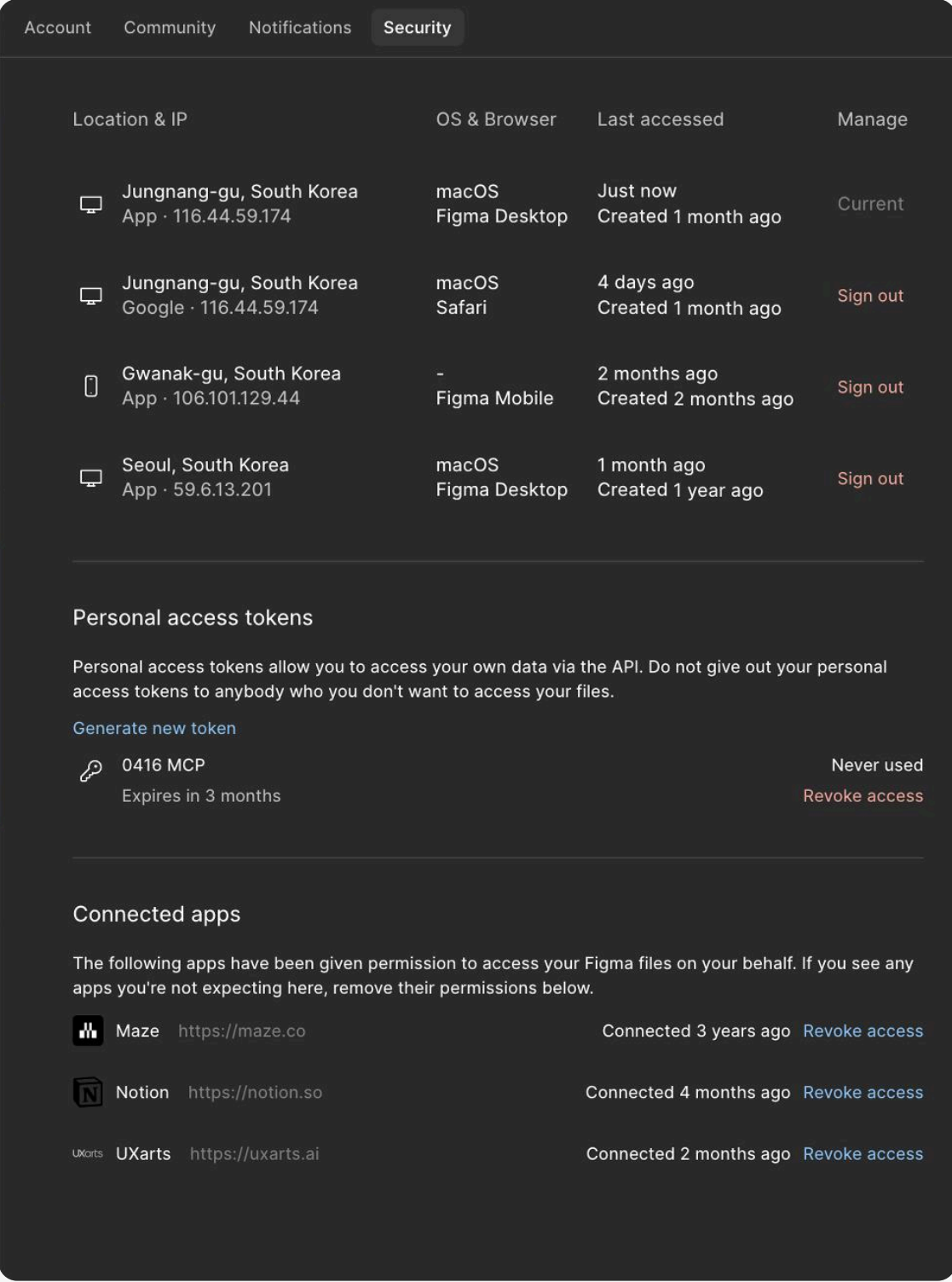


Claude AI와 연결해 코드까지 자동화했어요!

이후 AI-Native Design Workflow까지 구축했습니다.

Figma MCP와 Claude Code를 연결해 디자인 시스템을 코드를 자동 추출할 수 있습니다.

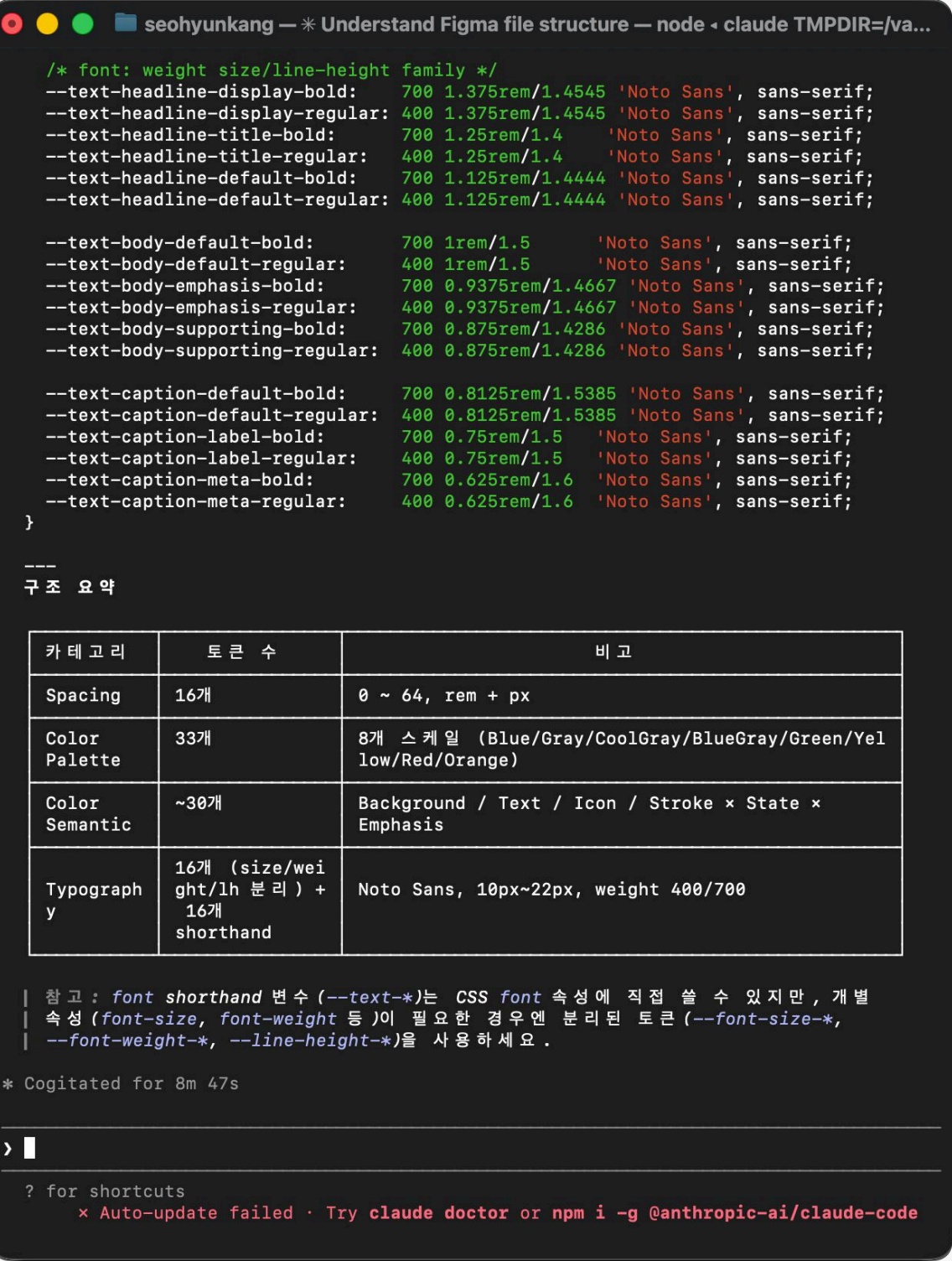
1 Figma API 발급



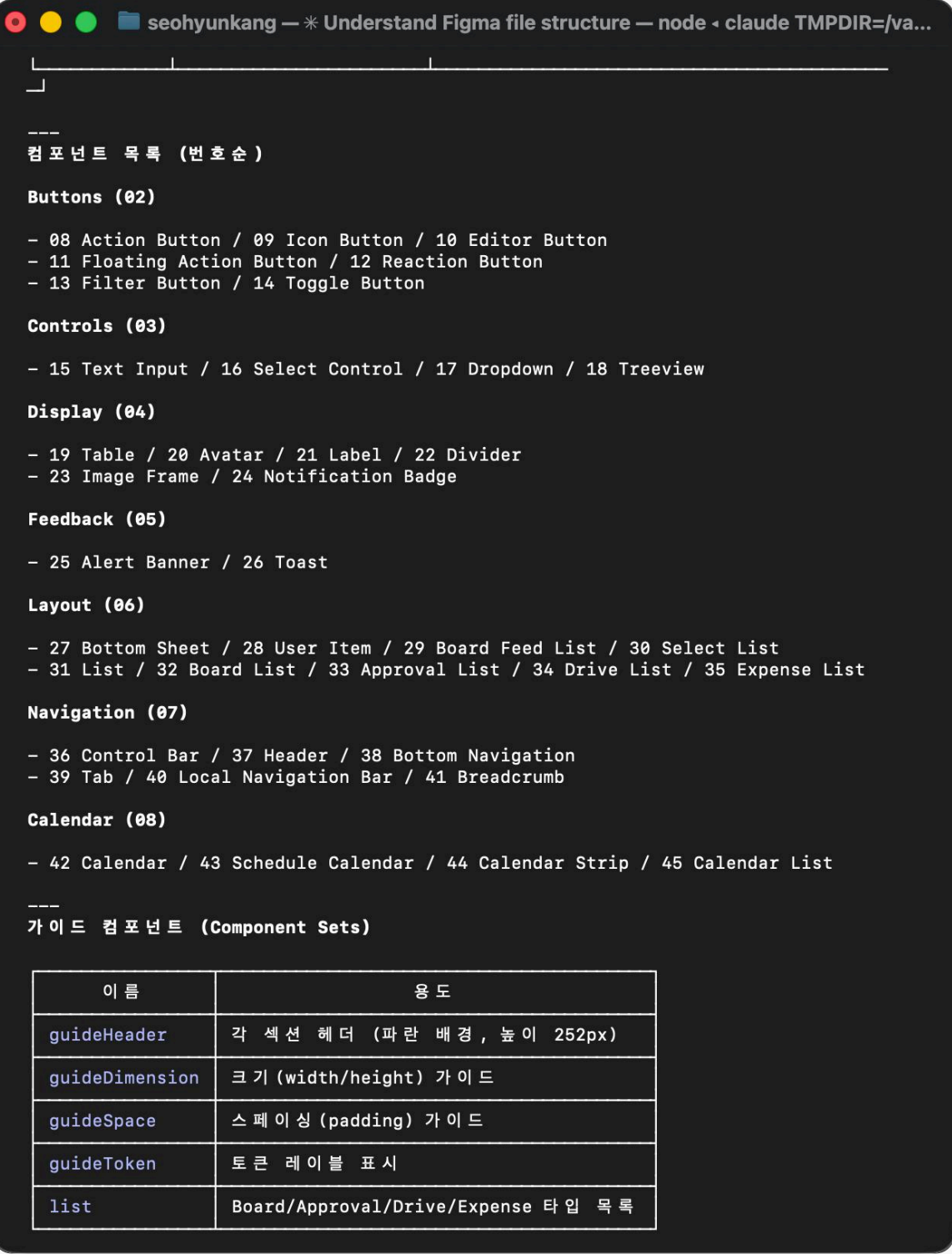
2 Claude Code MCP 연결



3 Claude Code: Design system 토큰 추출



4 Claude Code: Component 코드 생성



개선 후 이런 효과가 있었어요!

중복된 컴포넌트와 컬러를 통합해 운영 효율을 높였으며, 현재까지 9명의 디자이너가 7개월간 실무에 적용해 사용하고 있습니다.

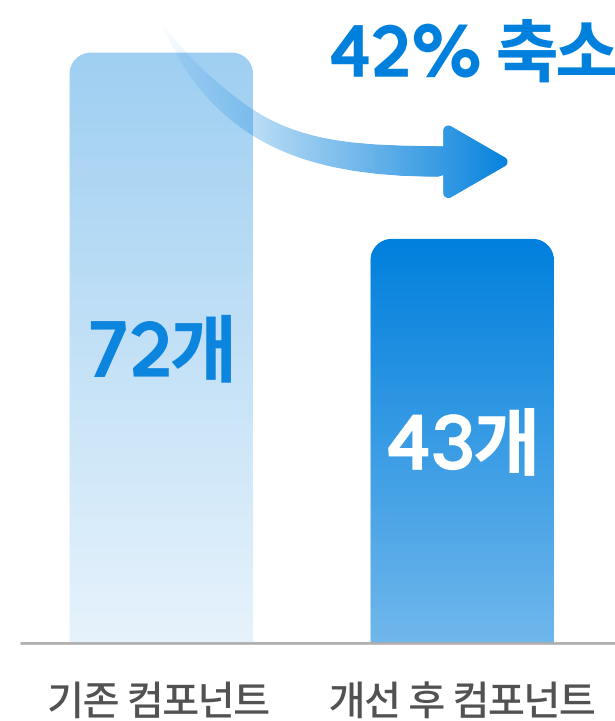
❗ 디자인 시스템은 아래 링크에서 더 자세하게 확인하실 수 있습니다.

디자인시스템 웹페이지 바로가기 | <https://www.kangseohyun.com/pf/ds/ds>

디자인시스템 Figma 바로가기 | <https://buly.kr/H6jsGqp>

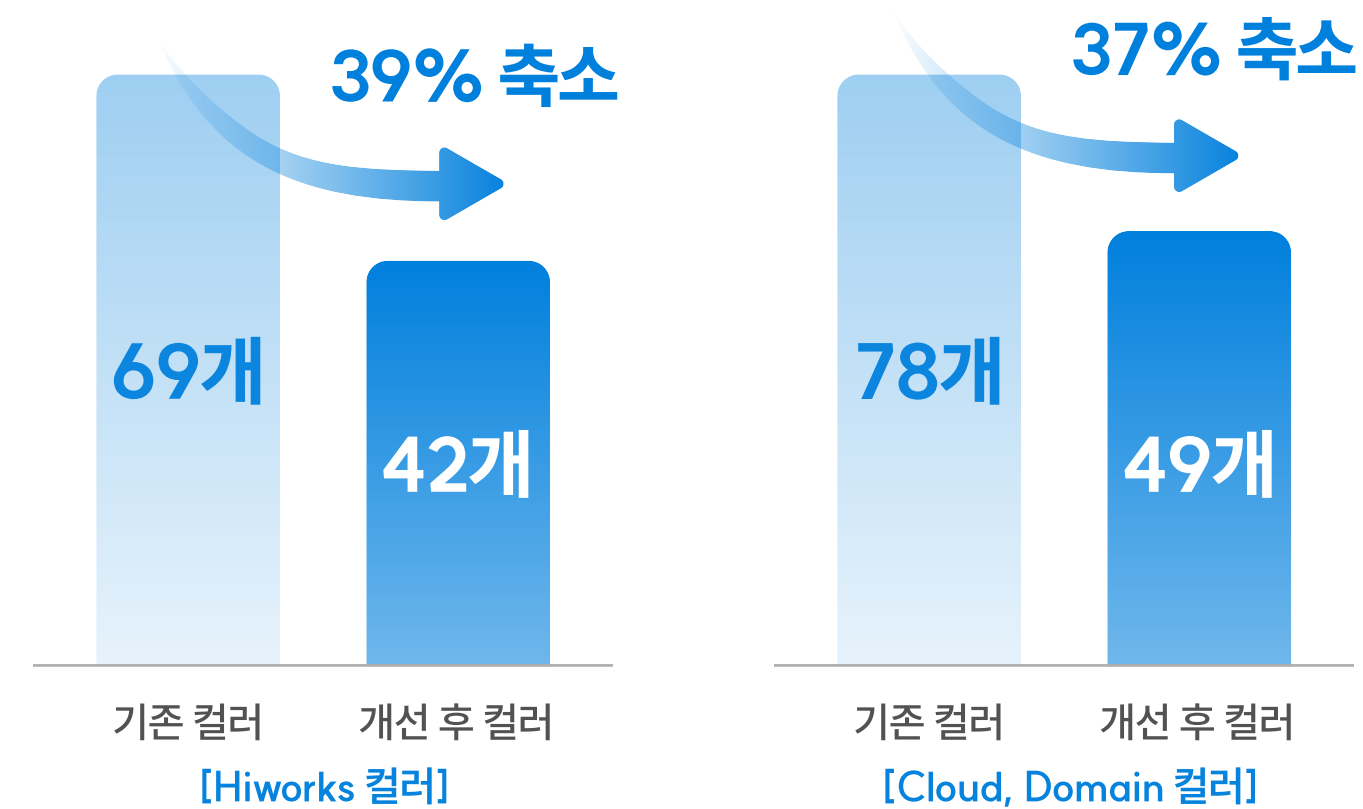
효과1

컴포넌트는 **42% 통합·축소**하여 운영 효율과 유지보수 편의성을 높였습니다.



효과2

HiWorks 컬러는 **39%**, Cloud/Domain은 **37%** 축소해 일관성을 높였습니다.



효과3

7개월간 9명의 디자이너가 실제 업무에 적용하며, 정식 디자인 시스템으로 자리 잡았습니다.



UX 개선

UX 리서치

SOPT: IT 동아리 앱, 모임 개설 UX 개선

SOPT는 대학생 연합 IT 벤처 창업 동아리로, 자체 제작한 앱을 통해 부원들이 스터디를 만들고 함께 공부할 수 있는 기능을 제공합니다.

약 3,000명 정도의 부원들이 사용하는 자체 앱의 모임 개설 UX의 문제점을 분석하고 개선했습니다. 항목 재구성과 UI 개선을 통해 작성 시간을 37% 단축하고, 완료율을 20% 향상시켰습니다.



아티클 링크 | <https://sopt-makers.tistory.com/2>

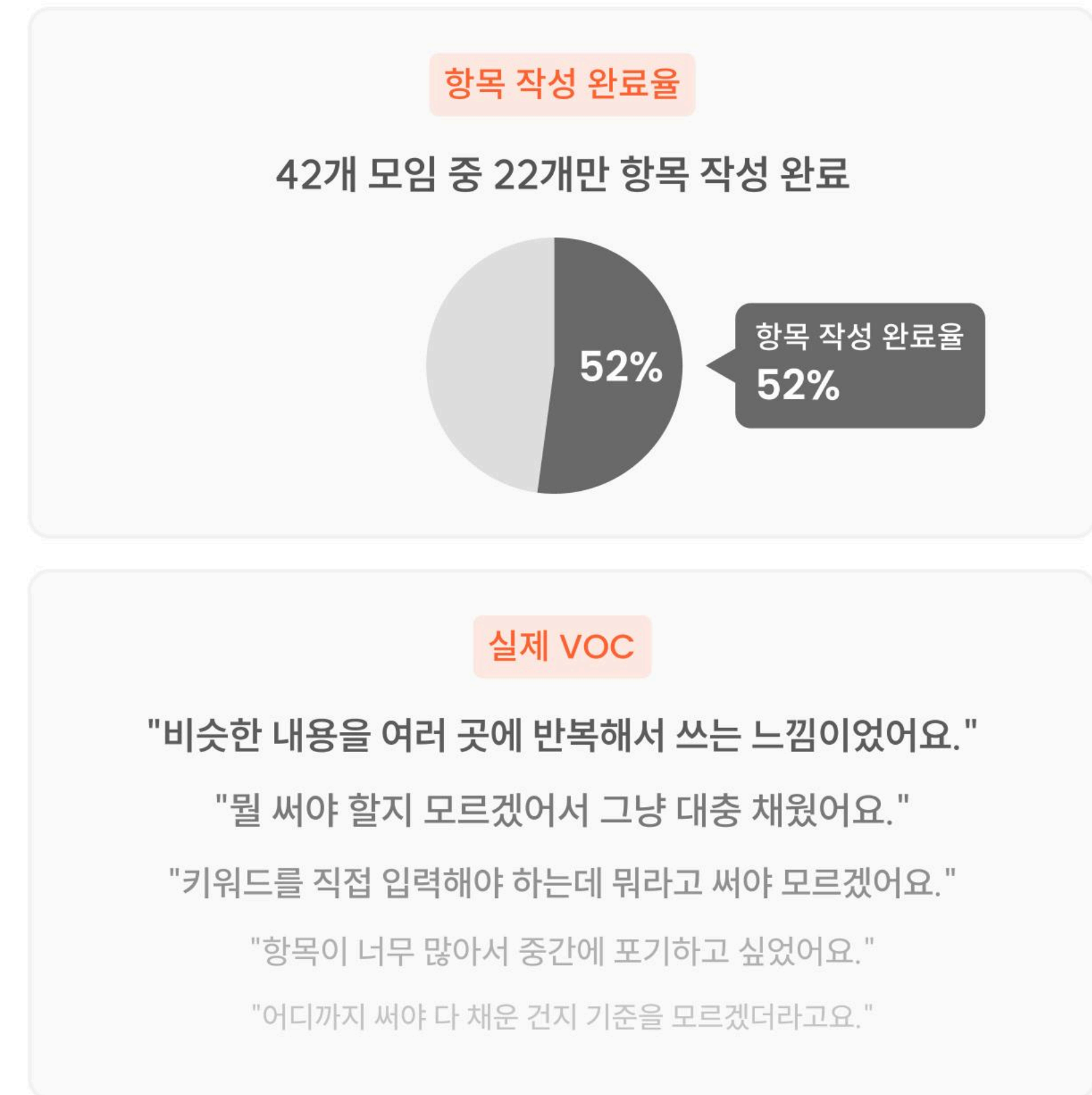
기간 | 2024.10-2025.01

리서치, 디자인 기여도 | 60%

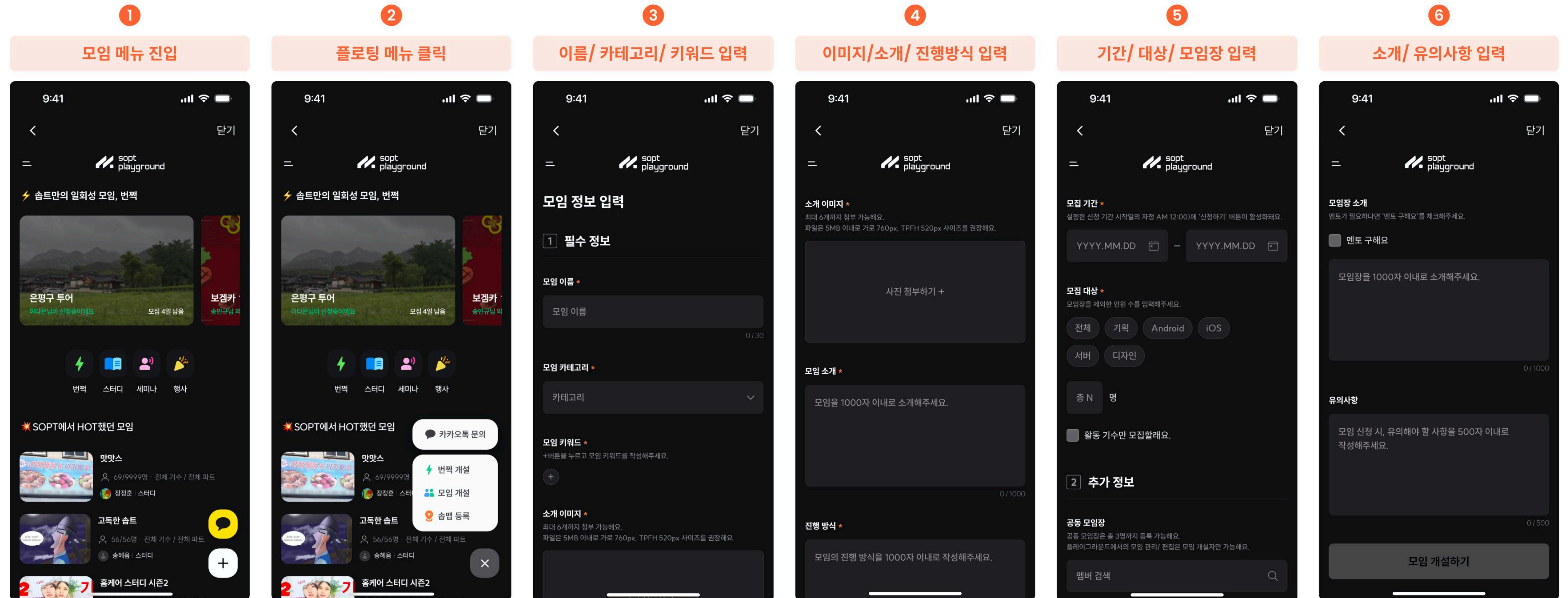
SOPT는 IT 동아리로, 자체 제작한 앱을 통해 부원들이 스터디를 만들고 함께 공부할 수 있는 기능을 제공합니다.

 개발자 알고리즘 문제풀이 스터디	 디자이너 Nano Banana 스터디
 기획자 앱 서비스 뜯어보기	 디자이너 Figma AI 학습 스터디
 개발자 CS 지식 스터디	 개발자 사프 기술 스택 공유 모임
 디자이너 Nano Banana 스터디	 기획자 Google Analytics 스터디
 기획자 ADsP 자격증 스터디	 개발자 코드 리뷰 스터디

모임을 개설할 때 작성해야 하는 항목들이 너무 많다는 부원들의 불만이 있었습니다.



그래서 직접 리서치를 하고 이를 개선해보고자 했습니다.
먼저 기존 플로우는 다음과 같습니다.



먼저 다음의 가설을 세우고 리서치를 설계했습니다.
스크리닝 설문으로 적합한 인터뷰어를 찾았습니다.

* 리서치 개요

리서치 가설

모임 개설자는 개설시 입력하는 [많은 정보들을 부담]스러워 할 것이다.

리서치 방법

In Depth Interview + User Test

리서치 인원

총 6명

리서치 대상자

동아리 앱을 통해 4회 이상의 모임 개설 경험이 있는 부원

* 리서치 목적

- Obj 1.

모임을 개설하는 전반의 과정 이해
- Obj 2.

단계별 사용자의 어려움/ 소요 시간 파악

* 리서치 질문

- RQ 1.

사용자는 모임을 개설하는 전반의 경험에 만족하는가?
- RQ 2.

사용자는 모임을 개설하며, 어디서 많은 시간을 소모하는가?

Google Sheet에 작성한 실제
리서치 설계서와 스크리닝 설문지입니다.

				스크리닝 1
서비스 이용	1	전체	객관식 복수	SOPT 활동을 34기 에 처음으로 시작하셨나요?
스크리닝 2				
활동 기수	2	전체	객관식 복수	활동했던 SOPT 기수 를 모두 선택해주세요.
스크리닝 3				
개설 여부	3	전체	객관식 단일	플레이그라운드 내 모임 서비스를 통해 모임 개설을 해본 경험 이 있으신가요?
개설 환경	4	3번에 1번	객관식 단일	모임 개설 시, 어떤 환경에서 개설하셨나요?
개설 횟수	5		객관식 복수	모임을 개설해 본 횟수 가 어떻게 되시나요?
크루_모임 개설 서비스 프로세스 개선				객관식 복수 선택 가능)
35기 SOPT의 공식 스터디 개설기간 전, 모임을 개설하는 유저의 경험을 확인하고, 개선이 필요한 부분이 있는지 파악하고자 함.				
모임을 개설하는 전반을 이해하고, 단계별로 사용자가 겪는 어려움이나 만족도를 파악				
RQ 1. 사용자는 모임을 개설하는 전반의 경험에 만족하는가?				
RQ 2. 모임을 개설하고, 운영하는 사용자는 지금 크루의 모임 개설 방식에 만족하는가?				
<< 핵심 질문 이후 꼬리 질문 >> - 모임에 신청한 사람을 승인하는 기준이 어떻게 되는가? - '각오'의 내용이 신청 수락 여부에 영향을 미치는 정도는?				
- 모임 개설자(6명) L G1. 34기 YB로 모임을 처음 개설해본 사용자 L G2. 34기 OB로 모임을 2번 이상 개설해본 사용자				
개설 모임 종류				객관식 단일
파트 내 스터디 - 공부 (4명) 전체 + 파트 내 스터디 - 취미/천목 (2명)				
인터뷰 + UT				객관식 단일
- 리서치 설계 : 9/5 ~ 9/8 (- 9/6 크루팀과 미팅) - 리크루팅 : 9/8 ~ 9/9 - 인터뷰 + UT : 9/9 ~ 9/12(대상자 가능 일정에 따라 변동. asap진행) - 결과 분석 : 9/12-9/13 - 결과 전달 : 9/14일 이내				객관식 복수
- 보상 일정 : 인터뷰 종료 후 1주 이내 - 보상 금액 : - 산정 예정				

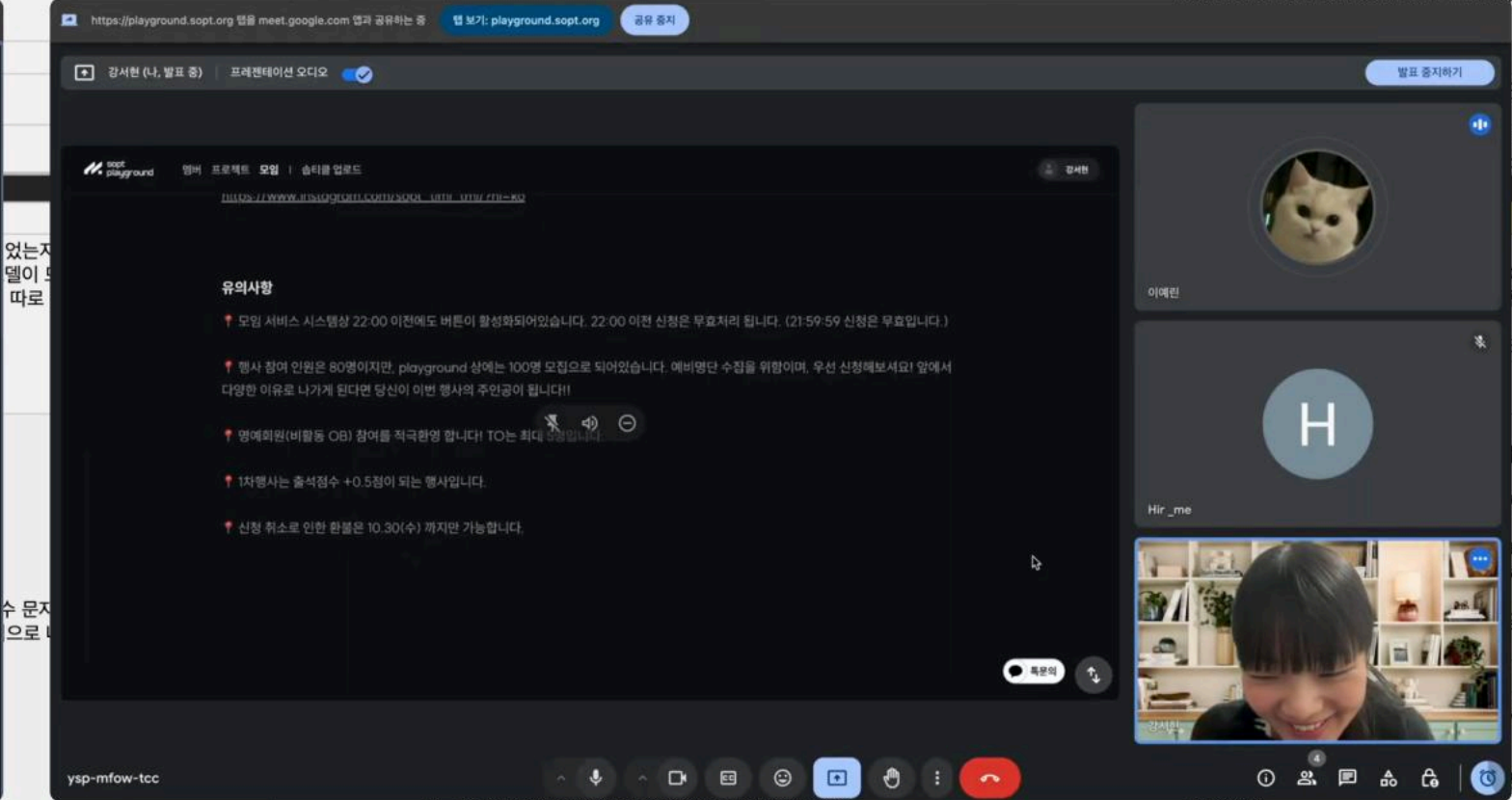
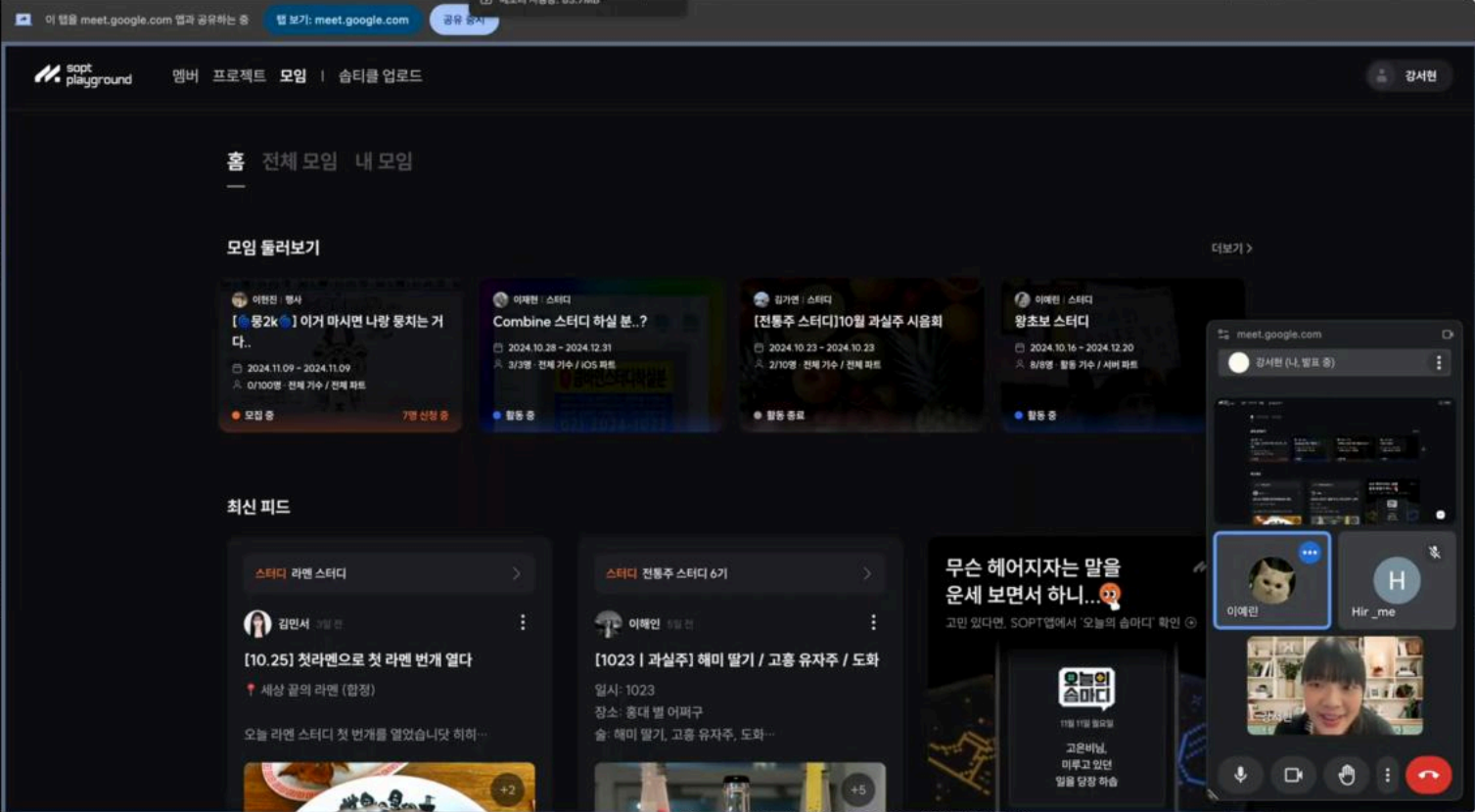
6명을 동아리 부원을 대상으로 비대면 인터뷰와 UT를 진행하고 Google Sheet에 정리했습니다.

Google Sheet에 작성한 실제 인터뷰지입니다.

섹션 1)	1단계_모임 정보/-모임 제목, 모임 카테고리, 소개 이미지, 모임 소개, 진행 방식 소개 링크를 삽입하시는 과정은 어떨까요?	- 노선(출석인증/과제 확인) 링크는 플레이그라운드 내에 삽입하기 보다는 카톡방을 통해서 전달했다. - 모임피드가 아니라 노선으로 출석인증을 했다-> 이유: 플레이그라운드 자체가 익숙치 않아서 피드의 기능을 사용할 생각조차 못했다. "피드"에 대한 기능의 존재 인지를 못했다. - 피드에서 과제를 공유해야하는데, 구성원들의 과제가 대외적으로 공개되기 때문에 피드에서 과제를 공유하고 싶지 않을 수도 있을 것이라 생각이 들었다. - 노선에서는 과제를 공유하기 보다는 과제를 했는지 안했는지만 체크했다.(피그마 링크는 남김.) - 모임소개나 아래있는 진행방식 소개나 마찬가지로, 예시로 간단하게 제목 아래에 적혀있을 좋겠다.
	스터디 진행 방식의 내용을 구성하는 방법이 어떻게 되시나요? ↳ 내용의 단계/순서를 분류하기 위해 어떤 방법을 사용하시나요?	- 자동 넘버링 기능/ 폰트 굵기 차이 주기 정도는 가시성을 위해 있어도 좋을 것 같다. - 글자 수 넘쳤을 때: 당시에는 불편하다고 느끼지 않았지만, 텍스트 필드가 스크롤로 필드가 늘어나기 보다는 필드 영역 자체가 늘어나도 좋을 것 같습니다." -> 그렇기 때문에 텍스트 필드에 예시가 제공되면, 모임 소개/진행방식을 보다 더 쉽게 작성가능할 것 같다. - 모임소개라는 라이팅을 변경해도 좋을 것 같다. - '모임소개 및 진행방식'으로 합쳐도 좋을 것 같다. - 두가지를 구분하려면 확실히 어떤 내용을 써야하는지 먼저 제안을 해주면 좋을 것 같다.
	↳ 구성했던 내용이 효과적으로 작성되었다고 생각하시나요?	
	↳ (아니라면)그 이유는 무엇인가요?	
섹션 2)	2단계_모집 정보 /-모집 기간, 모집 대상	- 모집기간의 기준: 파트 내에서 안내한 정해진 기간이 있었다. - 파트 선택 토글: 여러 파트들이 토글밖으로 버튼으로 나열되어있어도 될 것 같다. - 날짜를 키보드로 작성할 수 있나요?라는 의문을 가졌다. 날짜를 타이핑할 수 있는게 더 편할 것 같다. - 활동기간: 대략적인 SOPT 일정을 보고 대강 이때까지는 할 수 있겠다라고 생각하고 활동기간을 정했다. (구체적으로 활동기간을 고민하지 않았다.)

리서치 주제	
모임 개설 경험 전반	섹션 1)
- 링크가 올바르게 삽입되었는지 확인하지 못할 것이다. => 링크의 확실한 전달모델이 드러나길 원할 것이다. => 링크를 삽입하는 칸이 따로 존재하길 원할 것이다.	1단계_모임 정보 / - 모임 제목, 모임 카테고리, 소개 이미지, 모임 소개, 진행 방식 소개
	링크를 삽입하시는 과정은 어떨까요?
	↳ 링크가 제대로 삽입되었는지, 확인이 가능하셨나요? -> 이때 ㅇㅇ님은 어떻게 대응하셨나요?
	스터디 진행 방식의 내용을 구성하는 방법이 어떻게 되시나요?
	↳ 스터디 내용의 단계/순서를 분류하기 위해 어떤 방법을 사용하시나요?
- 이모티콘, 화살표 등 특수 문자 사용의 접근성이 낮아, 진행 방식(순서)를 효과적으로 나타낼 수 있는 양식에 대한 니즈가 있을 것이다.	↳ 구성했던 스터디 내용이 효과적으로 작성되었다고 생각하시나요?
	↳ (아니라면) 그 이유는 무엇인가요?
	2단계_모집 정보 / -모집 기간, 모집 대상
	캘린더 불편함 호소!
	모집 기간 모달이 뜰 때, 숫자 입력도 동시에 되었으면 좋을 것 같다는 생각. 단순 넘기는 액션보다는, 키보드로 입력하는 것도 충분히 도움이 될 것 같다
	섹션 3)
- '모집 기간'과 '활동 기간'의 개념이 모호할 것이다.	3단계_활동 정보 / -활동 기간, 개설자 소개, 유의사항
	활동기간까지 작성해주셨는데, 앞선 모집기간과 활동기간을 나누어 입력할 때 헷갈림은 없으셨나요?
	혼동이 있을 수 있을 것 같다. 다 똑같이 생겨서, 혼동이 생길 수 있었음.
	섹션 4)
	경험 전반
	모임 개설 양식이 실제 스터디 소개 및 운영 시 필요한 정보들로 구성되었다고 생각하시나요?
	↳ 어떤 정보들이 필요하지 않았나요? / 중요도가 높은 항목은 무엇인가요?
	모임 개설 과정에서 이해하지 못한 부분 / 작성이 어려운 부분이 있으셨나요?
	작성하는 양식 외에, 모임 신청자에게 더 알고 싶은 정보가 있으셨나요?
	모임정보,모집정보,활동정보 총 3개의 단계로 구성되어있다는 사실을 인지하시고 작성하셨나요?
	↳ 단계별 내용의 분류가 적절히 이루어졌다고 생각하시나요?
작성 후 모집글 게시 전에 검토 혹은 확인 과정이 필요할 것	게시된 글이 작성 시 예상했던 것과 다르게 느껴진 적이 있으신가요? 있다면 무엇인가요?
	작성한 글이 사이트에 게시되기까지의 과정에서 어떤 부분이 가장 중요하다고 생각하시나요?
각오 관련 추가 질문	섹션 5)
	각오 관련 질문
	네, 이제 모임 개설에 관해 제가 질문드릴 내용은 거의 다 드린 것 같아요. 마지막으로 추가 질문 몇 개만 더 드리고, 마무리 할게요! ->
	혹시 모임 개설이 아닌, 슬커톤/1,2차 행사를 포함하여 모임을 신청해 본 경험도 있으신가요?
	있다. 플레이스홀더로 써져 있어서(선택사항) 알고 있었다. 카톡방에서나 스터디에서나 자주 쓰이는 것 같아요.

리서치 주제	모임 개설 경험 전반	섹션 1)	1단계_모임 정보/-모임 제목, 모임 카테고리, 소개 이미지, 모임 소개, 진행 방식 소개 링크를 삽입하시는 과정은 어떨까요?	링크 삽입 스터디 노선 (비신청 신청자
			↳ 링크가 제대로 삽입되었는지, 확인이 가능하셨나요? -> 이때 ㅇㅇ님은 어떻게 대응하셨나요? 스터디 진행 방식의 내용을 구성하는 방법이 어떻게 되시나요?	진행 방식 카테고리 매우 단 하지만
			↳ 모임소개나 아래있는 진행방식 소개나 마찬가지로, 예시로 간단하게 제목 아래에 적혀있을 좋겠다.	스터디 단독방 했다. 신청자 간단하
			↳ 구성했던 내용이 효과적으로 작성되었다고 생각하시나요? ↳ (아니라면)그 이유는 무엇인가요?	
		섹션 2)	2단계_모집 정보 /-모집 기간, 모집 대상	활동기 달력이 모집기 캘린더 시작일
		섹션 3)	3단계_활동 정보 /-활동 기간, 개설자 소개, 유의사항	활동기 정기행 스터디 -> 마지
		섹션 4)	경험 전반	개설자 스터디 마라탕 헤더 알



섹션 1)	1단계_모임 정보/-모임 제목, 모임 카테고리, 소개 이미지, 모임 소개, 진행 방식 소개 링크를 삽입하시는 과정은 어떨까요?	- 노선(출석인증/과제 확인) 링크는 플레이그라운드 내에 삽입하기 보다는 카톡방을 통해서 전달했다. - 모임피드가 아니라 노선으로 출석인증을 했다-> 이유: 플레이그라운드 자체가 익숙치 않아서 피드의 기능을 사용할 생각조차 못했다. "피드"에 대한 기능의 존재 인지를 못했다. - 피드에서 과제를 공유해야하는데, 구성원들의 과제가 대외적으로 공개되기 때문에 피드에서 과제를 공유하고 싶지 않을 수도 있을 것이라 생각이 들었다. - 노선에서는 과제를 공유하기 보다는 과제를 했는지 안했는지만 체크했다.(피그마 링크는 남김.) - 모임소개나 아래있는 진행방식 소개나 마찬가지로, 예시로 간단하게 제목 아래에 적혀있을 좋겠다.
	스터디 진행 방식의 내용을 구성하는 방법이 어떻게 되시나요? ↳ 내용의 단계/순서를 분류하기 위해 어떤 방법을 사용하시나요?	- 자동 넘버링 기능/ 폰트 굵기 차이 주기 정도는 가시성을 위해 있어도 좋을 것 같다. - 글자 수 넘쳤을 때: 당시에는 불편하다고 느끼지 않았지만, 텍스트 필드가 스크롤로 필드가 늘어나기 보다는 필드 영역 자체가 늘어나도 좋을 것 같습니다." -> 그렇기 때문에 텍스트 필드에 예시가 제공되면, 모임 소개/진행방식을 보다 더 쉽게 작성가능할 것 같다. - 모임소개라는 라이팅을 변경해도 좋을 것 같다. - '모임소개 및 진행방식'으로 합쳐도 좋을 것 같다. - 두가지를 구분하려면 확실히 어떤 내용을 써야하는지 먼저 제안을 해주면 좋을 것 같다.
	↳ 구성했던 내용이 효과적으로 작성되었다고 생각하시나요?	
	↳ (아니라면)그 이유는 무엇인가요?	

섹션 2)	2단계_모집 정보 /-모집 기간, 모집 대상	캘린더 불편함 호소! 모집 기간 모달이 뜰 때, 숫자 입력도 동시에 되었으면 좋을 것 같다는 생각. 단순 넘기는 액션보다는, 키보드로 입력하는 것도 충분히 도움이 될 것 같다
		활동 기간과 모집 기간을 처음에 혼동해서 이야기할 달력이 2개 있어서 첫번째 기간 설정을 보고 당연히 활동 기간이라고 생각 모집 기간: 모집기간은 일주일정도로 설정했는데, 일주일도 길다고 생각했 캘린더 모달: 시작일 캘린더 모달이 닫히면 자동적으로 종료일의 캘린더 모달이 열리

섹션 4)	경험 전반	모임 개설 양식이 실제 스터디 소개 및 운영 시 필요한 정보들로 구성되었다고 생각하시나요? ↳ 어떤 정보들이 필요하지 않았나요? / 중요도가 높은 항목은 무엇인가요? 모임 개설 과정에서 이해하지 못한 부분 / 작성이 어려운 부분이 있으셨나요? 작성하는 양식 외에, 모임 신청자에게 더 알고 싶은 정보가 있으셨나요? 모임정보,모집정보,활동정보 총 3개의 단계로 구성되어있다는 사실을 인지하시고 작성하셨나요?	잘 구성되 항목을 중 추가적인
-------	-------	--	------------------------

인터뷰와 사용성 테스트를 분석한 결과,
작성 항목간 겹치는 내용이 많고, 주관식 항목이 많은게 문제였습니다.

가설

모임 개설자는 개설시 [입력하는 많은 정보들을 부담]스러워할 것이다.

검증 결과

사용자는 [주관식 작성 항목]이 너무 많다고 느꼈어요.

검증 결과

여러 항목들의 [각 내용을 구분해서 작성하는데에 어려움]이 있었어요.

Notoion으로 인터뷰 결과를 정리하고, 팀원들 앞에서 발표했어요.

1. 모임 정보 입력 단계
(모임 제목, 모임 카테고리,소개 이미지,모임 소개,진행 방식 소개)

1 [모임 소개],[진행 방식 소개]를 구분해서 작성하는데 어려움이 있었어요.

- 모임 소개와 진행 방식의 분류기준이 모호해요. ⭐⭐⭐⭐
- (VOC) 모임소개/ 진행 방식 구분이 모호한 것 같아요. 그렇기 때문에 텍스트 필드에 예시가 제공되면, 모임 소개/진행방식을 보다 더 쉽게 작성가능할 것 같아요.
 - (VOC) 텍스트 필드 내에 형식이나 가이드 라인이 예시로 적혀있으면 보다 더 편하게 작성하기 편할 것 같아요. 이전에는 다른 모임들 참고해서 작성했었어요.
 - (VOC) 진행 방식이 모임 소개에 포함된다고 생각한다. 그래서 애매하다고 느껴요.
진행 방식 그대로 가고, 모임 소개 이름을 다른 경로 교체하는건 어떤지.각 역할이 확실하게 구분된다면 나뉘야겠지만 아니라면 합치는게 좋아보여요.
 - (VOC) 모임 소개, 진행 방식 소개에 예시를 제공하면 좋을 것 같아요.
작성시 넘버링, 이탤릭체등 (밑줄, 글머리) 가시성을 위한 입력 방식이 추가되면 좋을 것 같아요.

key 2 진행방식 작성 시 모임 서비스 뿐만 아니라 노션과 같은 외부 서비스를 함께 사용하는 모습을 보였어요.

- 이유 1 모임 서비스만으로는 스터디 정보를 다 담기에는 어려움을 느끼고 있었어요. ⭐⭐⭐⭐⭐
- 이유 2 노션이 포함된 파트 내 스터디의 사전 프로세스가 있었어요.
- 이유 3 모임의 독립적 공간이 필요해요.

▶ 주요 voc는 다음과 같았어요.

2. 모집정보 단계 (모집 기간, 모집 대상)

key1 모집 기간을 입력하며 캘린더 입력 방식에 어려움을 느끼고 있었어요.

캘린더에 날짜를 직접 입력할 수 있으면 편할 것 같아요.

- (VOC) 모집 기간 모달이 뜰 때, 숫자 입력도 동시에 되었으면 좋을 것 같아요.
- (VOC) 단순 넘기는 액션보다는, 키보드로 입력하는 것도 충분히 도움이 될 것 같아요.
- (VOC) "날짜를 키보드로 작성할 수 있나요?" 날짜를 타이핑할 수 있는게 더 편할 것 같아요.
 - ▶ 추가 행태

key2 모집 대상이 한눈에 잘 보이길 원하고 있었어요.

key2 모집 대상이 한눈에 잘 보이길 원하고 있었어요.

- (VOC) 파트 토크 살짝이지만 스크롤해서 다중선택 해야하는데, 애초에 각 파트들이 버튼 형식으로 한눈에 나열되어 있으면 한눈에 보이고 누르기 편하니까 더 편하지 않을까?라는 생각이 들었어요.
- (VOC) 여러 파트들이 토크받으로 버튼으로 나열되어있어도 될 것 같아요.
- (VOC) 모집 대상 파트를 스크롤 해야한다는 점이 불편해요. 버튼이 터치 없이 노출되어 있으면 좋을 것 같아요.

3. 활동 정보 (활동 기간,개설자 소개 (+멘토 구하기) , 유의사항)

key1 모집 기간과 달리 활동 기간 선정에 고려사항이 많았어요. ⭐⭐⭐⭐⭐

- 모집기간은 파트 내에서 가이드라인이 제공돼요.
따로 제 의견은 투영하지는 않았고, 임원진 측의 가이드라인이 있다보니 그걸 따라했던 것 같아요.
모집기간은 파트 내에서 안내한 정해진 기간이 있었어요.
모집 기간은 파트내 스터디이기 때문에 파트에서 정해진 기간대로 설정했어요.

이후,
움을 확인하고 싶다는 의견이 있었어요.

무게(분위기) 특성상 바로 개설하게 되는 건지는 모르겠지만 스터디장 입장에서는 어떤 정보를 입력했는지,들이 잘 입력됐는지 확인하는 모달이 한 번 있었으면 좋을 것 같다는 생각을 했다.

성 후, 미리보기 기능이 있었으면 좋겠다.
어버리면 부담스러워서 미리보기 기능이 있으면 부담이 덜어질 것 같다.

설자(😄)와 신청자(👉) ,
서 바라본 [각오] 입력에 대한 의견

선정에 영향을 얼마나 끼쳤나요?

멤버 선정에 필요하지 않았어요. 또, 인원이 많은 경우 순'으로 신청 받기 때문에 더 그런 것 같아요.

따라 필요할 수도 있을 것 같기는 해요.

해서 그런지, 각오에 대한 필요성이 막 크지는 않았던 파트 내에서 알고 있는 사이고, 어떤 사람인지 다 알고 (스터디원으로)승인에 있어서 중요하지 않았던 것 같

이 있었는지 기억이 잘 안나는 것 같아요. 여차피 갈원을 뽑는 경우라, 별로 신경쓰지 않았던 것 같아요."

장에서는 선착순으로 신청을 받기에 의미가 신청 수를 미치지 않는다고 생각해요. 하지만, 신청자들을 모하에 사람이 정해진 사람보다 더 많이 신청했다면,유무로 볼 것 같아요."

? 각오를 작성해야하는 것을 알고 있었나요? 어떤 내용을 주로 작성 하셨나요?

👉 신청자 입장
각오 작성을 해야한다는 것은 미리 알고 있었어요.
+ :: 다만, 각오가 스터디 신청에 영향이 간다고 생각하지 않아서 일반적인 내용을 적었던 것 같아요.

💬 실제 주요 VOC

" (중간생략)... 또 카톡방에서 윤팀장님이라 파트장님들이 각오가 선택사항인걸 알려주셔서, 그냥 작성하지 않고 넘겨도 된다는 것도 알고 있었어요."

" 각오 작성이 있다는 것을 미리 알고 있었어요. 또, 파트장님이 단톡방에서 각오 안적어도 된다고 말씀을 미리주셔서 그것도 알고 있었어요."

"각오를 작성할 때, 정성스럽게 작성하지는 않았어요. 인원수가 충분한 스터디들을 신청했기에 '각오'때문에 스터디에서 떨어질 것이라는 생각은 안들었던 것 같아요.
그래서, '열심히 / 적극적으로 참여하겠다'라는 형식적인 각오를 작성했던 것 같아요. "

1 이렇게 개선했어요!

작성 항목을 줄이고,
항목의 특성에 따라 명쾌하게 다시 분류했습니다.

2 이렇게 개선했어요!

각 항목의 Placeholder에, 사용자가 어떤 내용을
입력해야 할지 힌트 메시지를 추가했습니다.

* 작성 항목 삭제 및 재분류

As Is		To Be		
1 필수 정보	2 추가 정보	1 모임 정보	2 모집 정보	3 추가 정보
모임 이름	공동 모임장	모임 이름	모집 기간	공동 모임장
모임 카테고리	모임장 소개	모임 카테고리	모집 대상	모임장 소개
모임 키워드	유의사항	모임 키워드		환영 태그
소개 이미지	환영 태그	소개 이미지		
모임 소개		모임 소개		
진행 방식				
모집 기간				
모집 대상				

* Placeholder 로 작성 내용에 대한 힌트 메시지 추가

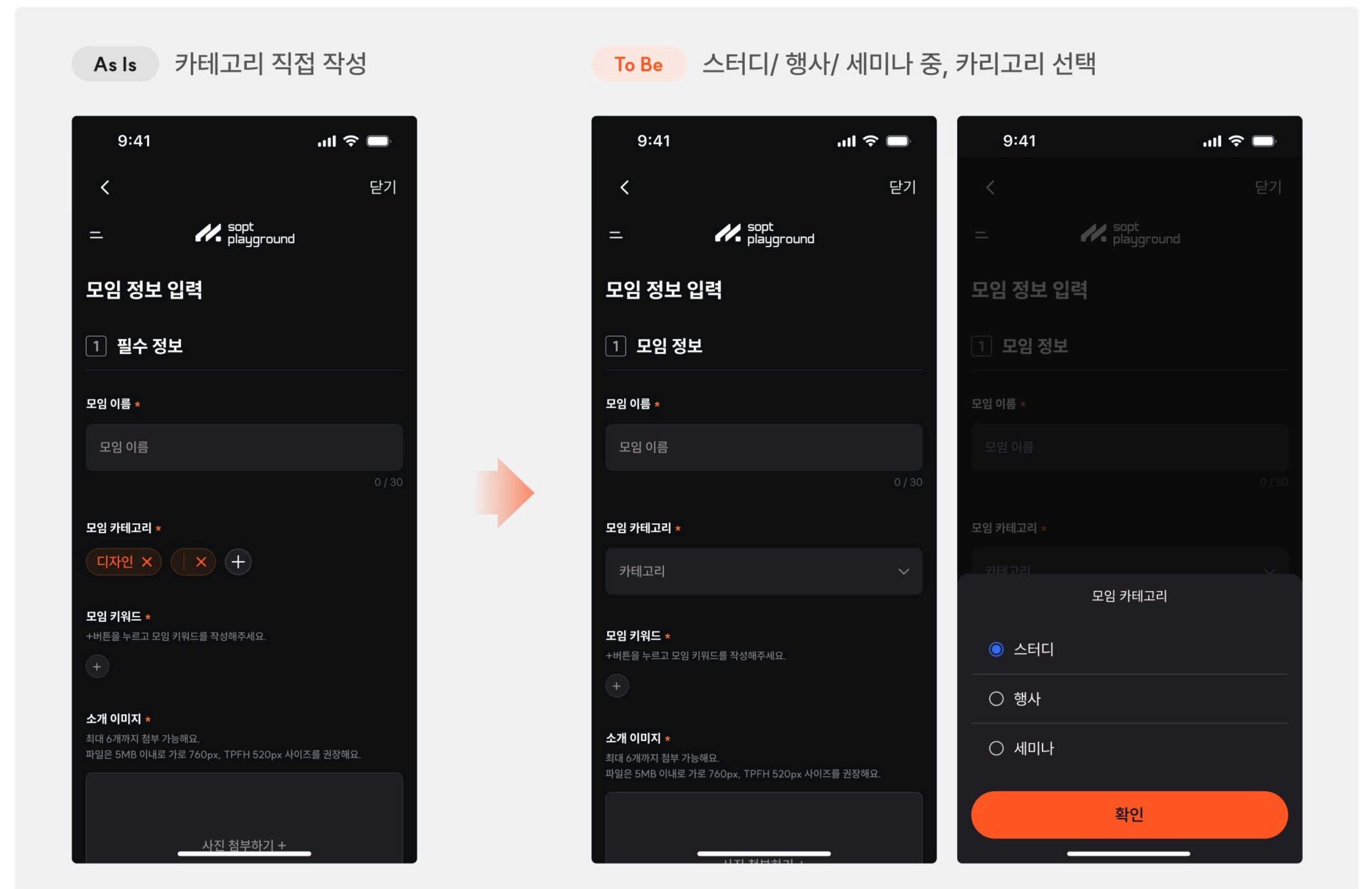
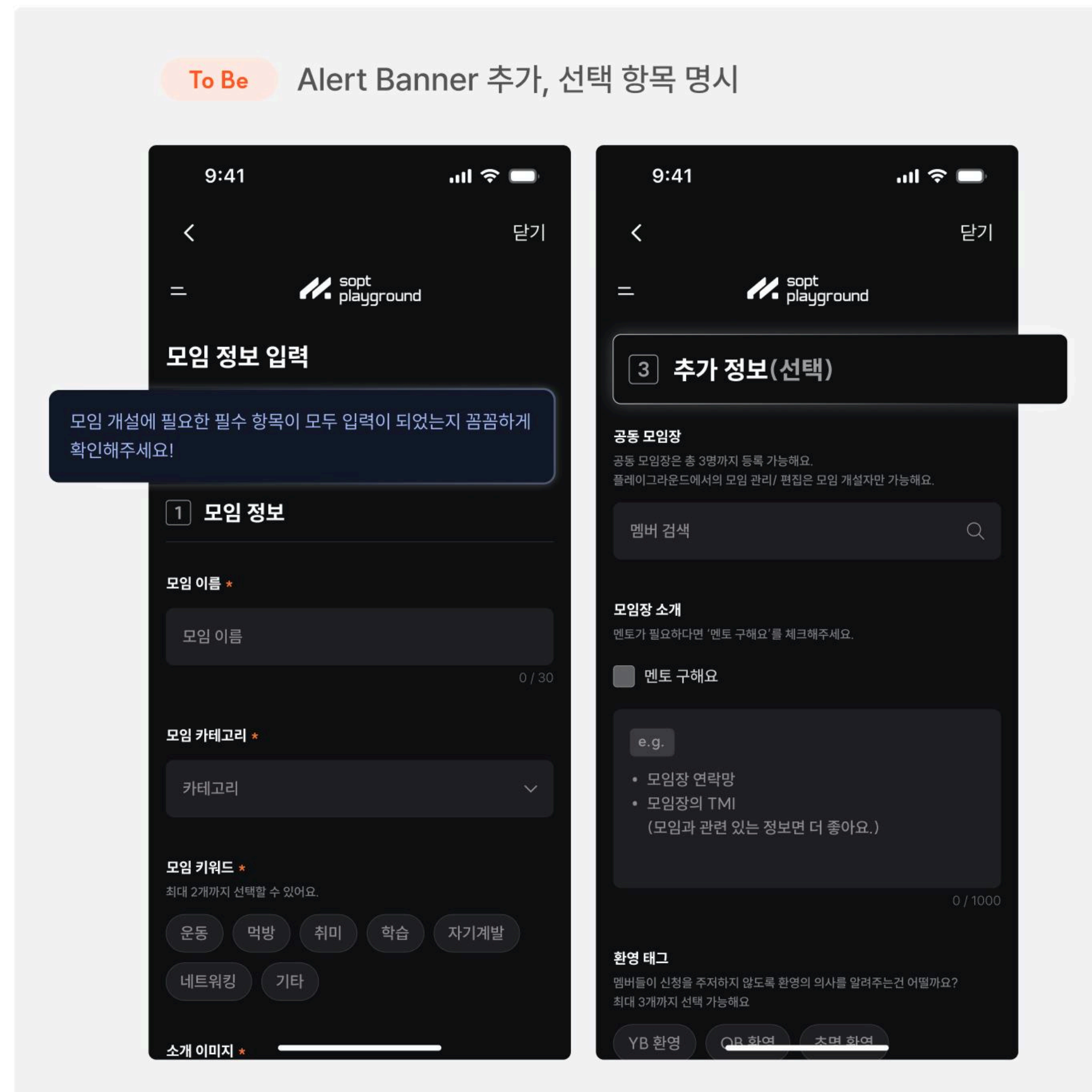
As Is	To Be
모임 소개 * 모임을 1000자 이내로 소개해주세요. 모임장 소개 멘토가 필요하다면 '멘토 구해요'를 체크해주세요. ☐ 멘토 구해요 모임장을 1000자 이내로 소개해주세요.	모임 소개 * e.g. • 모임 성격 • 모임 개설 배경/ 목적 • 모임의 효능 모임장 소개 멘토가 필요하다면 '멘토 구해요'를 체크해주세요. ☐ 멘토 구해요 e.g. • 모임장 연락처 • 모임장의 TMI (모임과 관련 있는 정보면 더 좋아요.)

③ 이렇게 개선했어요!

작성 완료를 유도하는 Alert Banner를 추가하고,
선택 항목은 서브 타이틀로 명시했습니다.

④ 이렇게 개선했어요!

주관식으로 직접 작성해야 하는 모임 카테고리 항목을
선택할 수 있도록 객관식으로 변경했어요.



SaaS

생산성

Morib, SaaS형 생산성 서비스 릴리즈

Morib은 데스크탑 작업 시 작업에 필요한 서비스만 사용할 수 있도록 제한해 **집중을 돕는 SaaS 기반의 생산성 서비스**입니다.

10명의 팀원들과 6개월에 걸쳐 개발하고 직접 **릴리즈**까지 했으며, 현재 1년째 운영 중입니다. **Maze로 사용성 테스트**를 진행해 문제 구간을 확인하고, 온보딩 단계를 줄여 핵심 기능에 빠르게 접근할 수 있도록 개선했습니다.

다운로드 링크 | <https://www.morib.in/>

비헨스 링크 | <https://www.behance.net/gallery/208920433/Morib->

기간 | 2024.08-2025.01

팀원 | 디자인 2명, 기획 2명, 프론트엔드 4명, 백엔드 2명

디자인 기여도 | 70%



What ?

Morib은 어떤 서비스일까?

Morib은 데스크탑 작업 시,
작업에 필요한 서비스만 사용하도록 제한해
집중을 도와주는 생산성 서비스입니다.

How ?

Morib을 어떻게 사용할까?

오늘 할 일과 사용할 서비스 세트를 등록하고 타이머를
시작하면, 미리 등록한 서비스에만 접근 가능합니다.
다음은 타이머에 진입하기까지의 과정입니다.

[타이머 진입 과정, 5단계]

- 1 온보딩 '작업 시 사용할 서비스' 등록
- 2 홈화면 할 일 카테고리 생성
- 3 홈화면 오늘 할 일 선택
- 4 타이머 '사용할 서비스 세트' 선택
- 5 타이머 타이머 시작

But !

Morib의 문제

Morib 앱을 처음 실행할 때, **타이머 시작까지의 과정이 너무 길었습니다.**
이로인해 초기 유저들의 빠른 서비스 이탈이 예상되었습니다.

타이머를 시작하기까지 단계가 너무 많고, 시간이 오래 걸렸습니다.
그래서 다음과 같이 문제를 정의하고, 리서치를 시작했습니다.

문제 정의

타이머 시작 전 설정 과정이 복잡해 핵심 기능(타이머) 진입까지 오래 걸렸습니다.

😞 “타이머 한 번 시작하려는데 설정할 게 너무 많아요.”

😓 “빨리 집중하고 싶은데 시작까지 오래 걸려요.”

😞 “단계가 많아서, 중간에 나가게 돼요.”

리서치 설계

다음의 가설을 세우고 시나리오를 기반으로 9명의 사용자들을 대상으로 사용성 테스트를 진행했습니다.

가설 Morib 앱을 처음 실행할 때, 타이머 실행까지의 과정에 어려움을 느낄 것이다.

미션 홈화면에서, "할일 이름1"이라는 할 일을 추가하고, 타이머를 시작해주세요.

- 1 '작업 시 사용할 서비스' 등록
- 2 할 일 카테고리 생성
- 3 오늘 할 일 선택
- 4 '사용할 서비스 세트' 선택
- 5 타이머 시작

리서치 결과

가설대로 타이머 시작까지 시간이 많이 걸렸고, 그 경로에서 많은 혼란이 있었습니다.

사용성 점수	총 응답자	미션 성공률	평균 시간
72점	9명	71.4%	47.3s

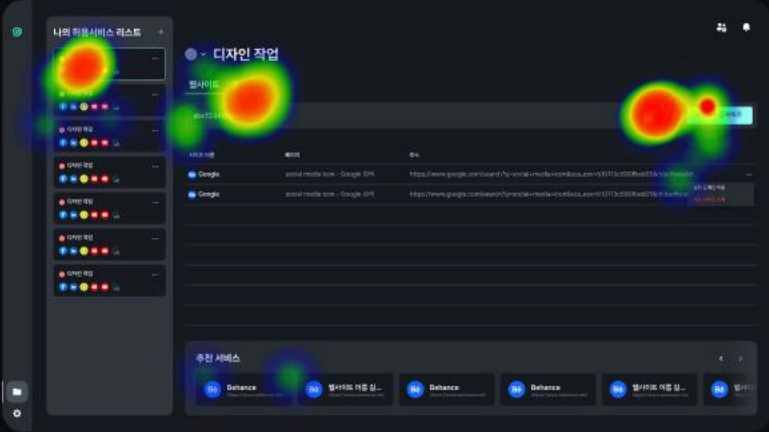
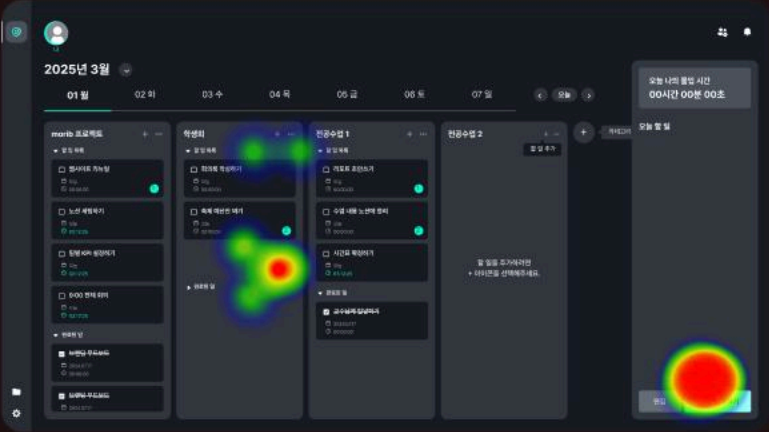
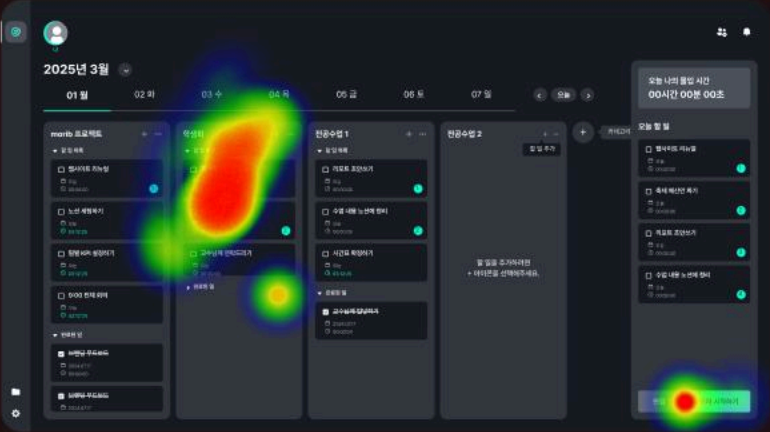
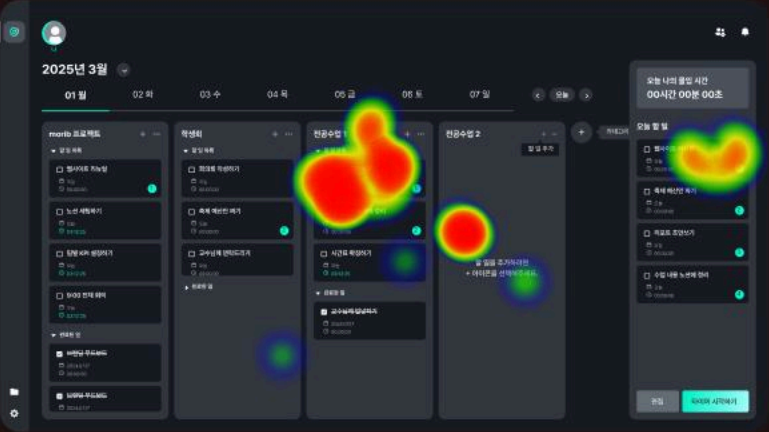
+ 성공률 자체는 높았음.
타이머 진입 후에는, 사용자의 혼란이 적음.

- 타이머 진입까지 필요 이상의 시간 소모.
홈 화면 에서, 타이머 진입 경로(CTA 버튼)를 찾지 못함

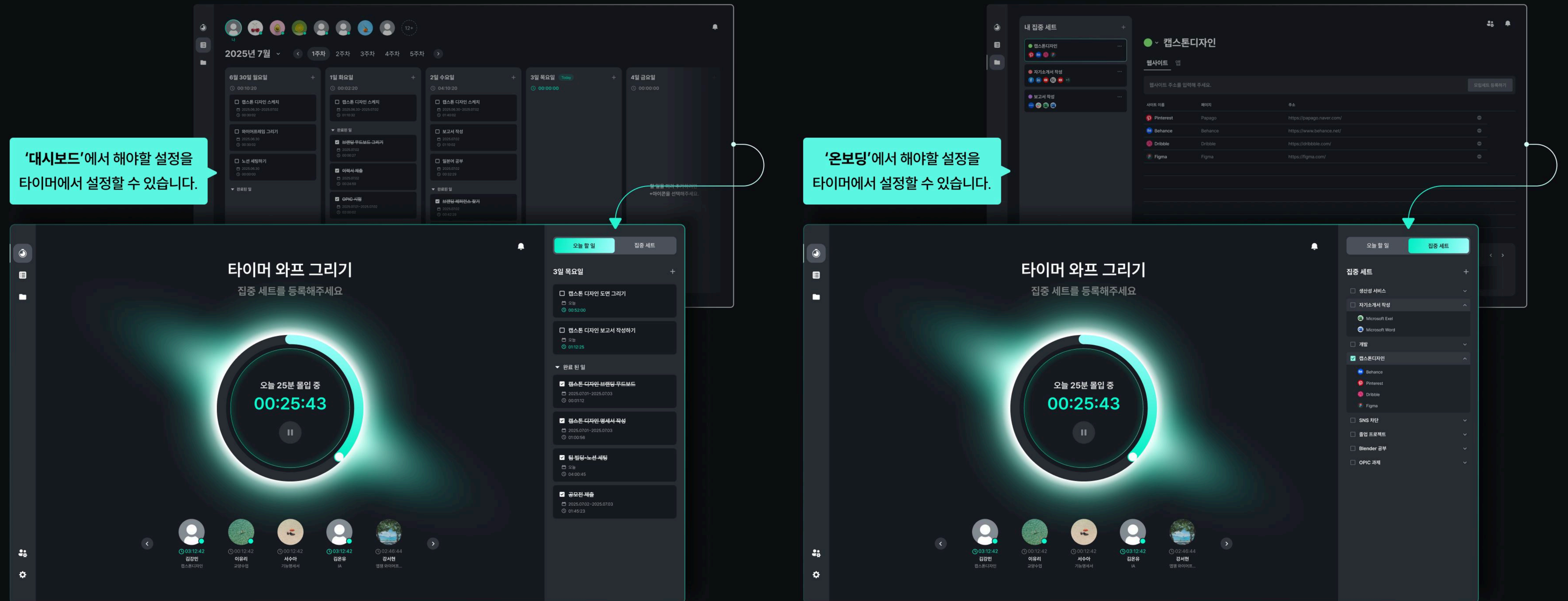
Maze로 사용성 테스트 결과를 자세히 분석했습니다.

타이머 진입 전 '대시보드'에서 불필요한 설정 단계가 많았고, 이를 줄여야 했습니다.

여기서 너무 많은 시간이 걸렸습니다. | 평균 소요 시간 19초

	# Screen 1-온보딩	# Screen 2-대시보드	# Screen 3-대시보드	# Screen 4-대시보드	# Screen 5-타이머
스크린					
평균 소요시간	3s	6s	7s	6s	2s
잘못된 클릭	4%	16%	18%	24%	0%
사용성 점수	92	75	70	73	100

대시보드·온보딩 단계에서 해야할 설정을, 타이머 우측 사이드 패널에서도 할 수 있도록 개선했습니다.
빠르게 타이머에 진입한 뒤에 필요한 설정을 이어서 진행할 수 있습니다.



개선 후, 타이머 진입까지 5단계를 2단계로 줄였습니다.

다시 진행한 사용성 테스트에서 타이머 진입까지의 시간을 15초 줄이고, 사용성 점수를 17점 올릴 수 있었습니다.

[평균 시간]

47.3s » 32.4s

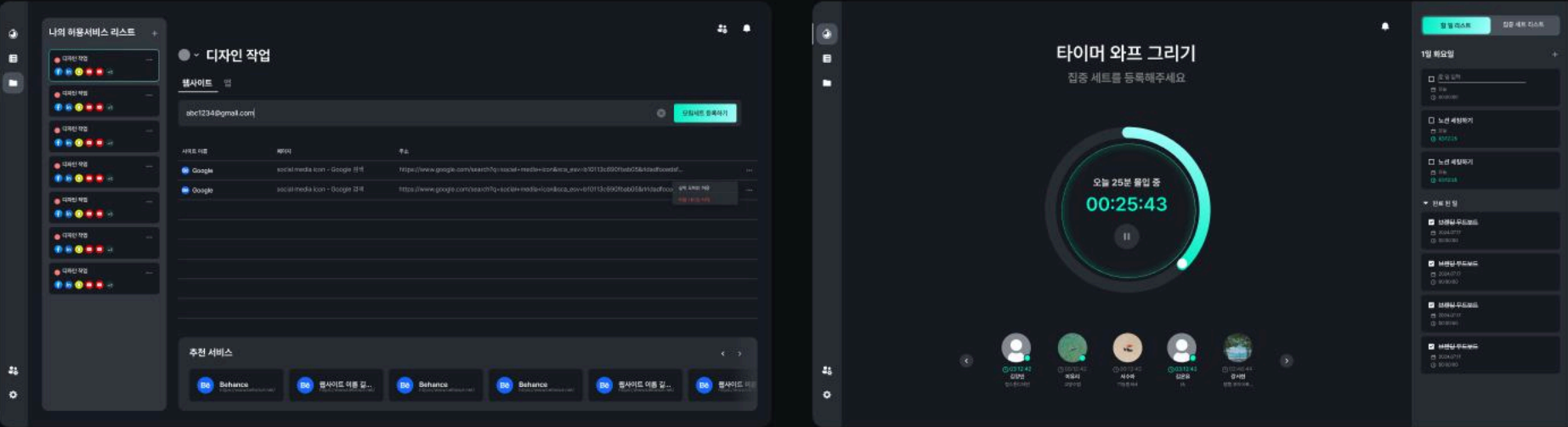
[사용성 점수]

72점 » 89점

* 사용성 점수는 오클릭률, 소요 시간, 미션 성공 여부를 종합적으로 반영해 산정한 점수입니다.

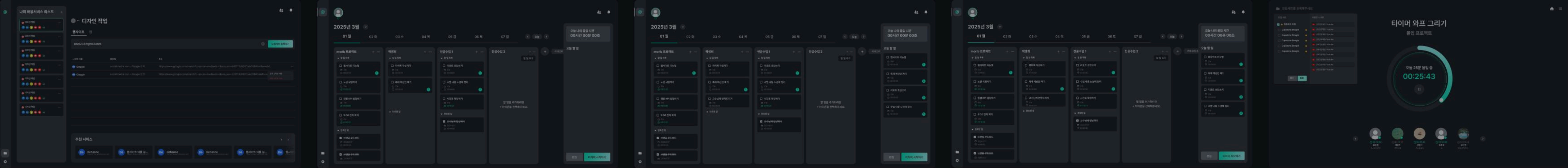
TO BE: 2 Step

- 1 온보딩에서, 사용할 서비스 세트 등록
- 2 오늘 할 일 선택 + 사용할 서비스 세트 선택



AS IS: 5 Step

- 1 온보딩에서, 사용할 서비스 세트 등록
- 2 대시보드에서, 할 일 카테고리 생성
- 3 대시보드에서, 할 일 생성
- 4 대시보드에서, 할 일 리스트업
- 5 타이머에서, 사용할 서비스 세트 선택

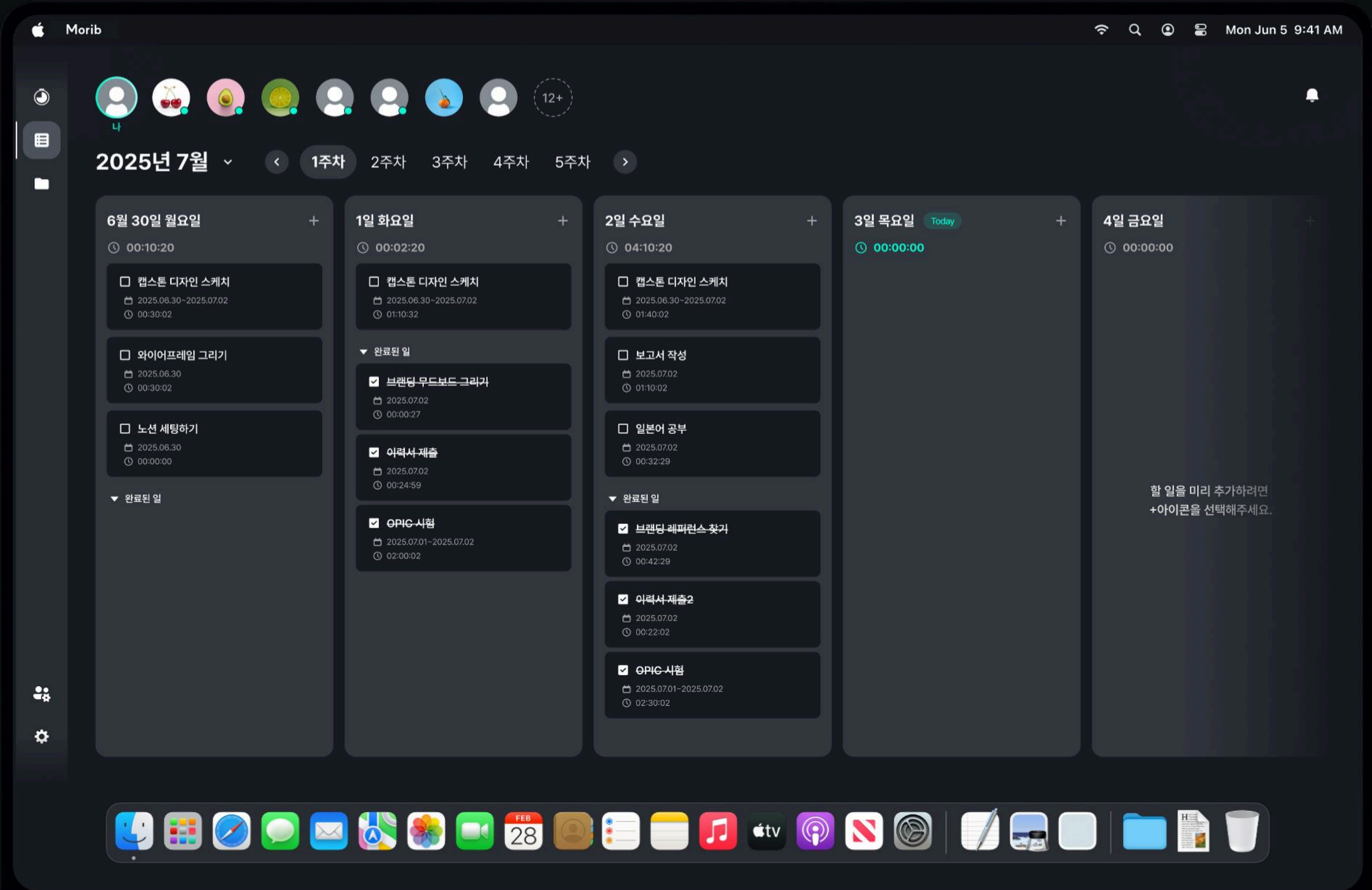


이렇게 서비스를 개선했습니다.
다음부터는, 서비스의 각 기능에 대한 설명입니다.



Morib 랜딩페이지 바로가기 <https://www.morib.in/>

Morib 비핸스 바로가기 <https://buly.kr/2qZ805v>



먼저 작업에 온전히 집중할 수 있도록,
'작업 시 사용할 서비스'를 등록하고 이를 묶어서 하나의 집중 세트로 관리합니다.

🔔

📁

⚙️

내 집중 세트

● 캡스톤디자인

● 자기소개서 작성

● 보고서 작성

🟢 ▼ 캡스톤디자인

🌐 웹사이트

📱 앱

🌐 웹사이트

https://kr.pinterest.com/

모립세트 등록하기

사이트 이름	페이지	주소	
Pinterest	Papago	https://papago.naver.com/	⊖
Behance	Behance	https://www.behance.net/	⊖
Dribbble	Dribbble	https://dribbble.com/	⊖
Figma	Figma	https://figma.com/	⊖

디자인 추천 서비스

Behance

Pinterest

Figma

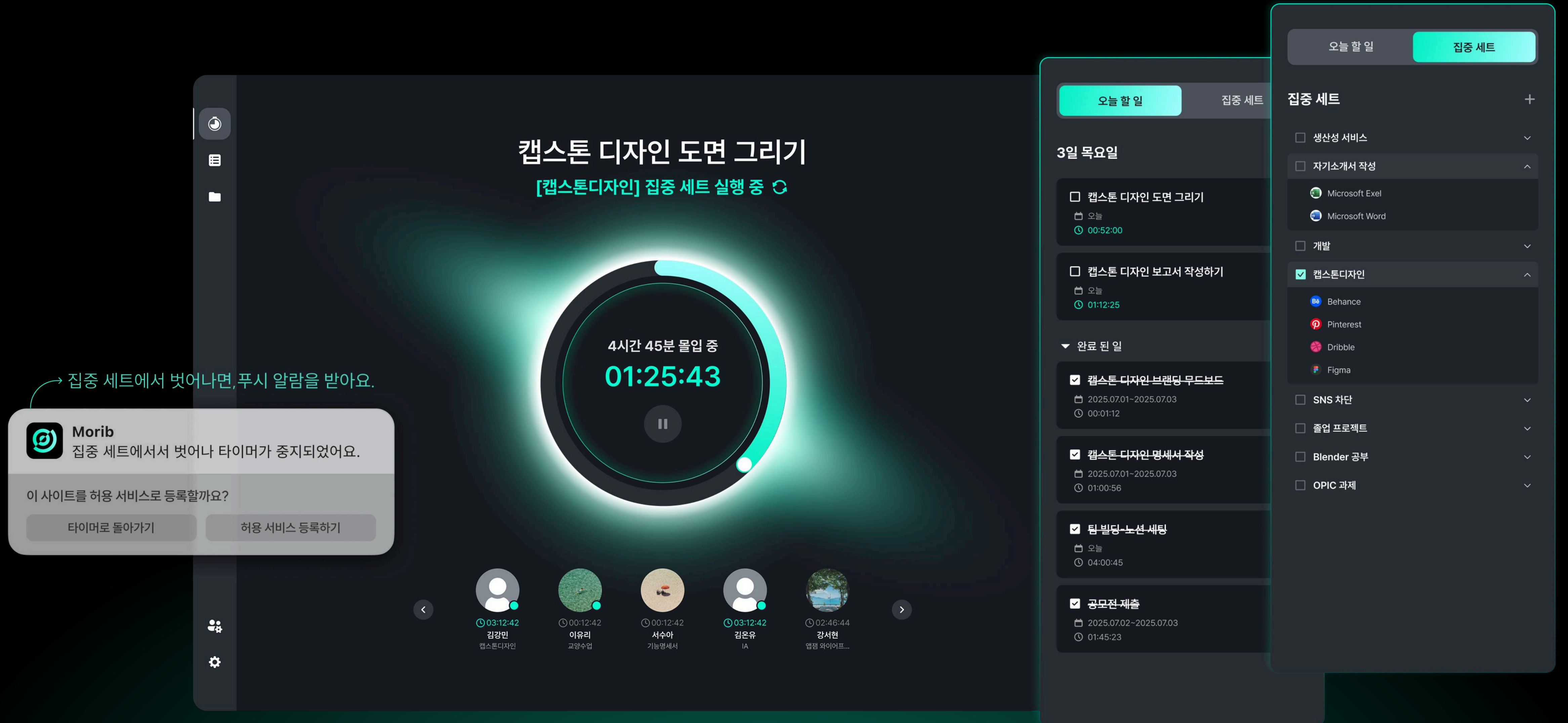
Notefolio

Dribbble

작업에 필요한 서비스들만 등록을 해서 작업에 방해되는 서비스는 차단하고 작업에만 집중해요.

각 작업 분야에 적합한 서비스를 추천해줘요.

만들어둔 집중 세트(작업에 사용할 서비스 세트)를 선택하고, 오늘 할 일을 추가합니다.
타이머를 시작해 작업에 집중할 수 있습니다.



대시보드에서는 날짜별로 해야 할 일을 정리하고 추가할 수 있습니다.

나

12+

2025년 7월

< 1주차 2주차 3주차 4주차 5주차 >

6월 30일 월요일

00:10:20

캡스톤 디자인 스케치

2025.06.30~2025.07.02

00:30:02

와이어프레임 그리기

2025.06.30

00:30:02

노션 세팅하기

2025.06.30

00:00:00

완료된 일

1일 화요일

00:02:20

캡스톤 디자인 스케치

2025.06.30~2025.07.02

01:10:32

완료된 일

브랜딩 무드보드 그리기

2025.07.02

00:00:27

아력사 제출

2025.07.02

00:24:59

OPIC 시험

2025.07.01~2025.07.02

02:00:02

2일 수요일

04:10:20

캡스톤 디자인 스케치

2025.06.30~2025.07.02

01:40:02

보고서 작성

2025.07.02

01:10:02

일본어 공부

2025.07.02

00:32:29

완료된 일

브랜딩 레퍼런스 찾기

2025.07.02

00:42:29

아력사 제출2

2025.07.02

00:22:02

OPIC 시험

2025.07.01~2025.07.02

02:30:02

3일 목요일 Today

00:00:00

할 일 입력

2025.07.03~2025.07.05

00:00:00

4일 금요일

00:00:00

2025.07.03

2025.07.05

2025년 7월 < >

월	화	수	목	금	토	일
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

기간 설정 ☒

친구 추가를 하며, 서로의 워크스페이스를 구경하고 서로의 작업시간과 작업내용을 공유할 수 있습니다.

친구

친구 목록

친구 요청

abc1234@gmail.com

친구 요청

받은 요청

김온유

onyou1234@gmail.com

수락

보낸 요청

김시현

sihyun@gmail.com

요청 취소

이유한

uhan1103@gmail.com

요청 취소

알림

김온유

onyou1234@gmail.com

수락

거절

이수민

sumin@gmail.com

수락

거절

김강현

hyun112@gmail.com

수락

거절

이수만

suman@gmail.com

수락

거절

03:12:42

김강민

웹스톤디자인

00:16:42

이유리

창의 컴퓨팅

00:12:02

한고은

기능명세서

01:12:23

김온유

IA

04:22:02

김서현

IA

00:12:09

이수민

IA

02:46:44

강서현

졸전

00:52:16

서수아

기능명세서

03:12:42

김푸른

IA

Trabble: 채팅 기반 여행 기록 앱 릴리즈

Trabble은 여행 중 나와의 대화를 기록하는 채팅 앱입니다. 체코 교환학생 시절, 8개월간 '나에게 보내기' 채팅방에 일기를 작성한 저의 경험으로부터, 현지에서 만난 개발자 친구와 기획부터 디자인·개발·배포까지 한 서비스입니다.

App Store 링크 | <https://apps.apple.com/app/id6745779775>

팀원 | 디자인 1명, 개발 1명

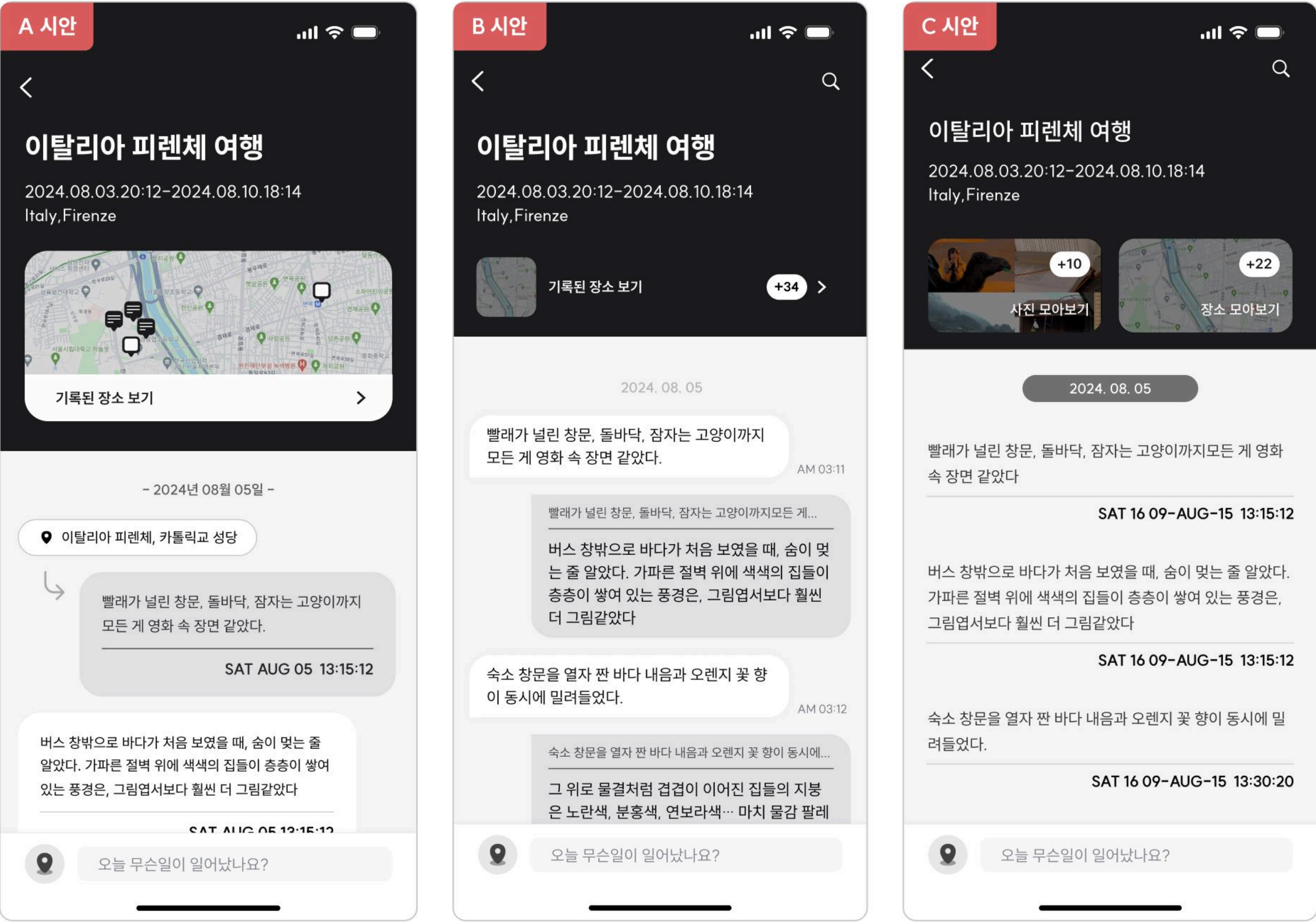
기간 | 2024.06-2025.06

디자인 기여도 | 100%



어떤 채팅형식이 여행 기록에 적합한지 알기 위해 4가지 시안을 가지고 A/B 테스트를 했습니다.
교환학생에서 여행을 같이 갔던 11명의 친구들을 대상으로 인터뷰와 테스트를 진행했습니다.

4개의 시안



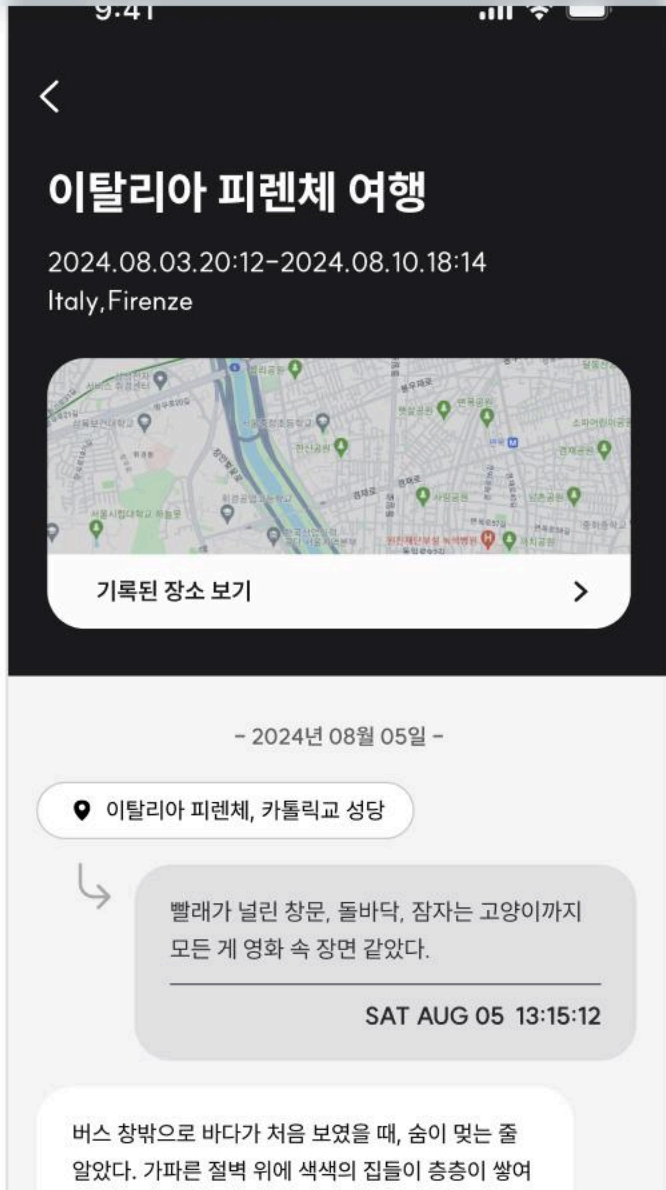
5가지 평가지표

- 기준 1 가독성**
'필요한 정보가 한 눈에 파악되었다.'
'내용 위에 정리가 잘 되었다.'
'중요한 텍스트가 잘 강조되어 한눈에 들어왔다.'
- 기준 2 정보 포괄성**
'정보가 균형 있게 제공되어 있었다.'
'내용이 이해되기에 부족함이 없었다.'
'불필요한 정보 없이 핵심 내용이 모두 담겨 있다.'
- 기준 3 익숙함**
'UI를 처음 접해도 별다른 학습 없이 쉽게 사용할 수 있었다.'
'정보의 배치와 흐름이 예상과 크게 다르지 않다.'
'사용자가 이전에 경험한 디자인/패턴과 유사하다..'
- 기준 4 직관성**
'전체 UI의 구조와 흐름이 이해하기 쉬웠다.'
'사용자 흐름을 쉽게 파악할 수 있다.'
'직관적으로 기능을 파악할 수 있다.'
- 기준 5 시각적 명료성**
'중요한 정보가 시각적으로 잘 구분된다..'
'정보가 명확하게 구분되어 있다.'
'색상, 글씨 크기, 아이콘 등이 의미를 명확히 전달한다.'

B시안의 Chatting이 가독성이 높고, D시안의 Header가 익숙하고 직관성이 높다고 평가받습니다.

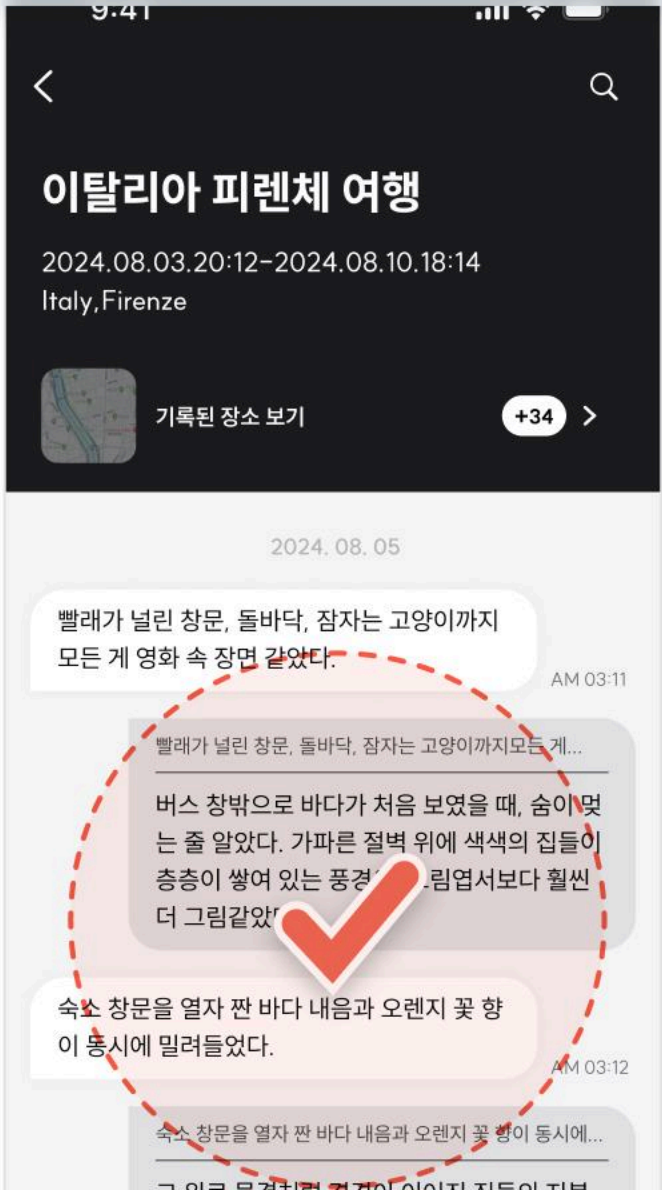
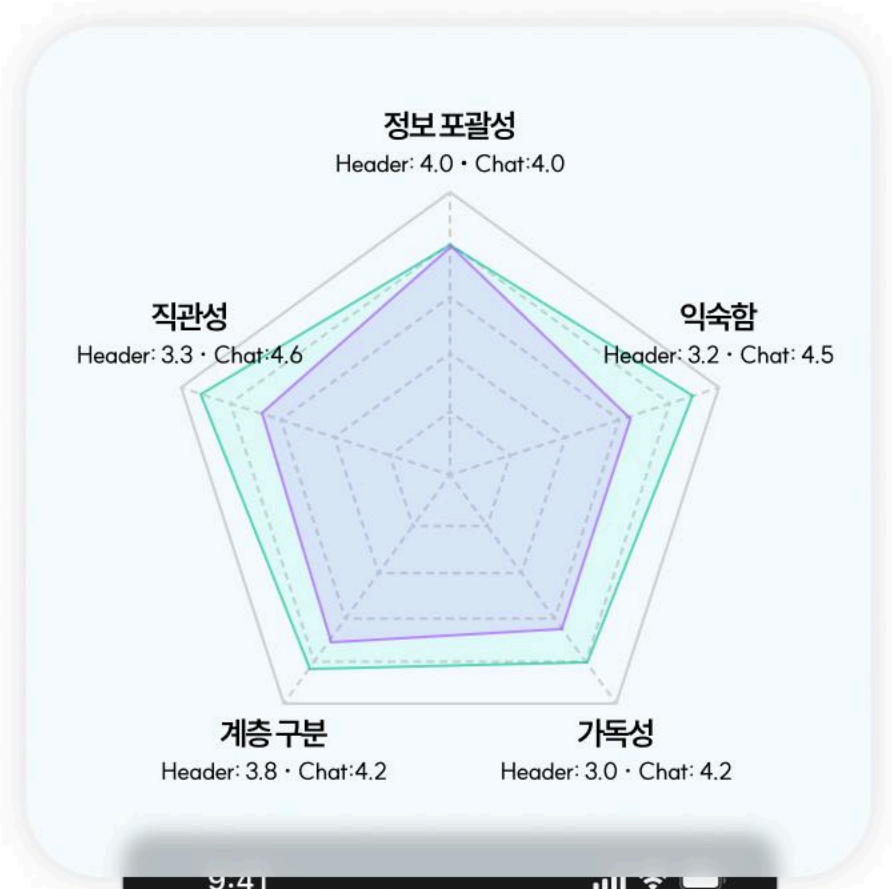
▶ Score: 3.65점

Header Score:3.4점 • Chat Score: 3.9점



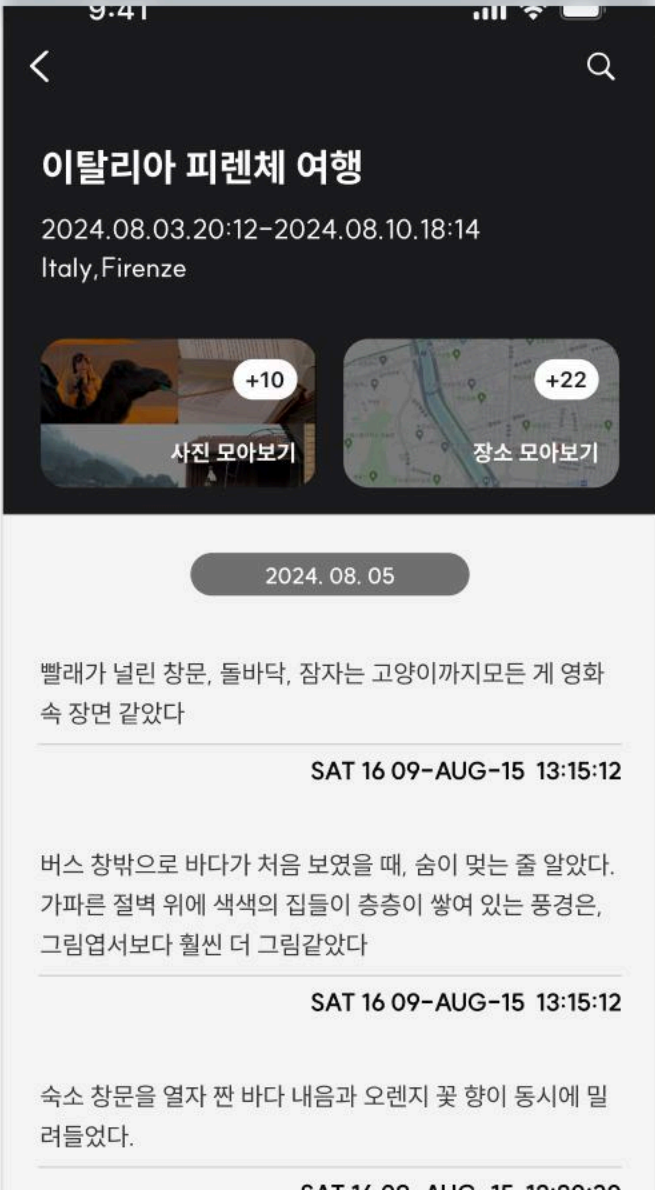
▶ Score: 3.9점

Header Score: 3.5점 • Chat Score: 4.3점



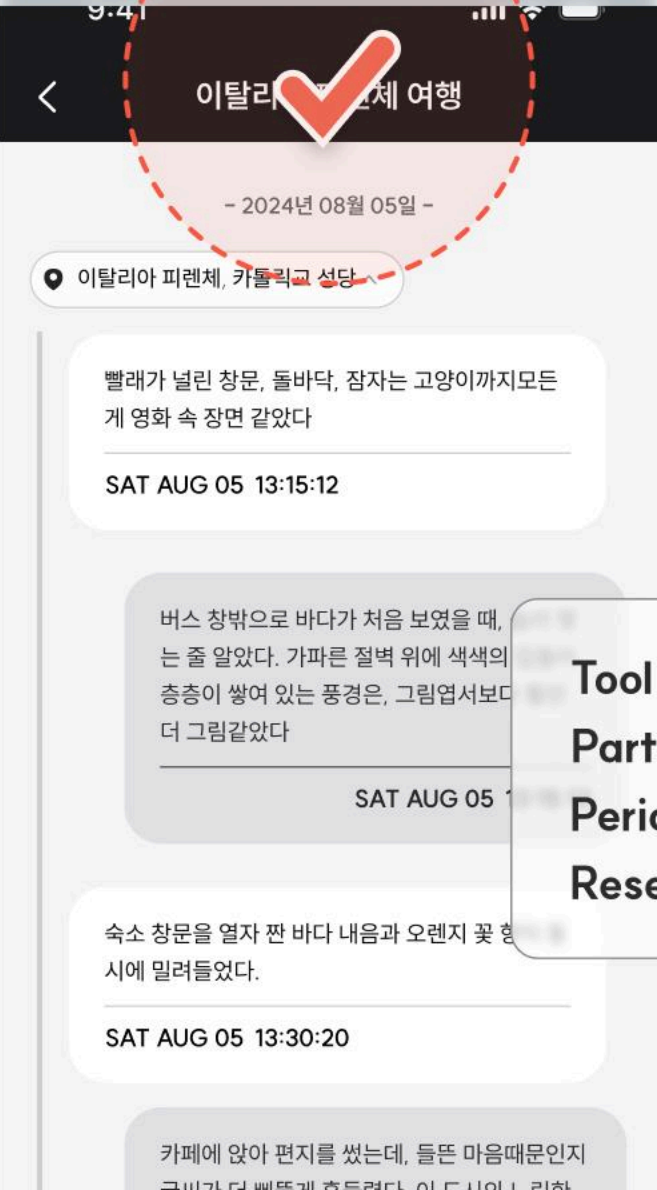
▶ Score: 3.4점

Header Score: 3.5점 • Chat Score: 3.3점



▶ Score: 3.65점

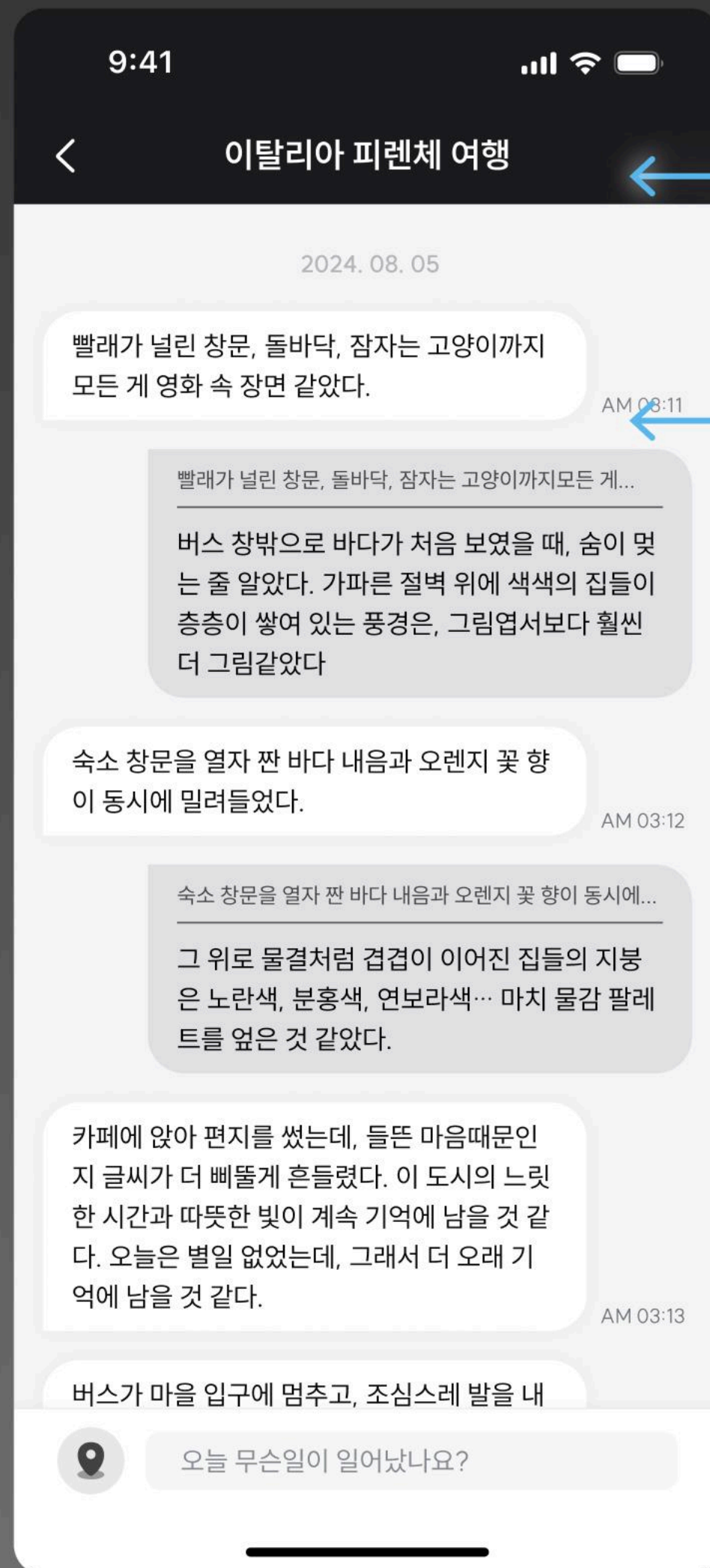
Header Score: 4.0점 • Chat Score: 3.3점



Chatting ●
Header ●

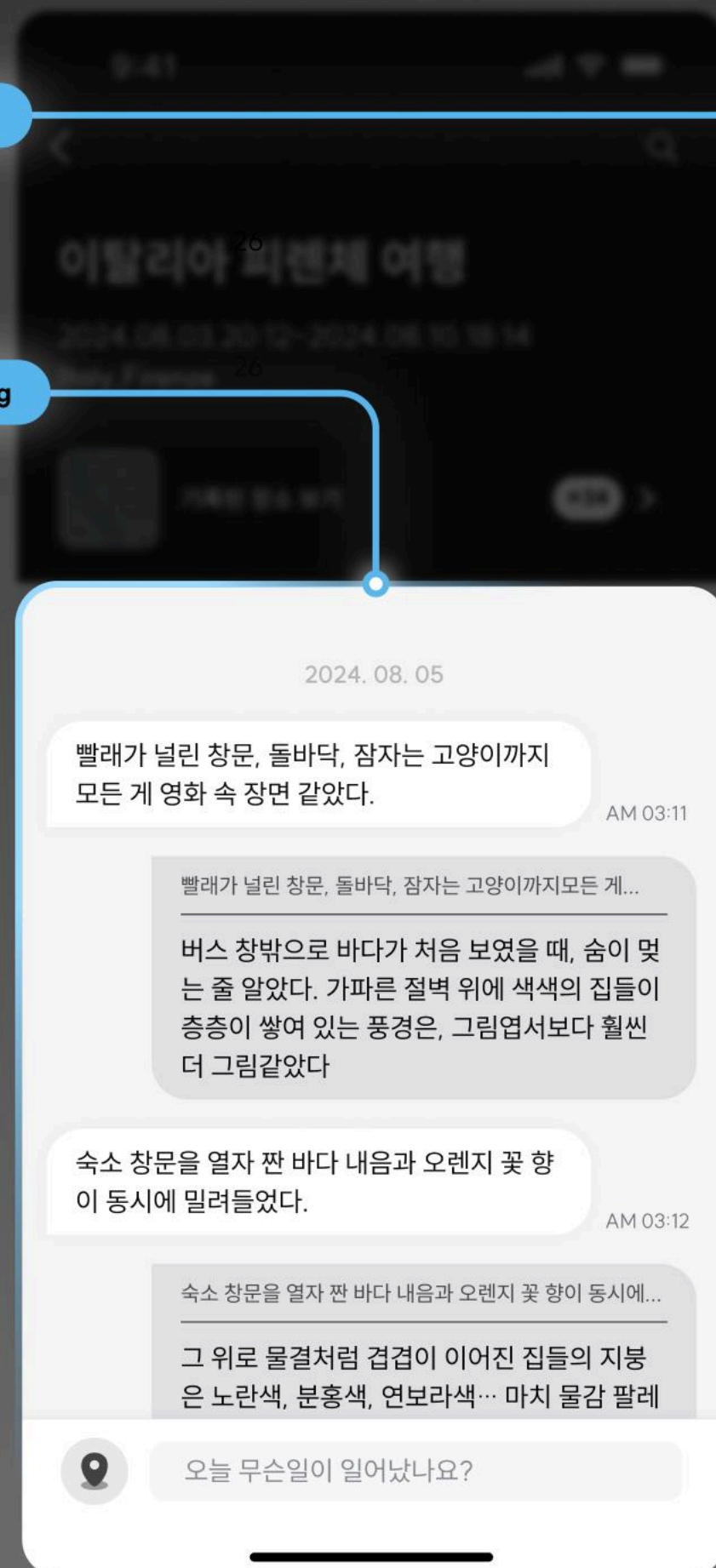
Tool Maze
Participants 16명
Period 1Week / 10.12 - 10.19
Research Method Survey

B시안의 Chatting과, D시안의 Header를 채택하여 다음과 같이 서비스를 디자인했습니다.



D 시안의 Header

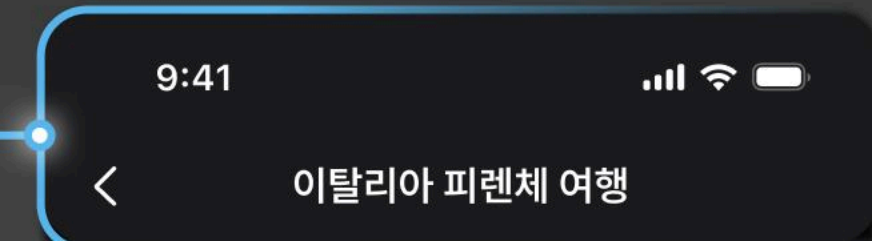
B 시안의 Chatting



VOC about Chatting

다른 형식은 구분이 명료하게 되어있지 않아요.

아무래도 이게 제일 익숙해서 기록하기도 편할 것 같아요.



VOC about Header

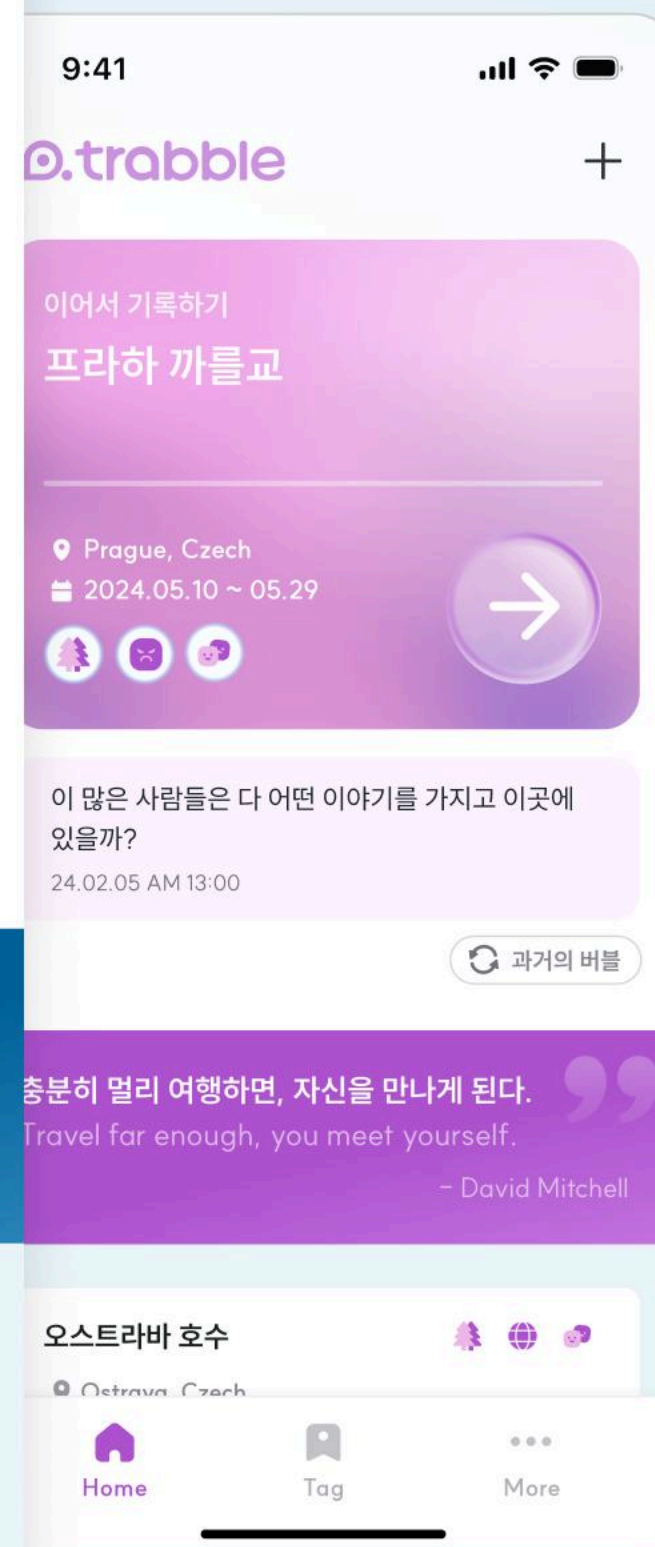
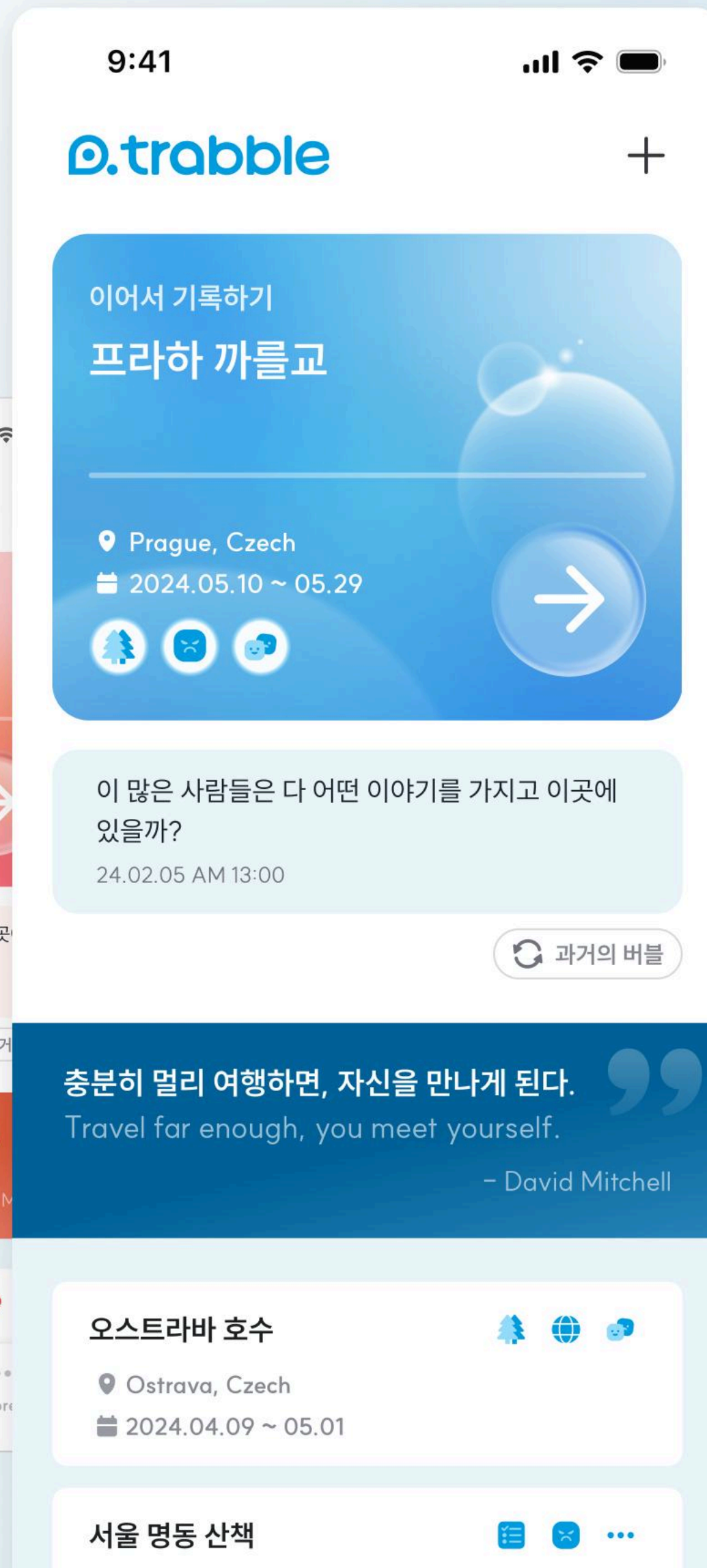
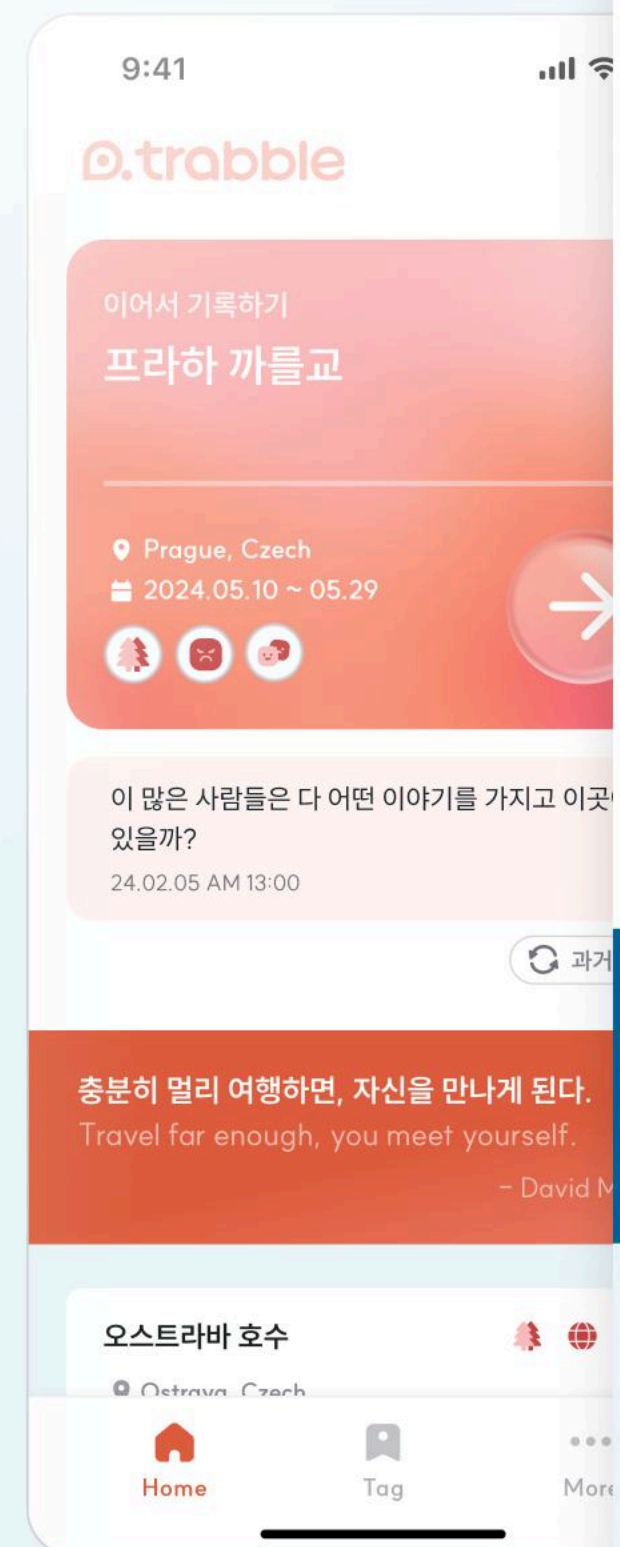
헤더 부분이 많은 정보가 있을 필요가 없다고 생각해요.

위 헤더 부분이 크면 기록하기에 불편할 것 같아요.

Trabble은 이런 서비스예요!

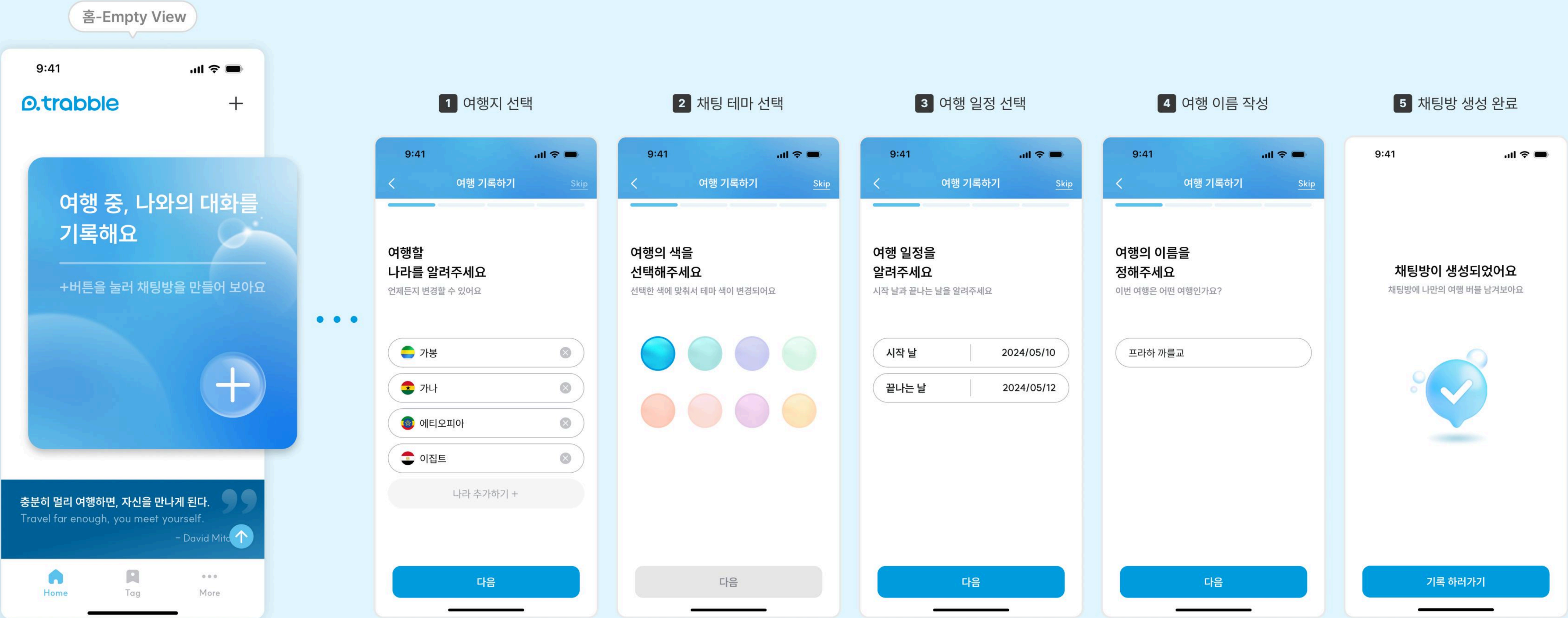
여행 중 채팅 형식으로 여행을 기록하며
Trabble을 남겨요.

Trabble 메인 홈



Trabble은 이런 서비스예요!

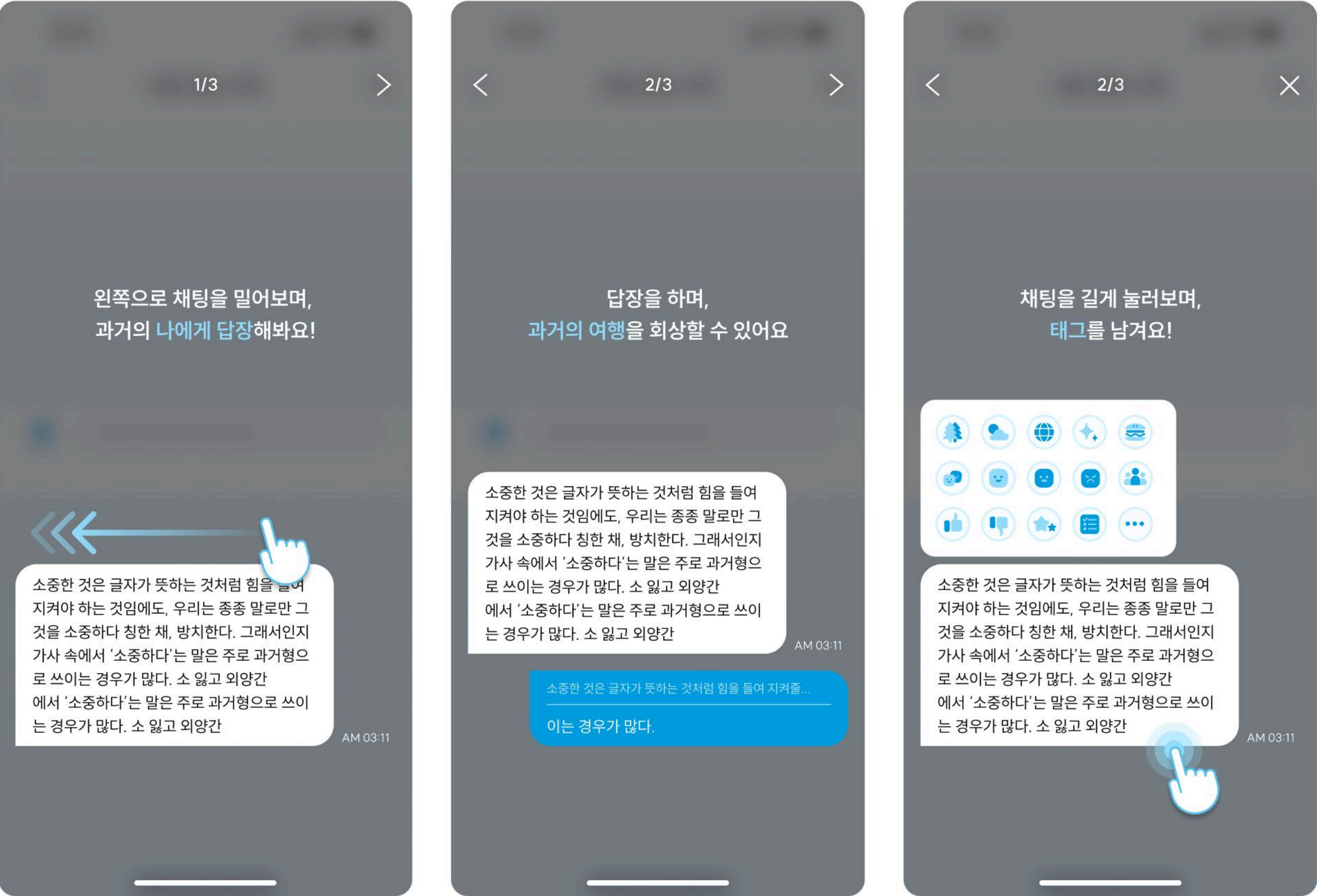
채팅방을 만들어 나와의 대화를 시작합니다.



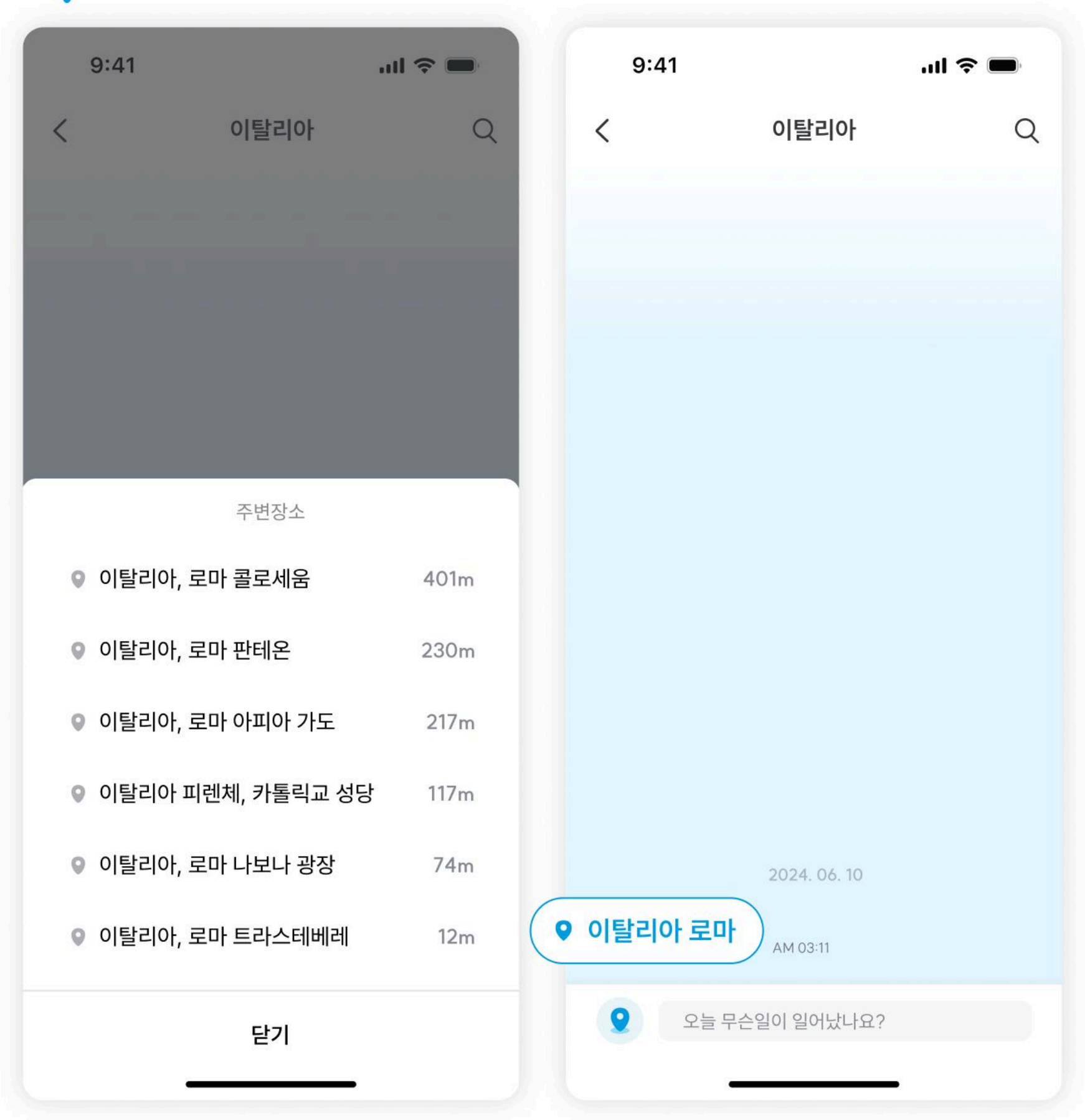
Trabble은 이런 서비스예요!

채팅 시작 전 튜토리얼을 보고 사용 방법을 익힙니다.
GPS로, 내가 여행 중인 장소를 기록할 수 있습니다.

튜토리얼

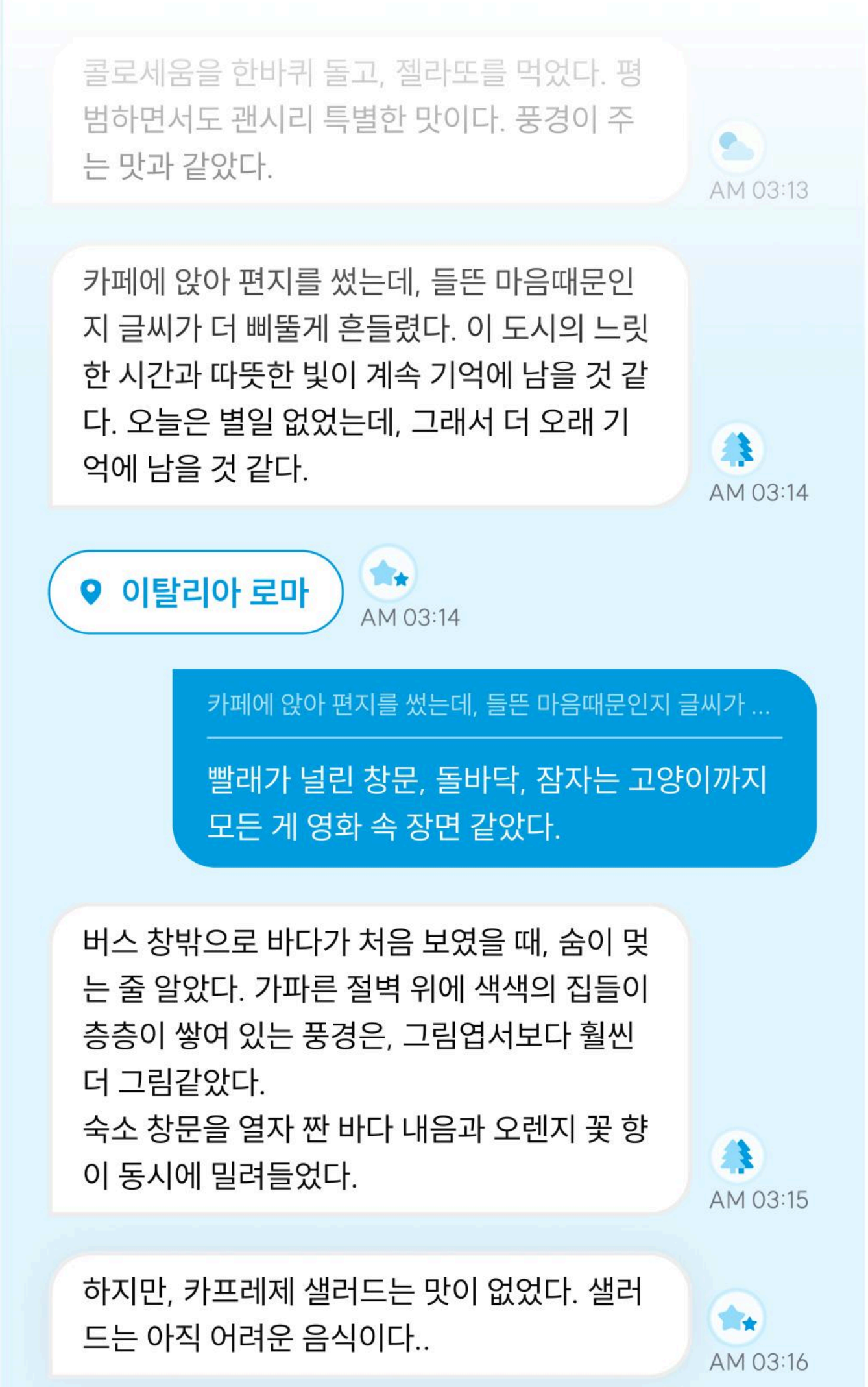


장소추가

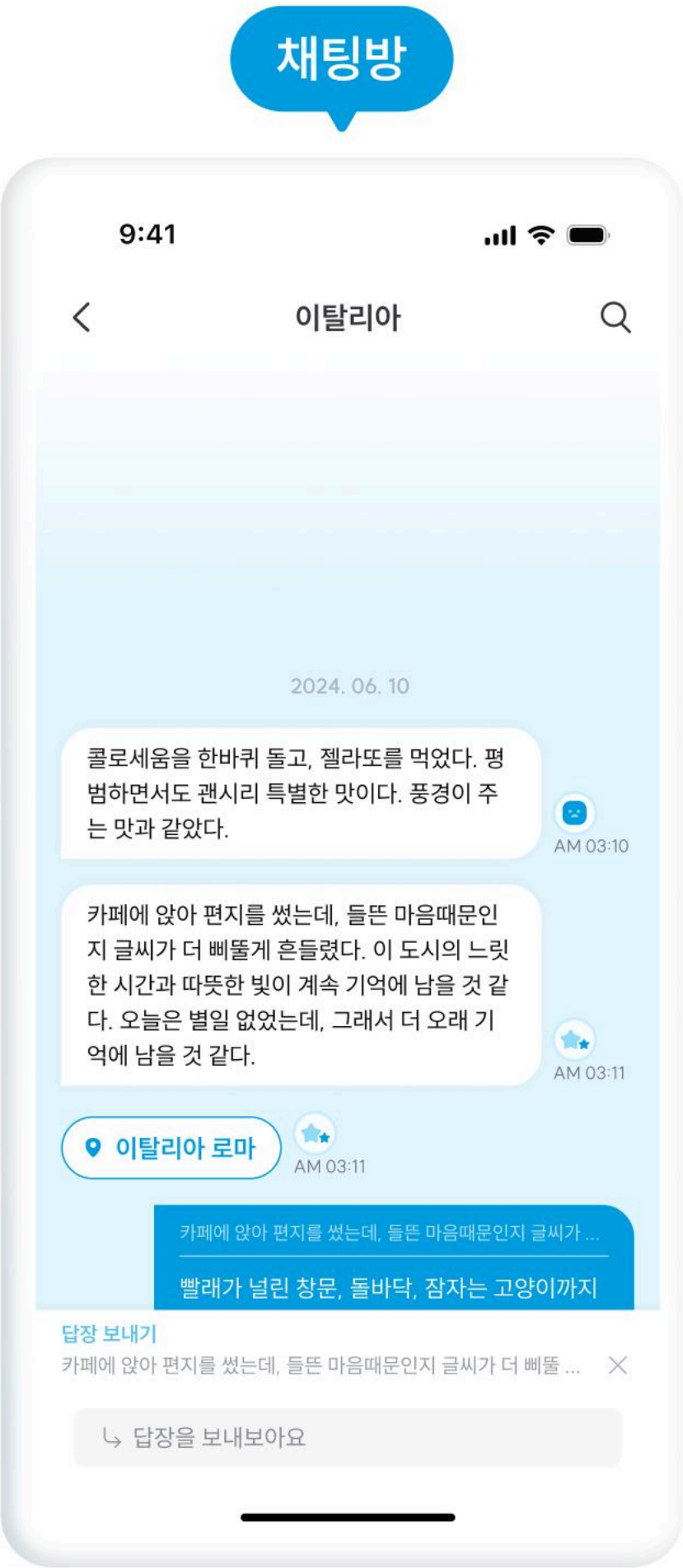


Trabble은 이런 서비스예요!

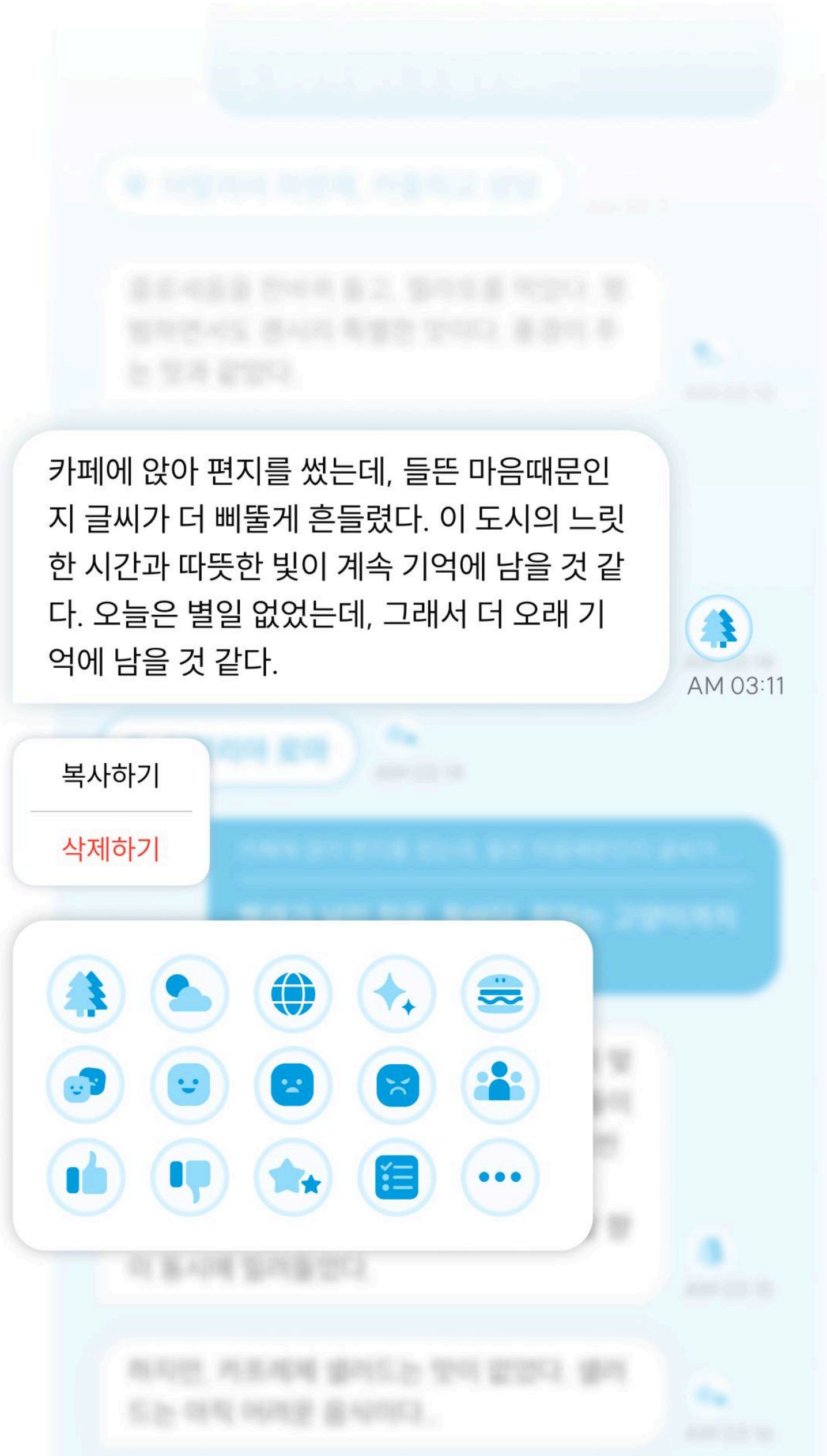
여행 중, 순간의 감정을 채팅으로 기록해요.
채팅마다 태그를 달아 기록을 더 구체적으로 구분할 수 있어요.



Zoom in



Press



Trabble은 이런 서비스예요!

태그별로 정리된 기록을 보며, 여행 중의 나를 되돌아봐요.

