

Portfolio

정지우 | Frontend Developer

 [개발 블로그 Link >](#)

 [Github Link >](#)

01 경력 & 기술 스택

02 Challengers 프로젝트

03 Myshipdan 프로젝트

04 오픈소스

정지우

현) 화이트큐브 Frontend Developer

경력) 2년 4개월

2022.11 - 현재 (6개월)	화이트큐브 Front End Developer (Full-time)
2021.8 - 2022.10 (1년 2개월)	벙커키즈 Front End Developer (Full-time)
2021.2 - 2021.8 (6개월)	벙커키즈 React Native Developer (Intern)
2016.3 - 2021.2	송실대학교 소프트웨어 학사

기술 스택

Frontend

★★★★★

React, Next.js, React Native, Recoil, React-Query, Redux, styled-component

★★★

Remix.js, Firebase, Supabase, Swift

Language

★★★★★

Typescript, Javascript

★★★

Objective-C, SwiftUI, Python

More

★★★

Github Action, AWS (Elastic Beanstalk, S3, Route53), Docker, Selenium Scraper

건강 습관 앱, **챌린저스**



누적 가입자 수 158만

최고 MAU 109만

누적 거래액 3300억+

- 01 디자인 시스템 배포 자동화
- 02 React Native iOS 위젯
- 03 혜택 만보기 광고 UX 개선
- 04 VOC 소리함 제안 및 개발
- 05 코어 모듈 웹 배포 시 겪은 NPM private 패키지 이슈
- 06 프레임워크 버전에 의존성을 가지는 버그의 근본적 원인 해결

 [앱스토어 Link >](#)

 [플레이스토어 Link >](#)

 [웹 Link >](#)

01



디자인 시스템 NPM 패키지 배포 자동화를 통한 개발 업무 효율화

[블로그 Link >](#)

Auto Publish - minor update				
auto_publish.yml				
201 workflow runs				
		Event	Status	Branch
✔ feat: change snack bar elevation upper than pop up		Auto Publish - minor update #201: Commit 65d6d7d pushed by jiwoolIncludeJeong	develop	3 days ago 1m 54s
✔ feat: add onCountEnd props at SquareBannerPopUp		Auto Publish - minor update #200: Commit f0b43da pushed by jiwoolIncludeJeong	develop	3 days ago 1m 39s
✔ chore: add space		Auto Publish - minor update #199: Commit cb3d461 pushed by jiwoolIncludeJeong	develop	3 days ago 1m 53s
✔ feat: add benefit action		Auto Publish - minor update #198: Commit 0b63aea pushed by jiwoolIncludeJeong	develop	3 days ago 2m 2s
✔ feat: add border and resolve handler at SquareBannerPopUp		Auto Publish - minor update #197: Commit e89bd37 pushed by jiwoolIncludeJeong	develop	3 days ago 1m 39s
✔ fix: add void return type at onPressContent		Auto Publish - minor update #196: Commit 96a93e1 pushed by jiwoolIncludeJeong	develop	3 days ago 1m 44s
✔ chore: remove unused import		Auto Publish - minor update #195: Commit aa5e3fa pushed by jiwoolIncludeJeong	develop	3 days ago 1m 42s
⚠ fix: resolve path alias err		Auto Publish - minor update #194: Commit 6285df8 pushed by jiwoolIncludeJeong	develop	3 days ago 37s

# chlngers-design-system-deploy	
의 결과는 success 예요	
v1.17.80	
fix: resolve pop up banner issue	
CDS-deploy-machine	
버전 v1.17.81	
의 결과는 failure 예요	
버전 v1.17.81	
의 결과는 success 예요	
v1.17.81	
chore: remove unused import	
CDS-deploy-machine	
버전 v1.17.82	
의 결과는 success 예요	
v1.17.82	
fix: add void return type at onPressContent	
CDS-deploy-machine	
버전 v1.17.83	
의 결과는 success 예요	
v1.17.83	
feat: add border and resolve handler at SquareBannerPopUp	
CDS-deploy-machine	
버전 v1.17.84	
의 결과는 success 예요	

문제/배경

디자인 시스템을 수정하고 앱에 적용하는 절차가 복잡하여
개발 업무 효율이 떨어짐

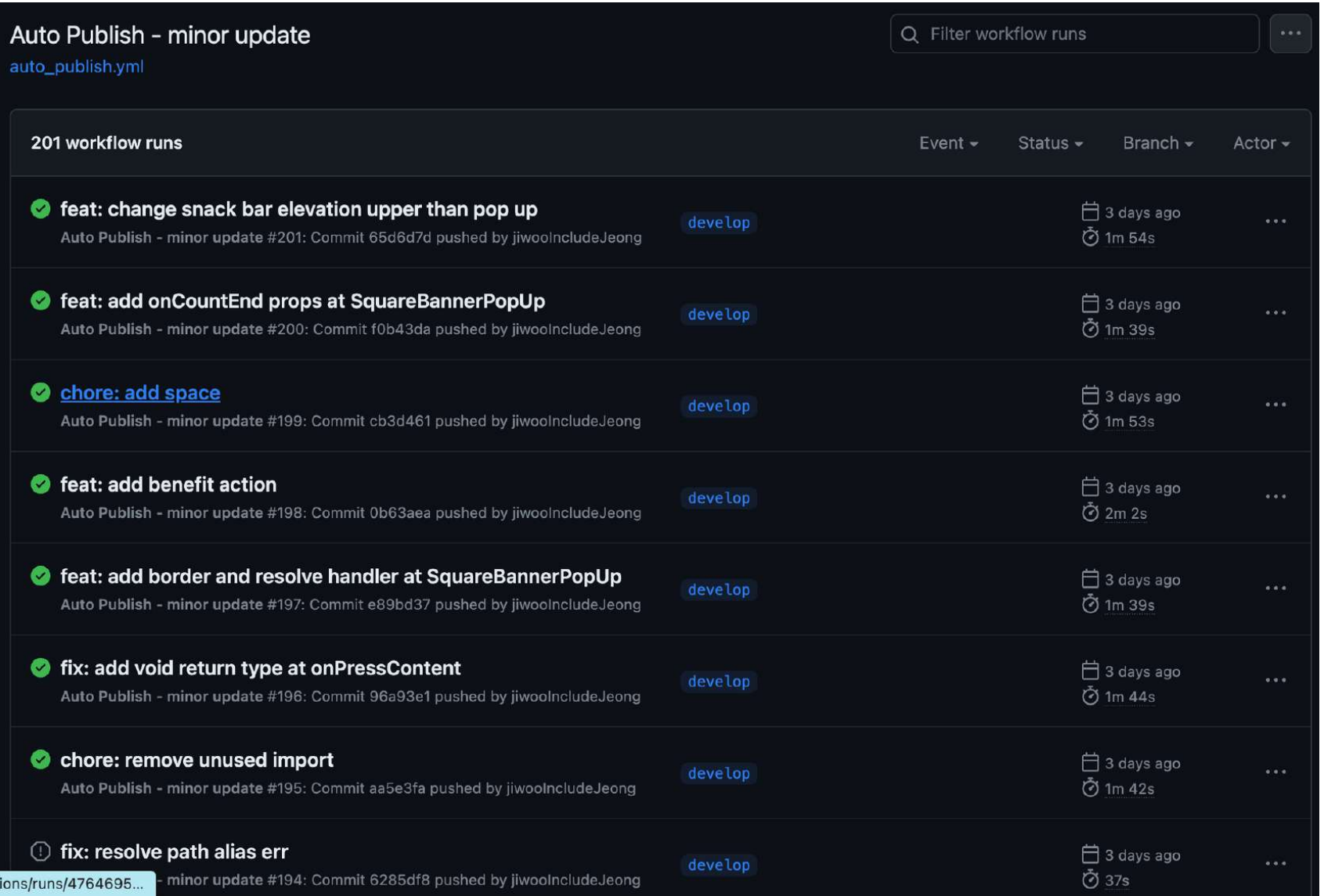
- 디자인 시스템을 앱에 적용하는 복잡한 과정 -

- 1. 디자인 시스템 코드 수정
- 2. commit
- 3. Pull Request
- 4. 마스터 브랜치 merge
- 5. npm 패키지 수동 배포
- 6. 챌린저스 앱 package.json에 버전 업데이트
- 7. 패키지 설치
- 8. 드디어 적용 완료 🍷

해결

“반복 과정을 자동화하자!”

- 01 Github Action을 통해 디자인시스템 저장소의 마스터 브랜치에 commit이 push되면 패치 버전을 올려 자동으로 배포
- 02 버전 입력이 가능한 수동 배포 Github Action도 추가
- 03 슬랙 웹훅으로 결과 자동 알림



실제 동작하고 있는 자동 배포 Github Action



슬랙 웹훅

회고

Keep

- 스스로 일의 범위를 확장하여 개발 워크플로 효율화 → 팀의 생산성 향상에 기여

Problem

- mono repo로 구성했다면, 매번 디자인 시스템을 배포하는 과정이 없었을 것
- 초기 프로젝트 설계에 참여했다면 mono repo 의견을 제안했을 것

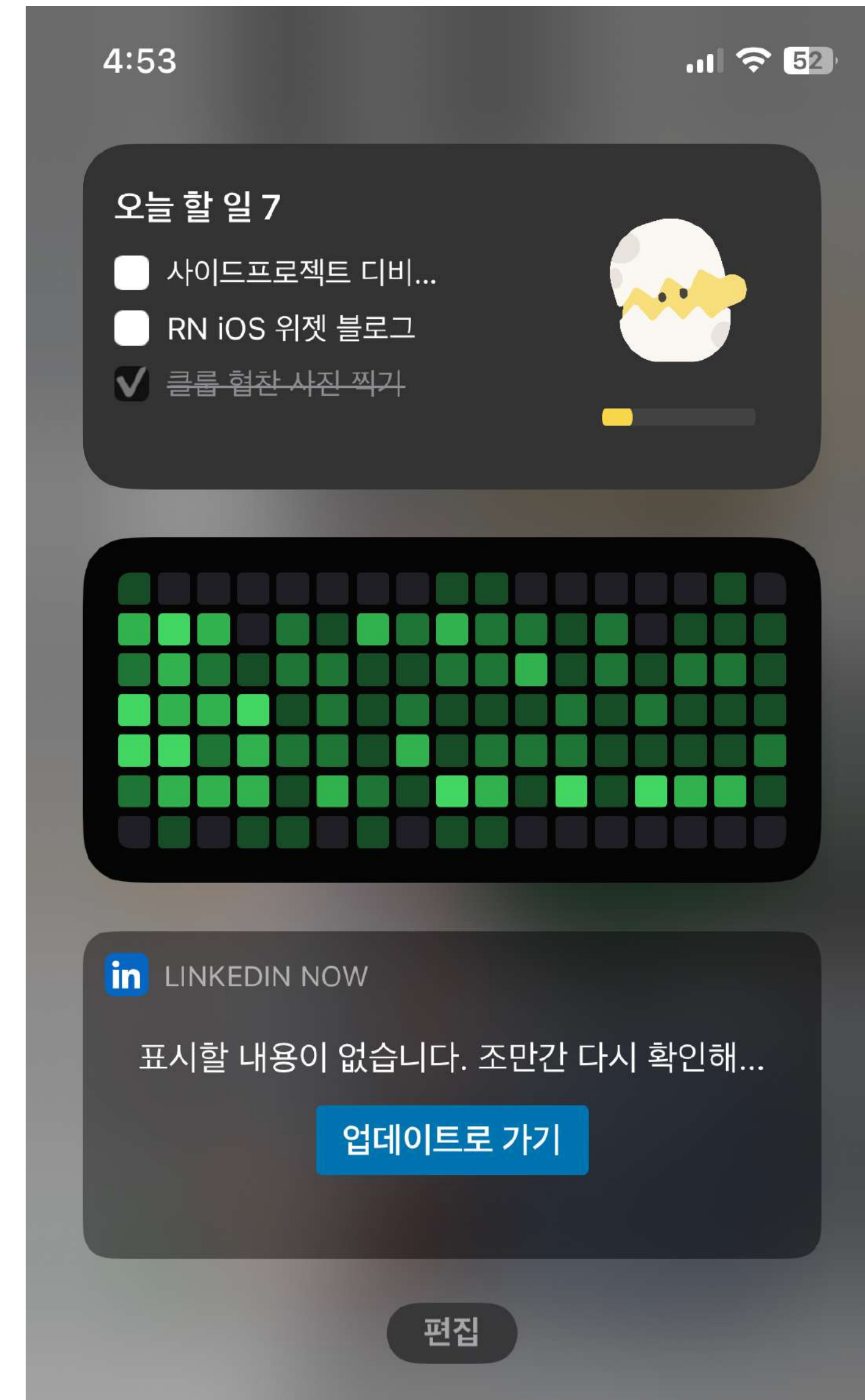
Try

- Github Action 캐싱을 통해 node 환경 세팅과 node_modules를 매 Action마다 설치하는 과정을 생략하여 Action 실행 속도 개선

React Native iOS 위젯 개발을 통한 갓생몬 메인스크린 기준 리텐션 8%p 상승

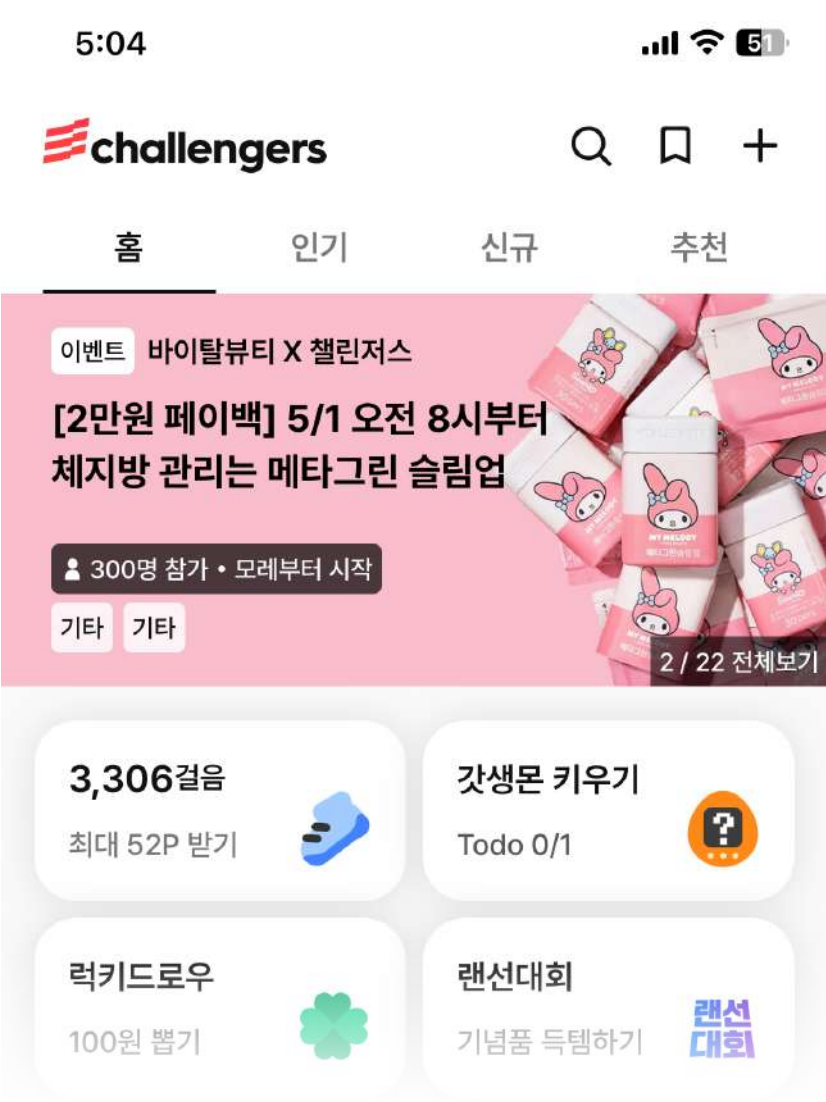
[블로그 Link >](#)

[Github Link >](#)

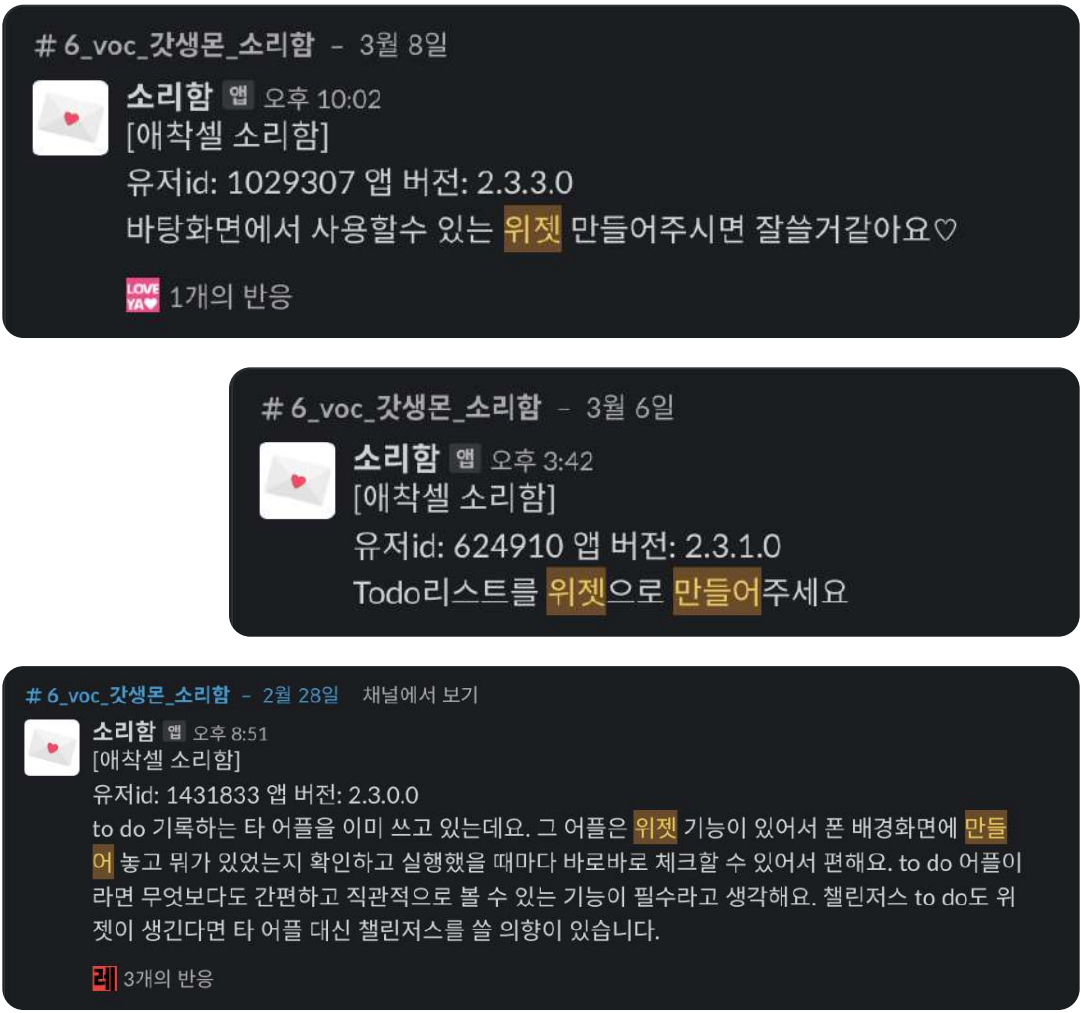


문제/배경

- Asis: 갓생몬 페이지는 ‘챌린저스 홈 탭 → 갓생몬 키우기’를 통해서만 진입 가능
- VOC: 사용자는 1 Depth 깊은 진입점에 불편함을 느낌



Asis 갓생몬 진입경로

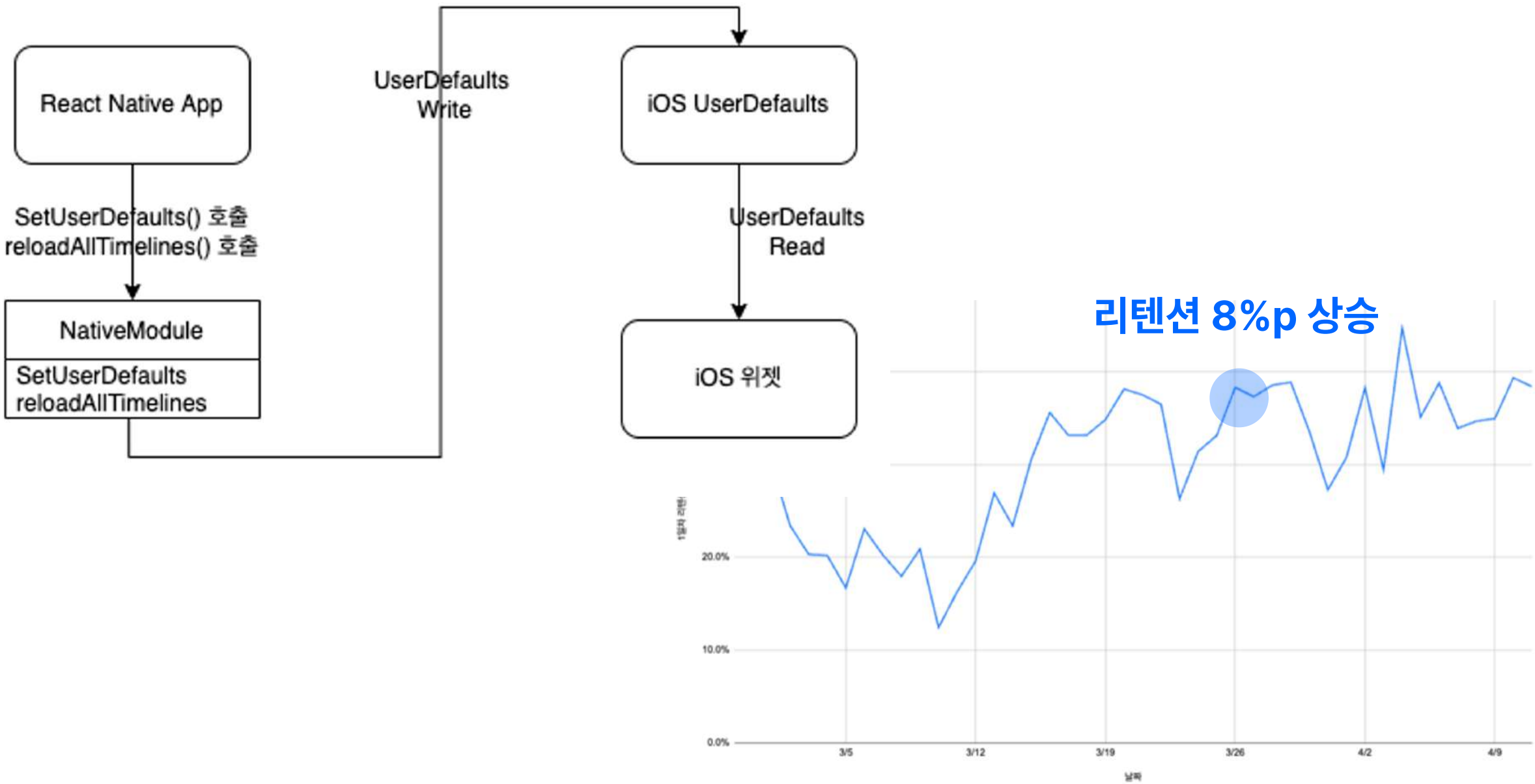


위젯을 원하는 VOC

해결

“다이렉트 진입점(위젯)을 만들어 갓생몬 메인스크린 리텐션을 상승시키자!”

- 01 iOS의 공유 저장소로 사용하는 UserDefaults에 데이터를 Write하는 React Native의 Native Module 개발
- 02 위젯에서 UserDefaults 데이터를 Read하여 보여줌
- 03 갓생몬 메인 스크린에 들어갈 때마다 UserDefaults에 데이터 Write하여 위젯 데이터를 새로고침



회고

Keep

- 경험이 없던 iOS 개발을 시도해 봄으로써 위젯의 동작원리를 이해하여 러닝코드를 만들어 프로덕션 배포
- 개발 도구에 제한을 두지 않고, 동작원리를 파악한 후 어떠한 도구를 통해서든 개발 시도

Problem

- 현재는 reloadAllTimelines()를 통해 위젯의 타임라인을 모두 새로고침
- 추후 여러 개의 위젯이 생겼을 때를 고려하여 reloadTimelines(ofKind: String)으로 개발했다면 확장성이 있을 것

Try

- 확장성을 고려하여 reloadTimelines(ofKind: String)을 사용하도록 리팩토링
- 백그라운드에서 React Native의 UserDefaults 데이터 Write 함수를 호출하여 자정마다 위젯 데이터 새로고침

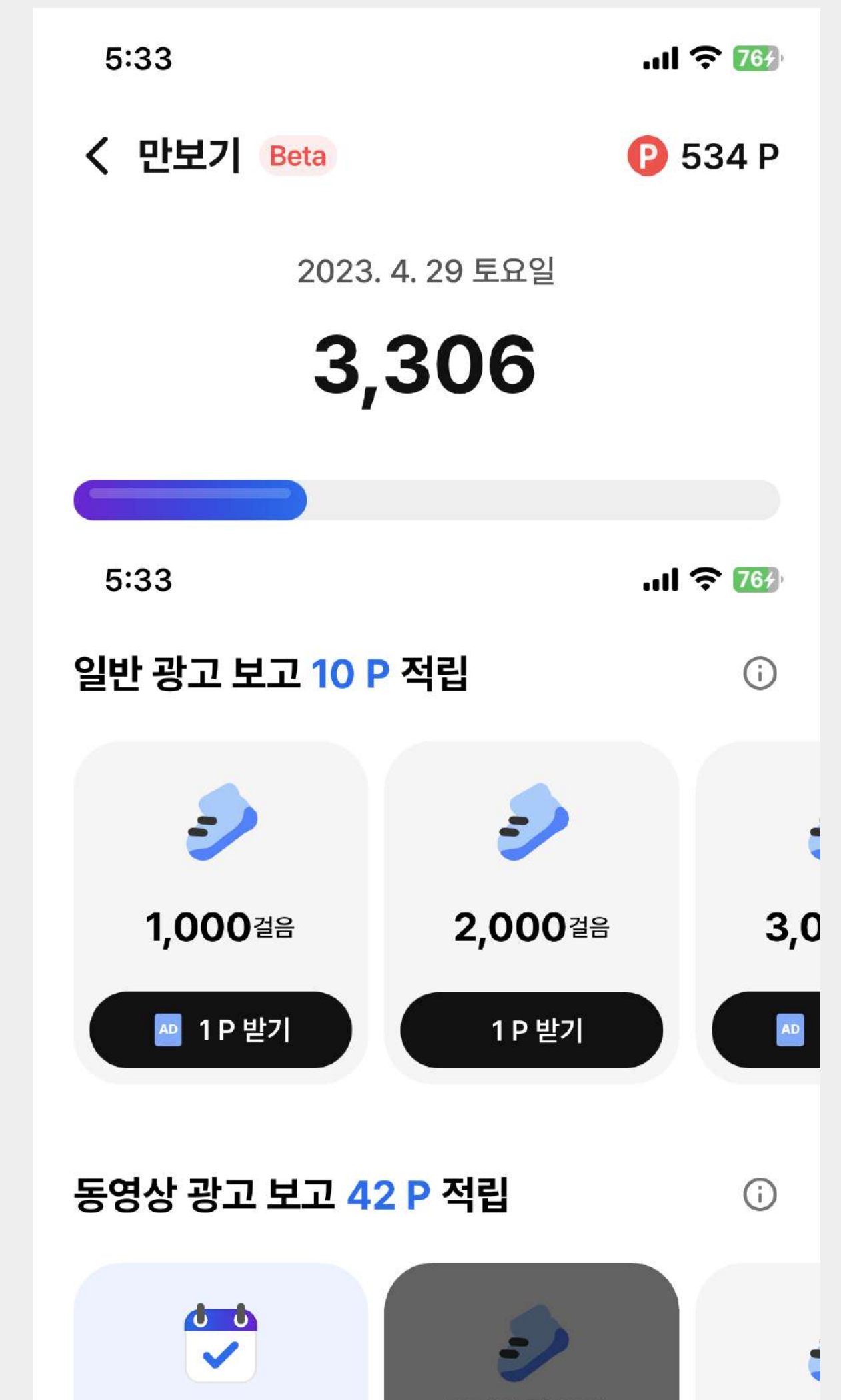
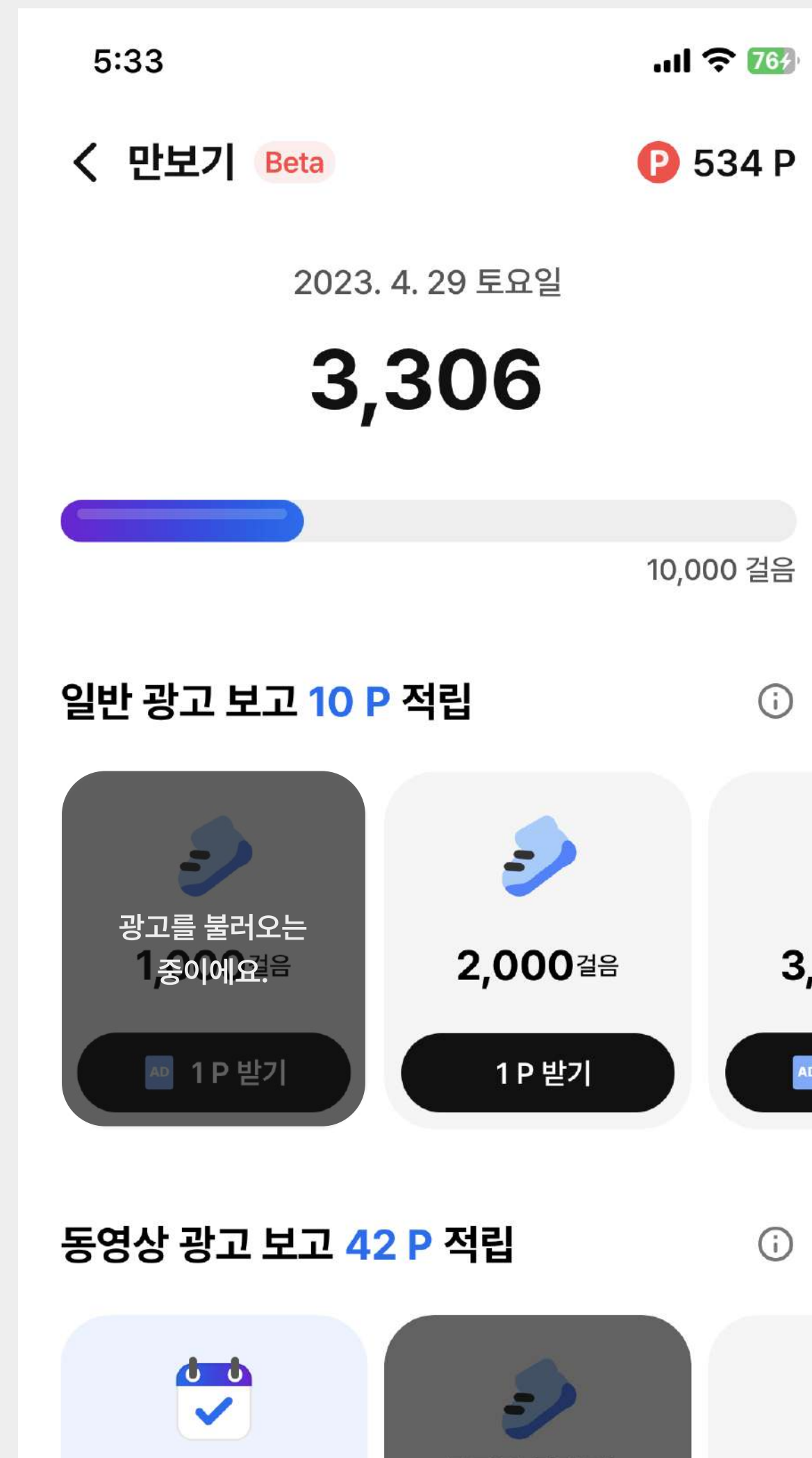
03

혜택 만보기 광고 UX 개선을 통해 1만+ DAU 회복, VOC 개선



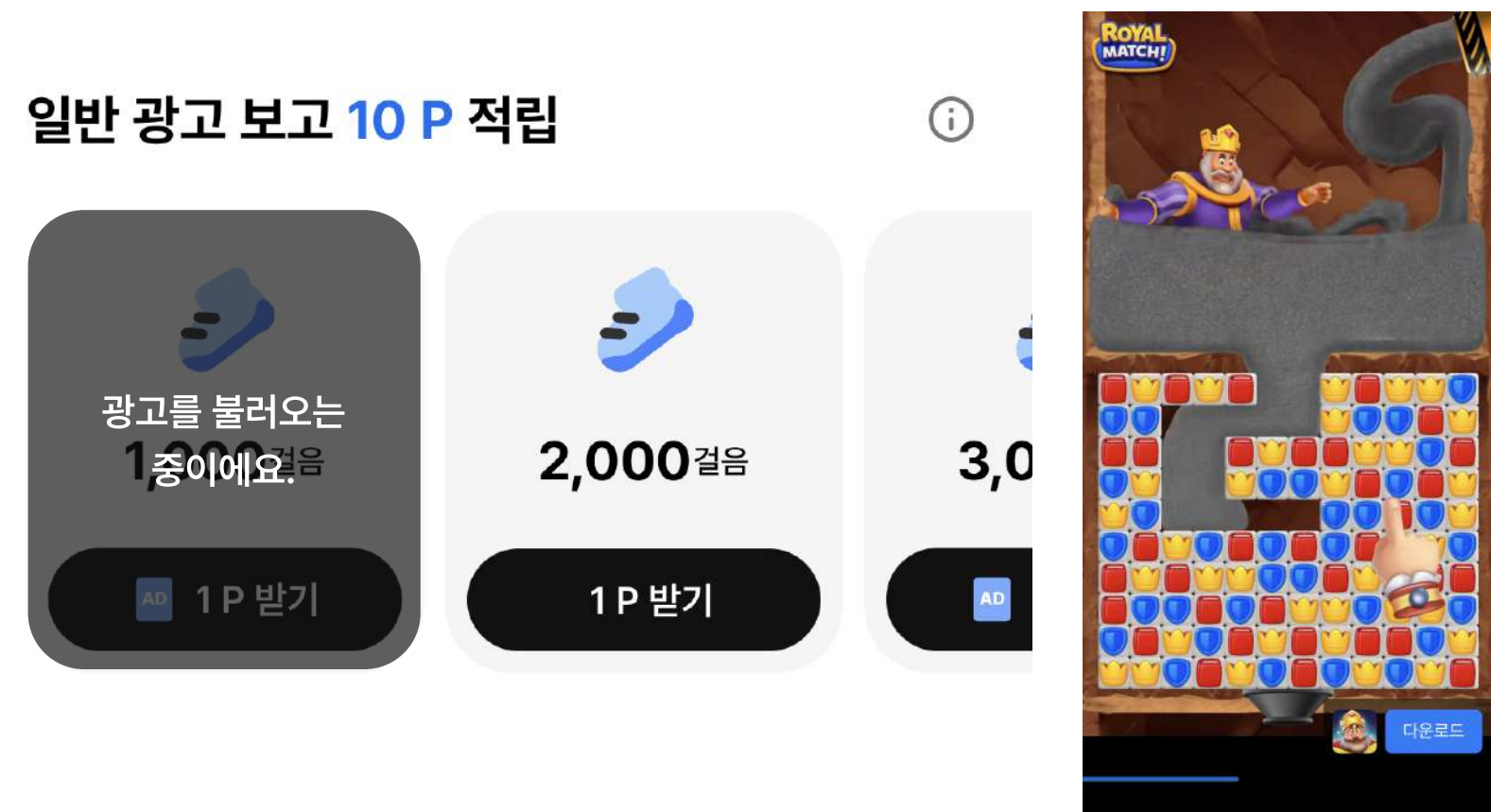
ASIS

TOBE



문제/배경

- Asis: 혜택 만보기 동작 방식
 - 일정 걸음 수 달성 → ADX 제공 광고 로딩 → 포인트 받기
- ADX 제공 광고 로딩이 유저 별로 최대 30분까지 소요
- 앱 평점 4.6에서 3.9로 하락(구글플레이스토어 기준)

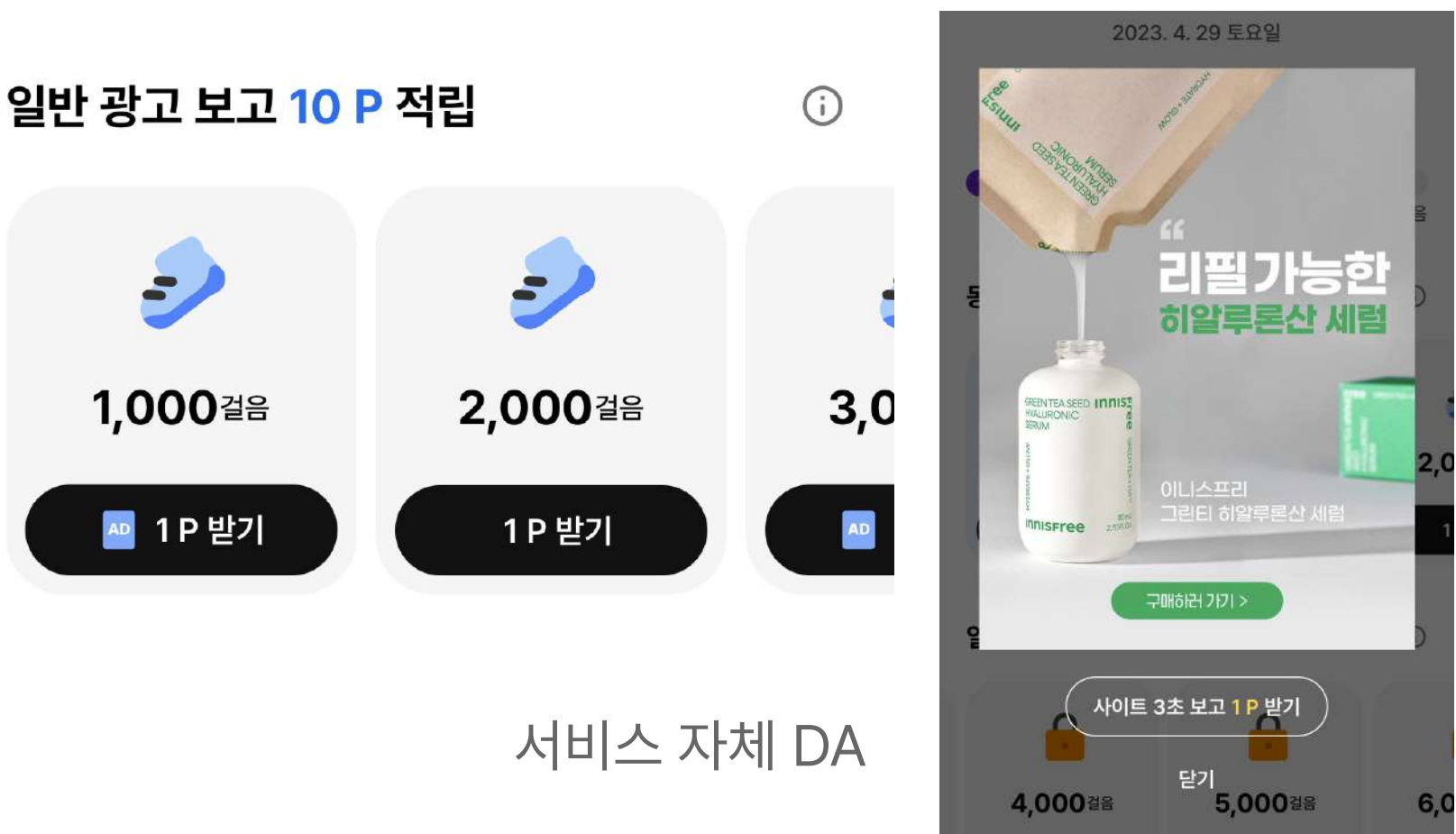


광고가 불러오기까지 30분을 기다리고, 유저가 보게되는 광고

해결

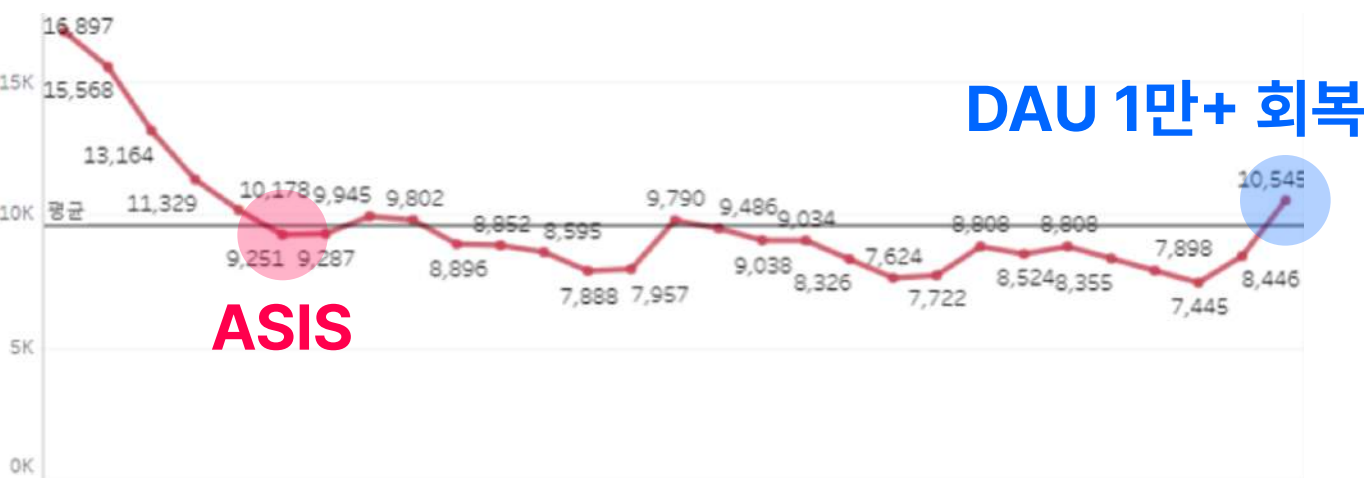
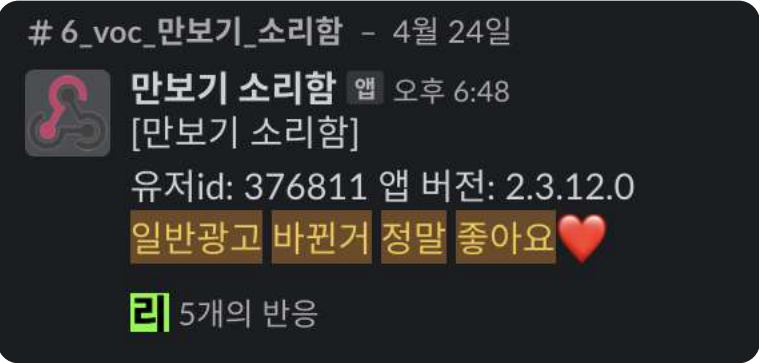
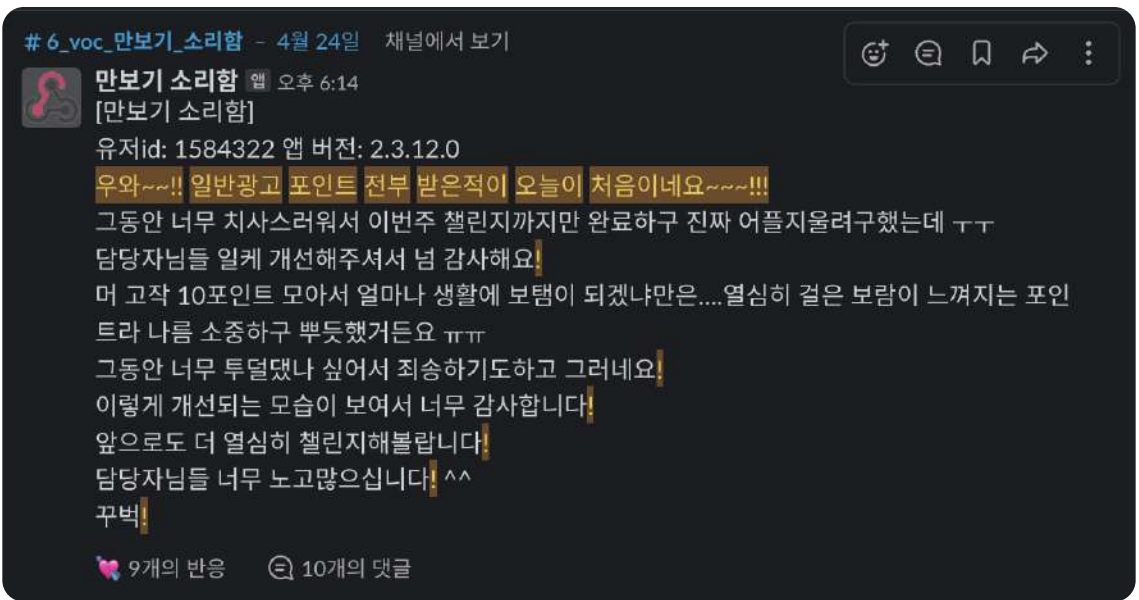
“서비스 자체 DA(Display Ad) 상품을 추가해서 사용자가 광고를 기다리지 않게 하자!”

- 01 ADX 제공 광고와 서비스 자체 DA를 함께 요청하여, 먼저 요청이 성공한 광고를 보여주는 아이디어 제안
- 02 ADX 제공 광고가 로딩 중이더라도 서비스 자체 DA가 집행 중일 때는 포인트를 받을 수 있도록 개선



서비스 자체 DA

바뀌자마자 긍정적인 VOC 🥳



회고

Keep

- “유저 입장에서의 불편함을 해소하고 사업적으로도 수익 구조를 만들 수 있을 것 같다”는 가설을 가진 아이디어 검증
- 외부 패키지(expo-web-browser)에 의존성을 가진 코드를 리팩토링을 통하여 의존성이 없도록 직접 구현

Problem

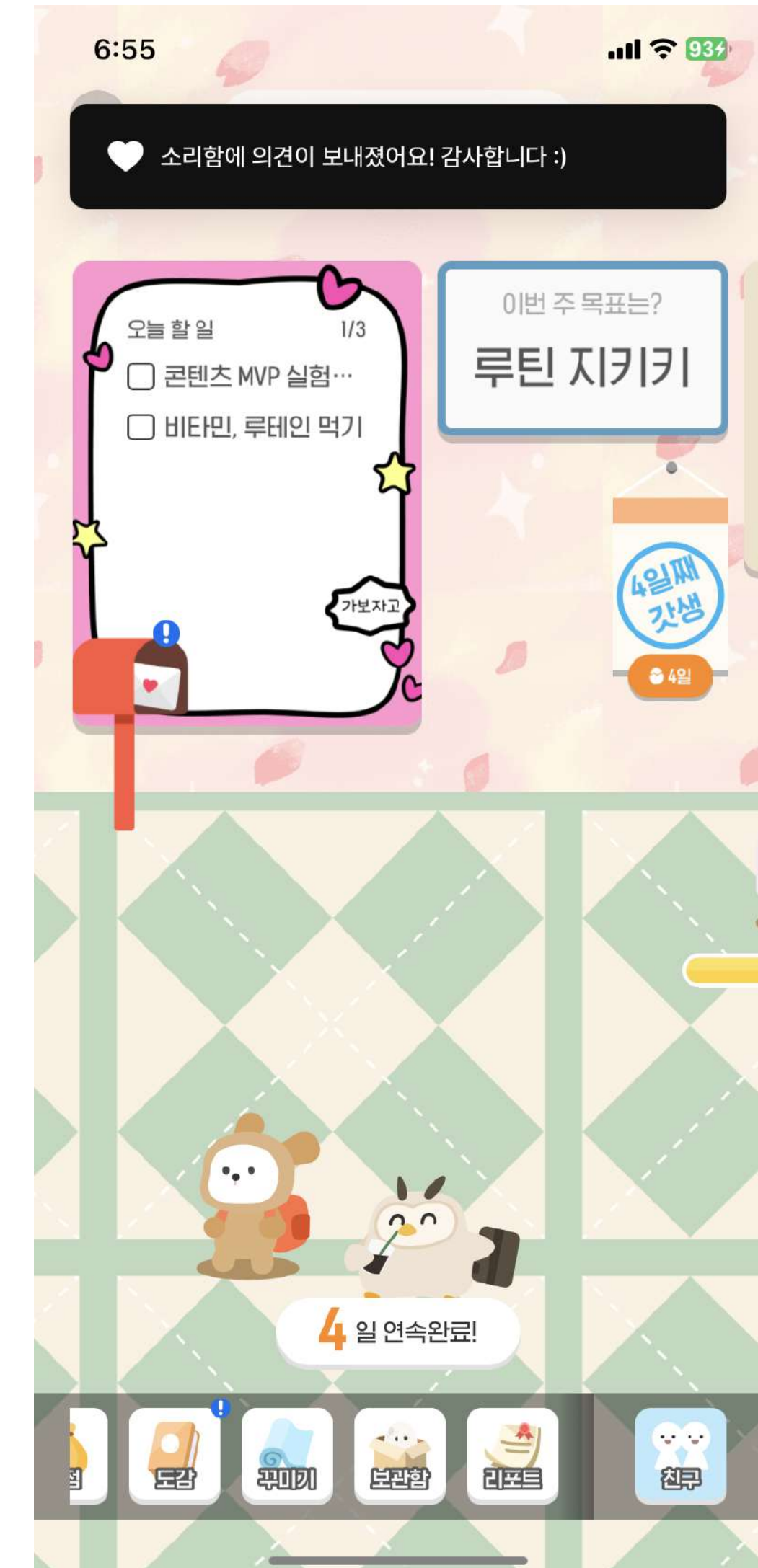
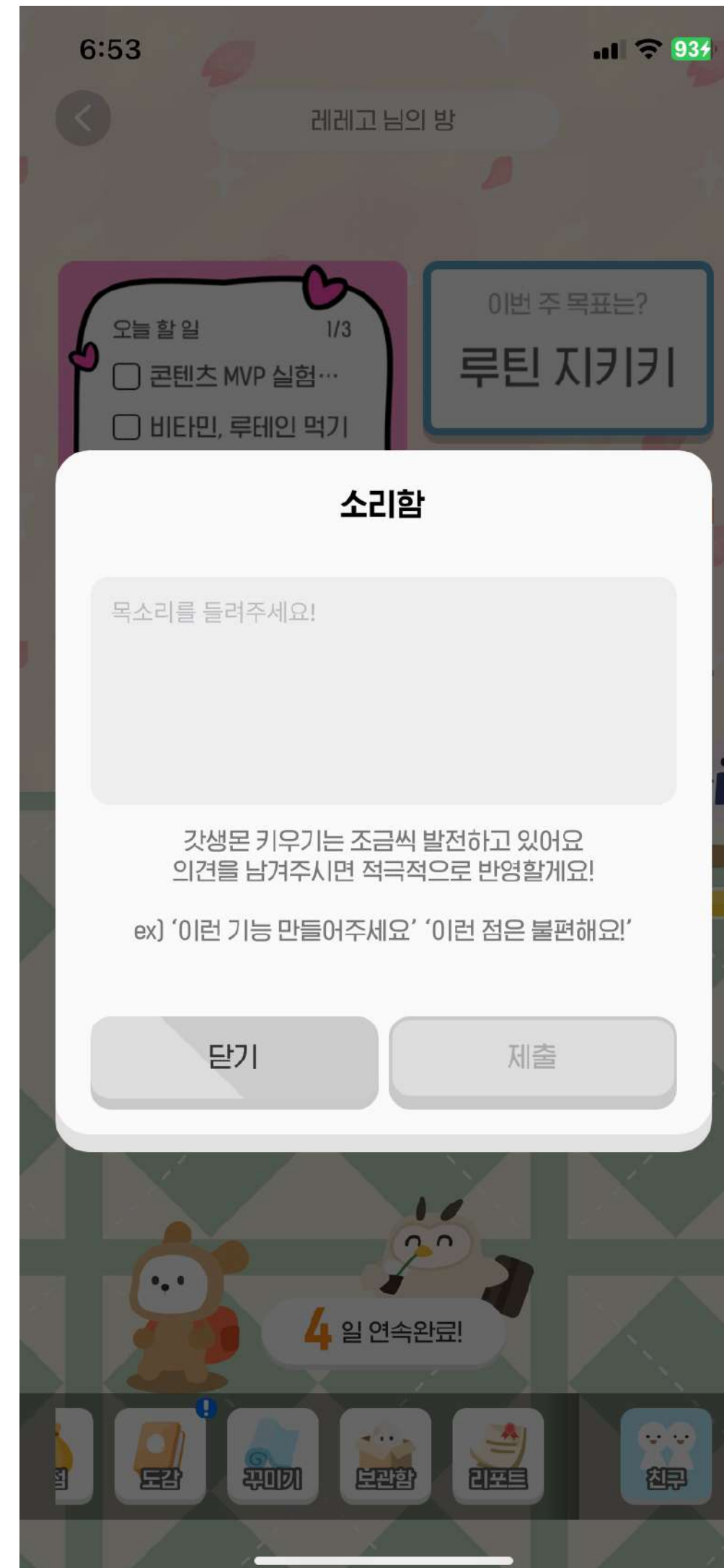
- 서비스 자체 DA 집행이 종료되면 다시 부정적인 결과의 DAU와 VOC가 등장할 우려

Try

- 자체 DA가 미집행 중일 때, ADX 광고와 함께 여러 광고도 요청하여 광고를 기다리는 유저 경험 개선

04

유저 니즈를 프로덕트에 빠르게 반영하기 위한 VOC 소리함 제안 및 개발



문제/배경

- Asis: 갓생몬 피쳐 개발의 기존 일정은 한 달 개발 후 정식 배포
- 유저의 니즈를 파악하는데까지 한 달이 소요될 것으로 예상
- 빠르게 유저의 니즈를 반영해나가며 피쳐를 고도화 하고 싶은 팀의 의견

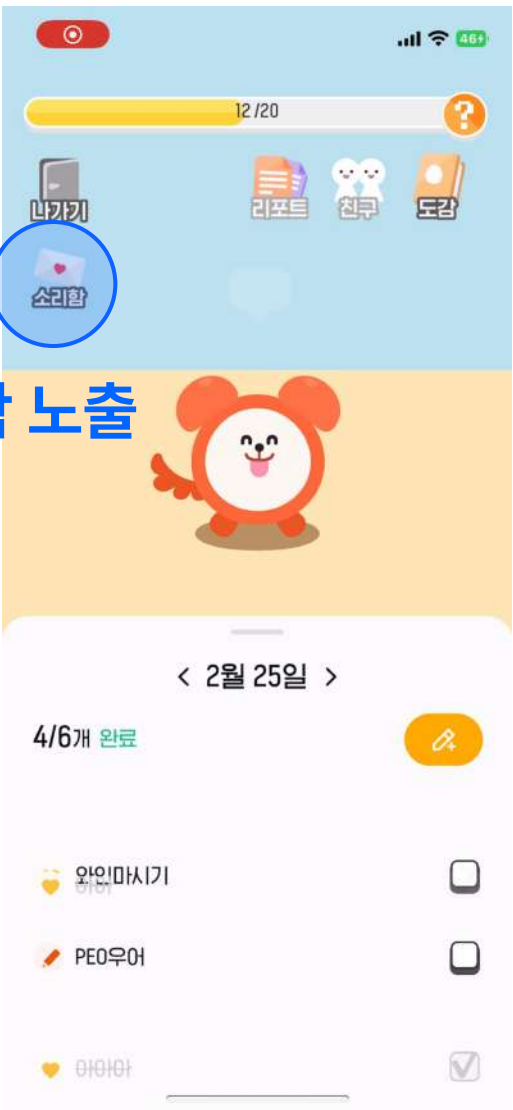
해결

“매주 배포하자!”

“정성적 VOC를 받을 수 있는 창구를 만들어서
유저 니즈를 빠르게 반영하자!!”

- 01 VOC를 자유롭게 작성할 수 있는 소리함을 갓생몬 메인 스크린에 노출해 정성적 VOC 획득
- 02 VOC를 카테고리화하여 기능 개발 우선순위를 유연하게 변경

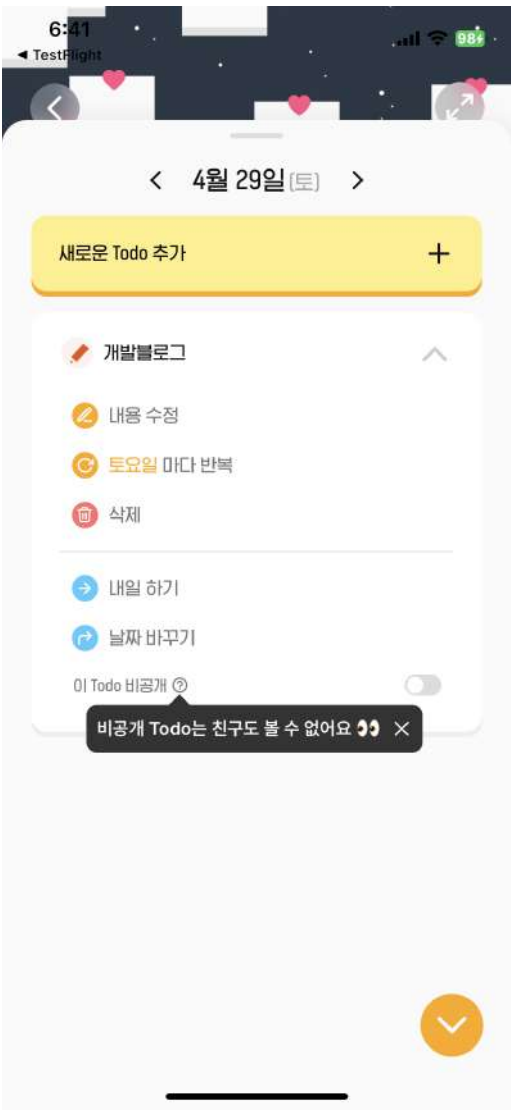
메인 스크린에 소리함 노출



1주차 베타버전



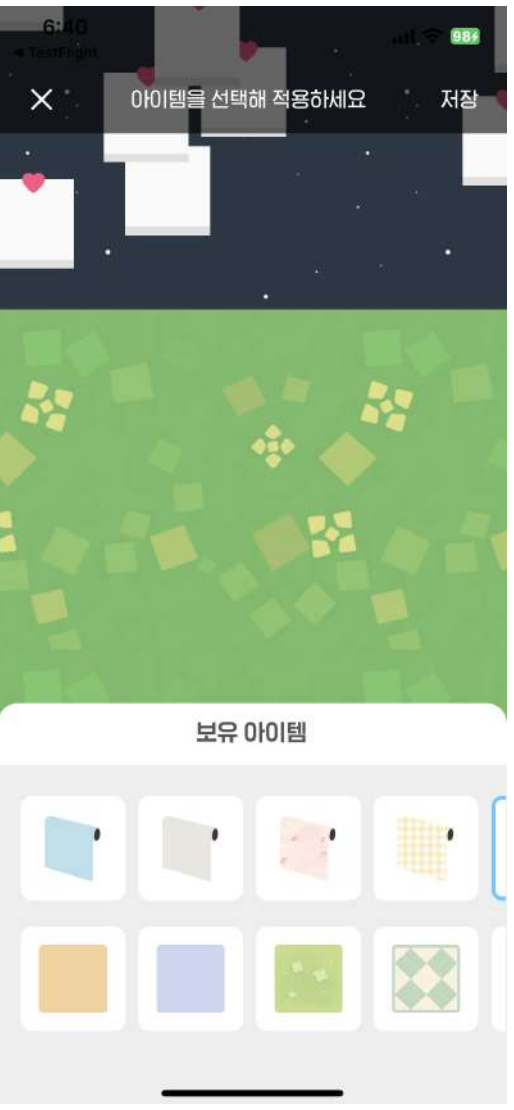
2주차 베타버전



3주차 베타버전



4주차 베타버전



회고

Keep

- 개발 중인 피쳐에 대해 지속적으로 정성적 VOC 수집
- VOC 카테고리화를 통해 유저의 니즈를 파악하여 우선순위 결정 및 피쳐 방향성 설정
- 기획에 유연하게 대처할 수 있는 확장성 고려한 개발

Problem

- 소리함 의견에 대해 응답할 수 있었다면 개선된 사항에 대한 업데이트 등 액션 유도 가능

Try

- 앱 푸시를 활용한 업데이트 등 유저 액션 유도

05

코어 모듈 웹 배포 시 겪은

NPM private 패키지 이슈 해결

 [블로그 Link >](#)

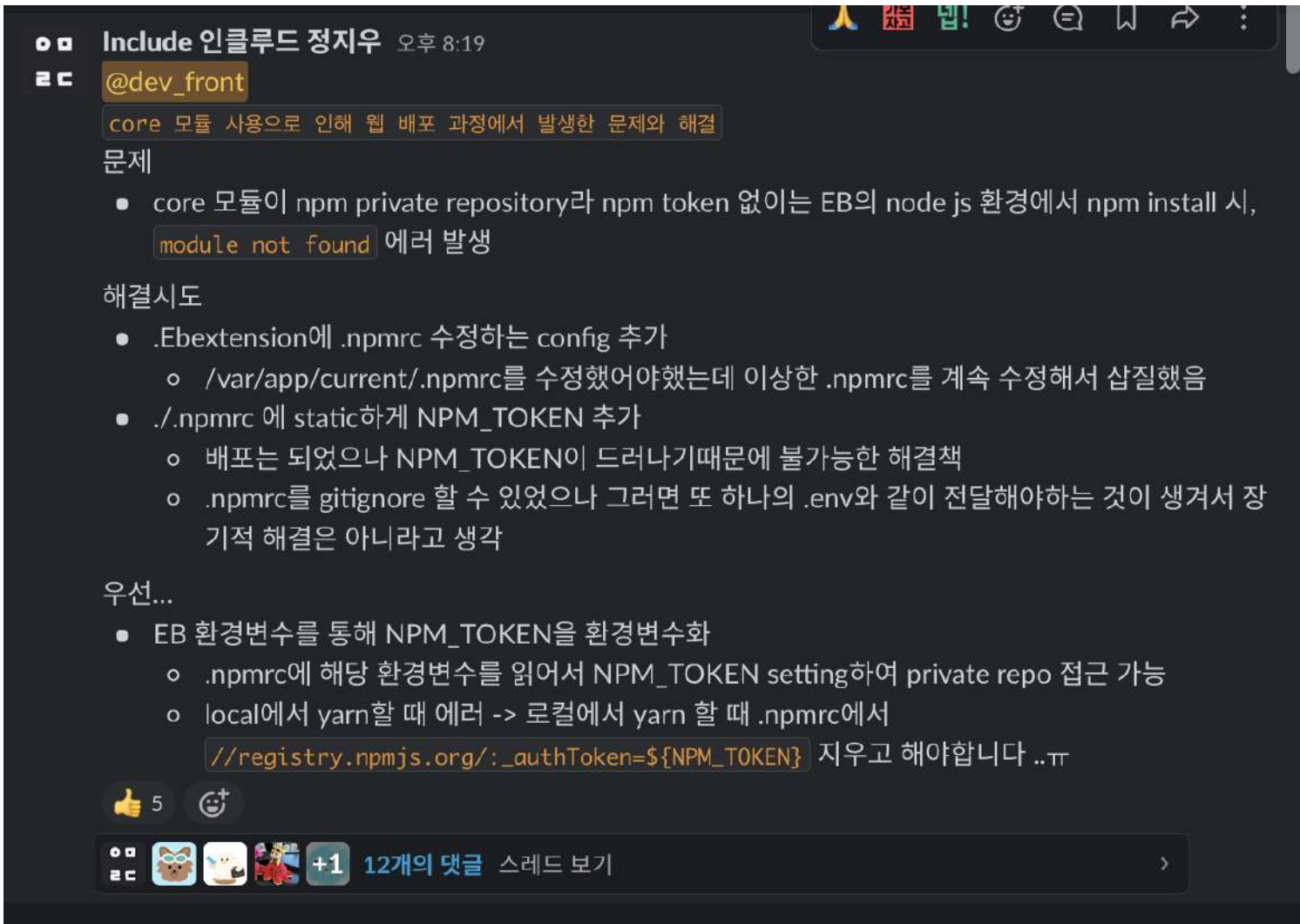
문제/배경

- 코어 모듈 NPM Private 패키지를 웹에 설치한 후, AWS Elastic Beanstalk 배포 과정에서 에러 발생
- AWS Elastic Beanstalk에서 코어 모듈에 접근 가능한 NPM 토큰이 설정되어있지 않은 이슈 파악

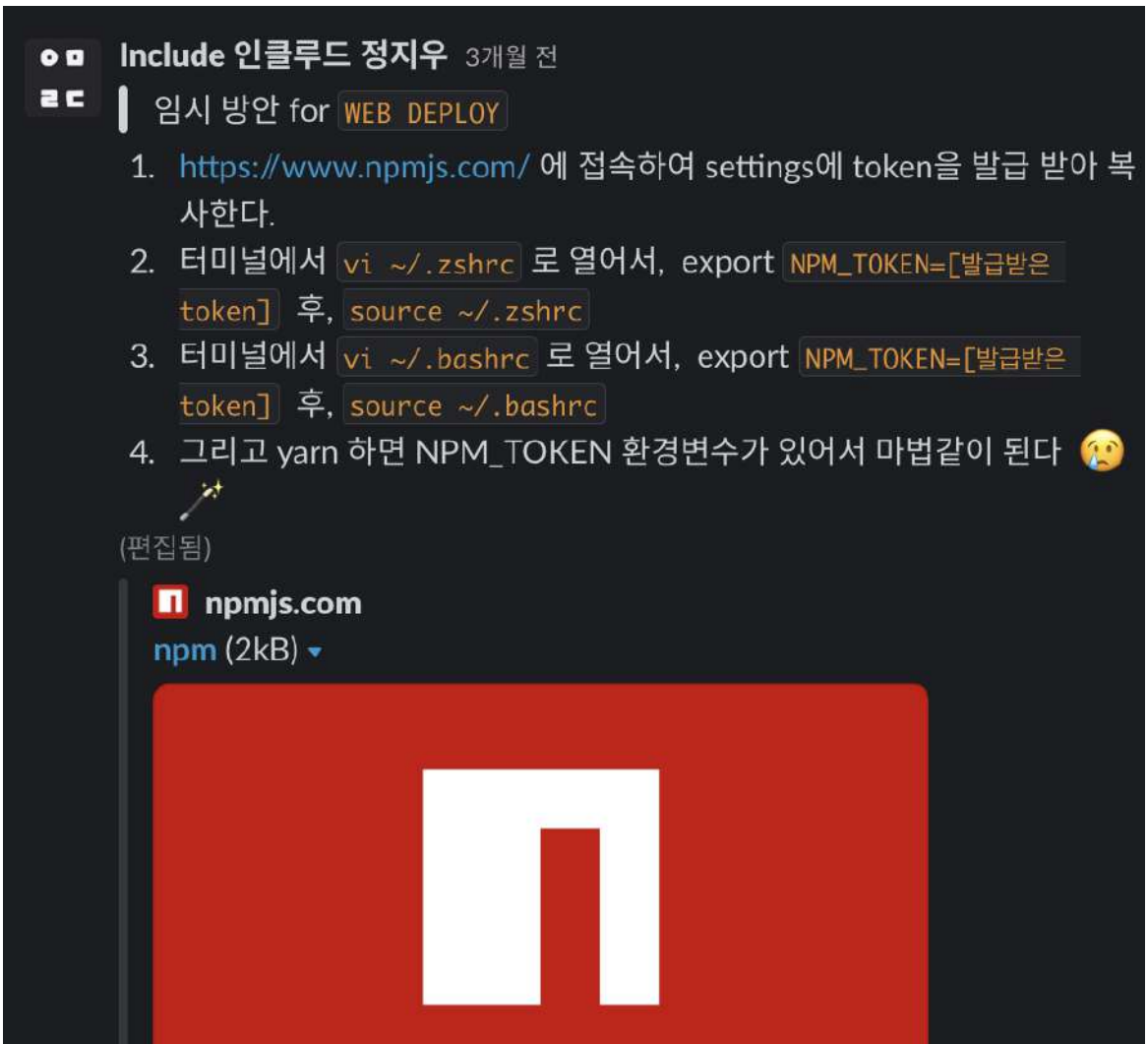
해결

“AWS Elastic Beanstalk의 환경 변수를 활용하여 웹 배포 환경에서 NPM private 패키지를 설치하자!”

- 01 NPM 토큰을 AWS Elastic Beanstalk의 환경변수로 구성하여 .npmrc 파일을 통해 해당 환경변수를 사용하여 배포
- 02 로컬 환경의 node 모듈 설치 시 npm error 발생
- 03 임시 방편으로 ~/.zshrc와 ~/.bashrc에 환경변수를 만들어 로컬 환경에서도 npm 모듈 설치 가능하도록 대응



AWS Elastic Beanstalk에서
NPM private 패키지 설치를 위한 노력



임시방편

회고

Keep

- AWS Elastic Beanstalk 에러 로그를 통한 정확한 원인 분석
- 환경변수를 통한 NPM 토큰 보안화
- 임시 방편이지만 로컬 환경에서의 에러 해결 방법 제시

Problem

- mono repo로 설계 했다면 NPM private 패키지로 분리되지 않았기 때문에 이슈가 발생하지 않았을 것

Try

- 배포 환경에서 “전역 설정”, “프로젝트 설정”, “.npmrc 파일” 등 3가지로 관리 되고 있는 NPM 관련 설정을 AWS Elastic Beanstalk의 환경변수만 사용하도록 통일

06

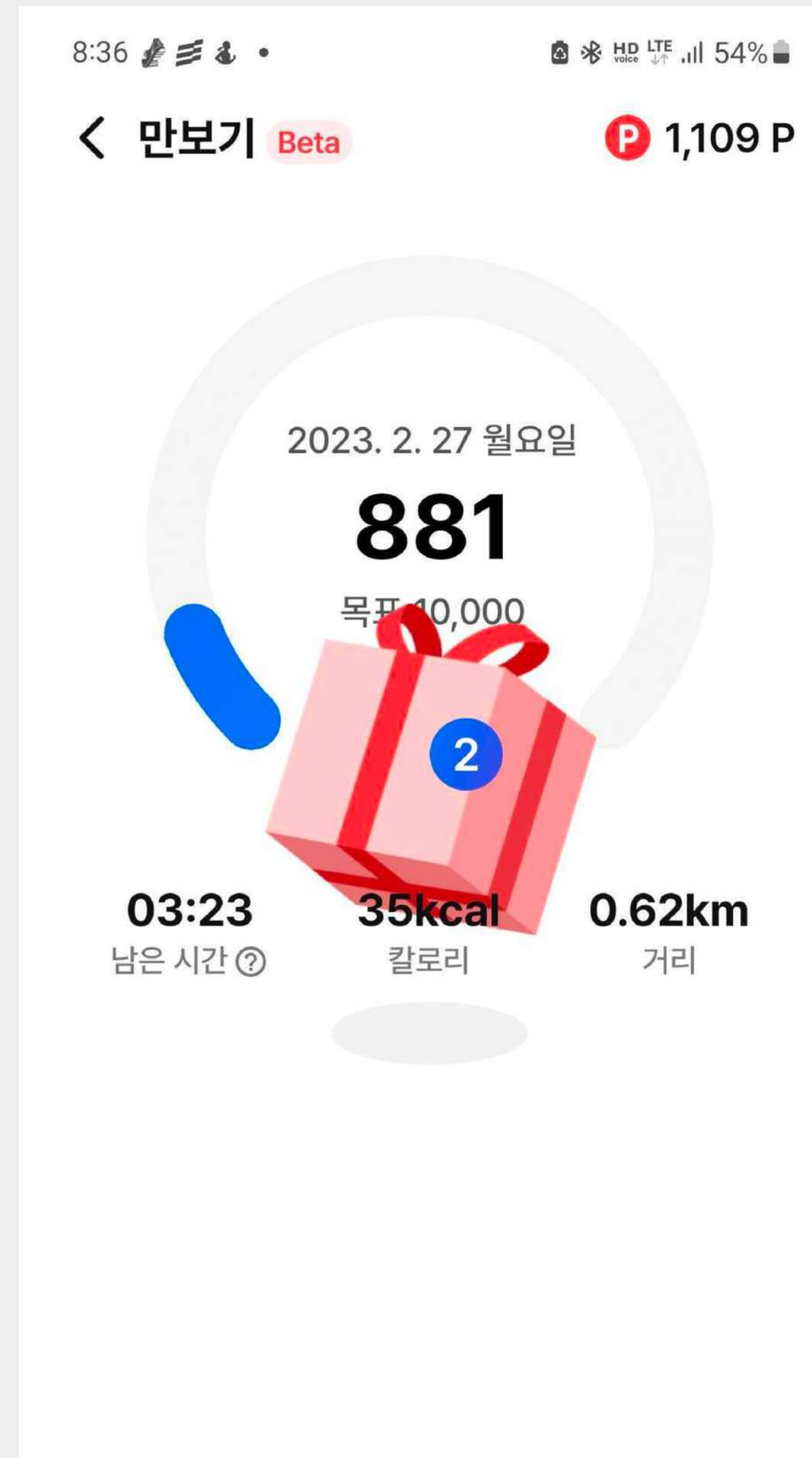
프레임워크 버전에 의존성을 가지는 버그의 근본적 원인 해결

[안드로이드 이미지 이슈 해결 블로그 Link >](#)

[iOS TextInput 이슈 해결 블로그 Link >](#)



안드로이드 이미지 이슈 해결



iOS TextInput 이슈 해결



문제/배경

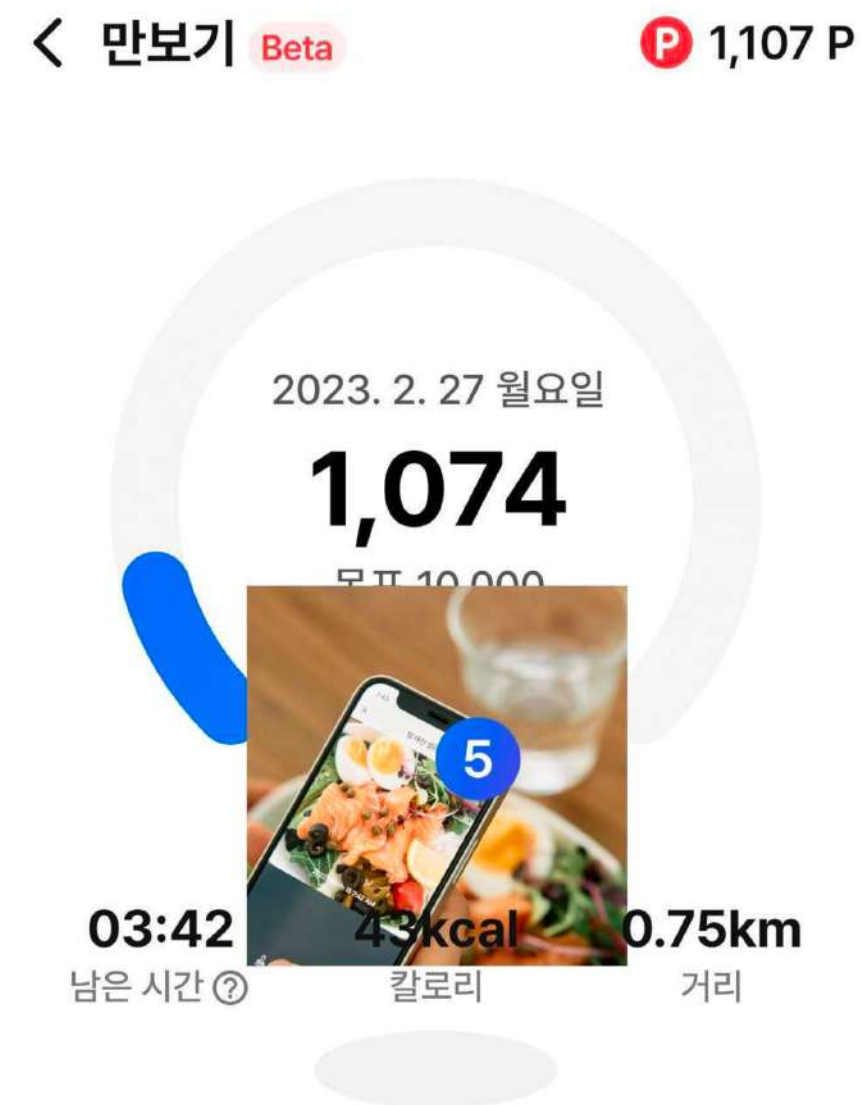
- Android 기기에서 마켓 버전 업데이트 시, 지정한 경로의 static image 아닌 다른 경로의 이미지가 보이는 버그 리포트

해결

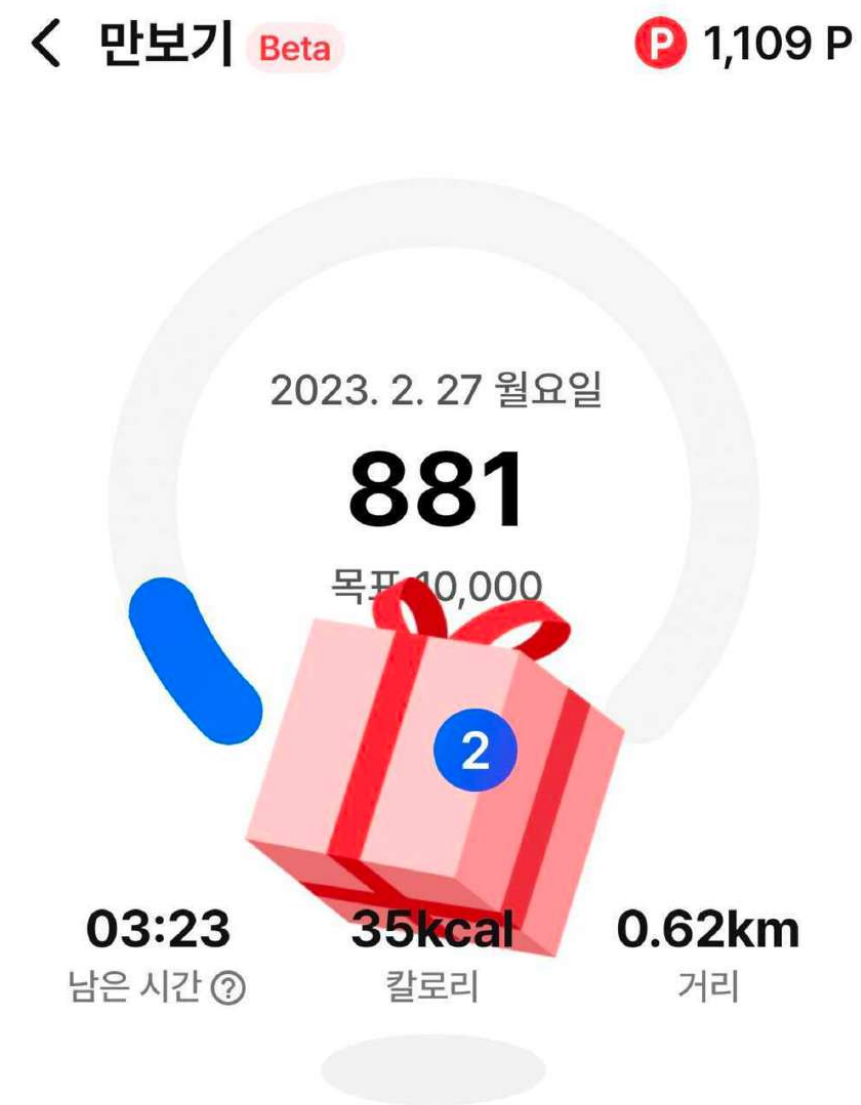
“프레임워크 버전에 의존성을 가지는
버그의 근본적 원인을 찾아서 해결하자!”

- 01 React Native Android에서 GIF 이미지를 사용하기 위해 android/app.gradle에 추가한 facebook의 fresco 2.6.0 버전 문제임을 확인
- 02 fresco 버전을 2.5.0으로 다운그레이드

ASIS



TOBE



문제/배경

- iOS 기기의 React Native 0.67.3 이전 버전에서, TextInput 한글 입력 중 커서 이동 후 재입력 시 글자가 깨지는 버그 리포트
- 사용 중인 React Native 버전(0.66.5)과 의존성을 가지는 패키지가 많아 React Native 버전 업그레이드를 할 수 없어 해당 이슈를 known issue로 처리 중

ASIS



TOBE



해결

“프레임워크 버전에 의존성을 가지는
버그의 근본적 원인을 찾아서 해결하자!”

- 01 React Native 0.67.3 릴리즈 노트를 확인하여 해당 이슈를 해결한 Pull Request 찾음
- 02 해당 Pull Request의 코드를 현재 버전의 React Native 모듈에 반영
- 03 patch-package를 통해 node 모듈이 설치 될 때도 변경되지 않도록 함

회고

Keep

- patch-package를 통해 해당 React Native 버전에 의존성을 갖는 패키지를 유지하며 해당 이슈만 해결하는 아이디어 제시
- 재현경로를 파악하여 이슈 해결
- known issue의 근본적 원인을 찾아 해결
- QA를 통해 사이드이펙트 확인

Problem

- 재현경로를 정확히 파악하지 않은 채 로컬 환경에서 발생하지 않아 known issue 처리
- 네이티브 코드에 대한 이해도가 낮아 실제 수정 코드를 완벽하게 이해하지 못함

Try

- known issue의 근본적 원인 찾아서 이슈 해결
- 프레임워크 버전에 의존성을 갖는 다른 이슈에 대해 patch-package를 통해 해결
- 전역에 영향을 미칠 수 있는 patch-package 사용은 검토 필요

AI 추천 기반 개인 맞춤 식단 구성 서비스, 마이쉽단



앱 배포 1.5달만에 1만+ 다운로드 달성
유일한 프론트엔드 개발자로서 시리즈A 투자 기여
웹/앱/어드민 개발

- 01 성능 최적화를 통한 LCP 개선
- 02 애니메이션 최적화
- 03 웹뷰 앱 개발/배포
- 04 이미지 다운로드 자동화를 통한 업무 효율화
- 05 딥링크 URL 생성 실행 파일 개발
- 06 스토리북 도입 및 웹 배포

 [앱스토어 Link >](#)

 [플레이스토어 Link >](#)

 [웹 Link >](#)

01



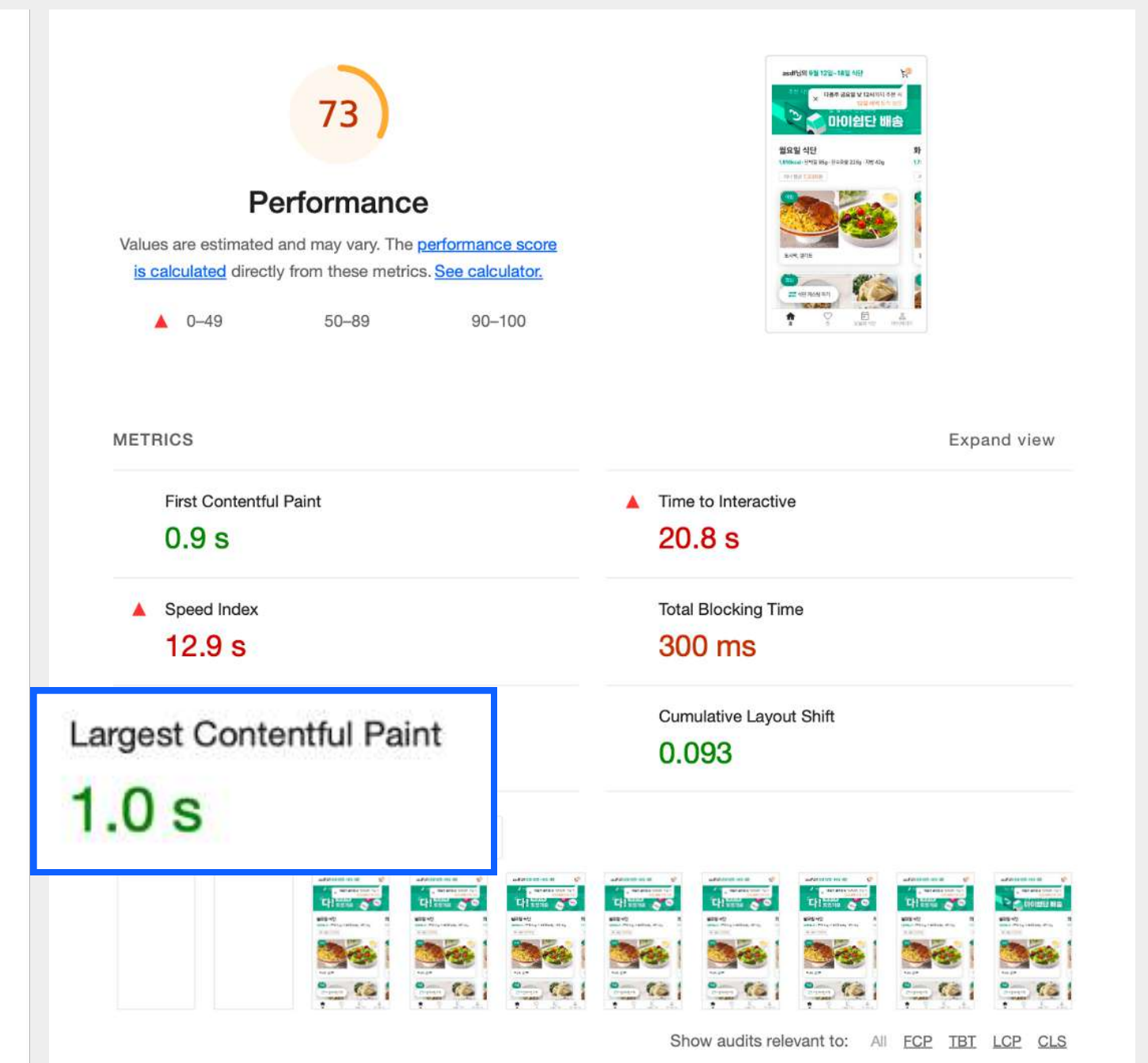
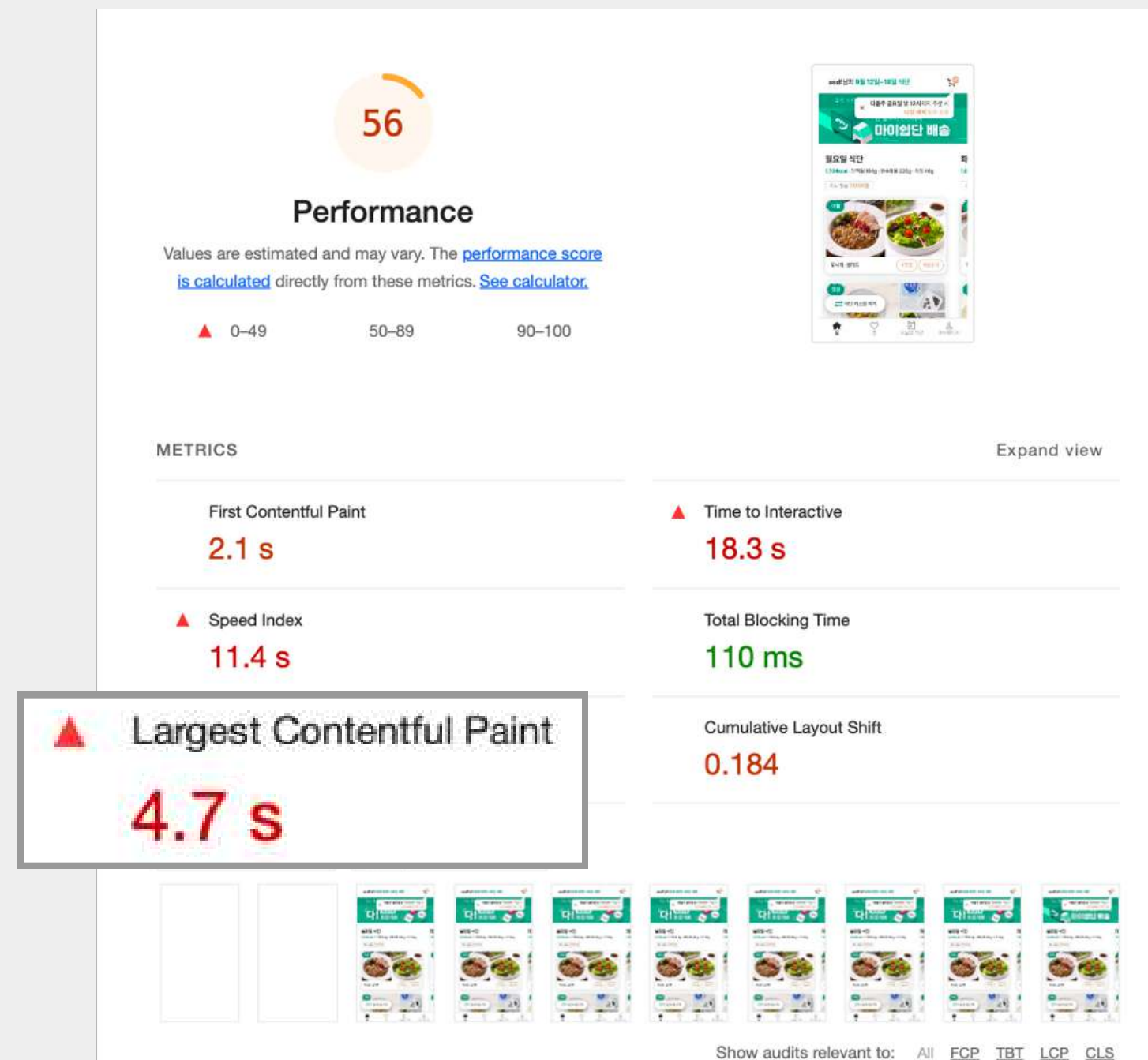
성능 최적화를 통한

LCP 4.7초 → 2.6초 개선

[발표자료 Link >](#)

ASIS

TOBE

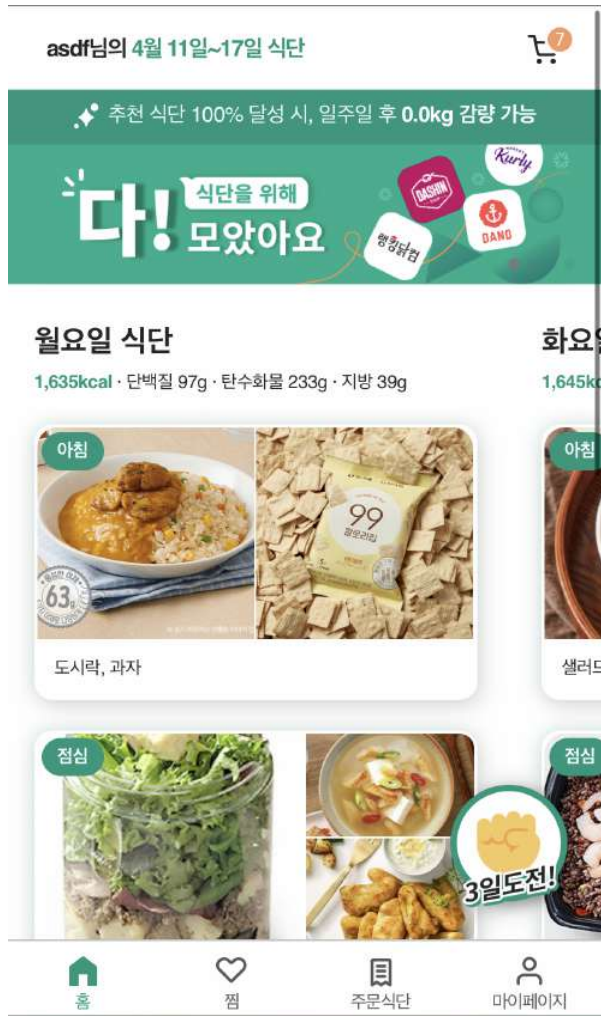


문제/배경

- 홈 스크린 내 이미지를 불러오는 시간이 오래 소요됨

- 서비스 내 이미지의 문제점 -

- 모든 이미지의 확장자 JPG, PNG
- 파일 크기 2MB 등 필요 이상의 큰 크기의 파일
- 모든 이미지를 한 번에 요청
- 홈 스크린 내 식단 이미지 최대 72개

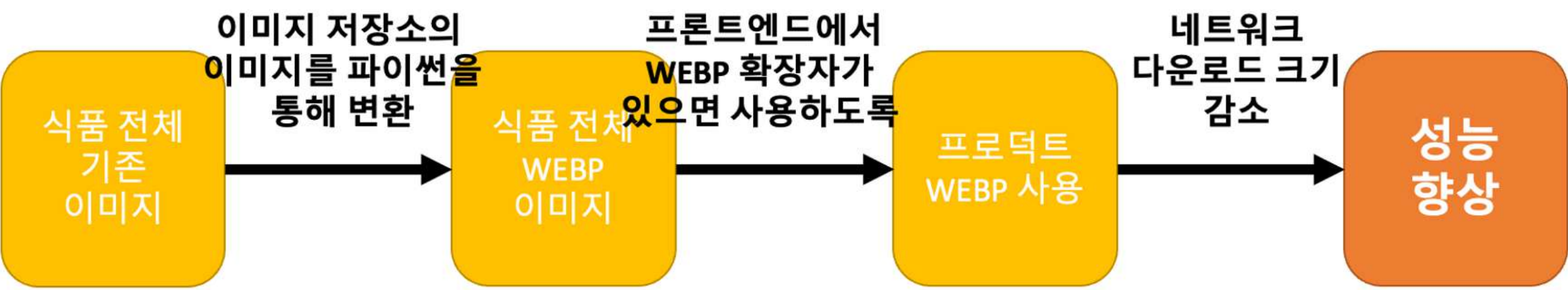


최대 72개의 식품 이미지를 보여주는 홈 화면

해결

- “이미지를 **WEBP**로 변환하자!”
- “이미지를 **리사이징** 하자!”
- “**Lazy Loading**을 구현하자!”

- 01 Python을 통해 AWS S3 버킷 내 모든 식품 이미지 WEBP 변환 및 리사이징
- 02 <picture/> 태그를 이용한 컴포넌트 개발을 통해 크로스 브라우징 이슈 해결
- 03 IntersectionObserver를 통한 이미지 Lazy Loading 구현



Python을 통한 식품 이미지 확장자 변환 및 리사이징 과정

문제/배경

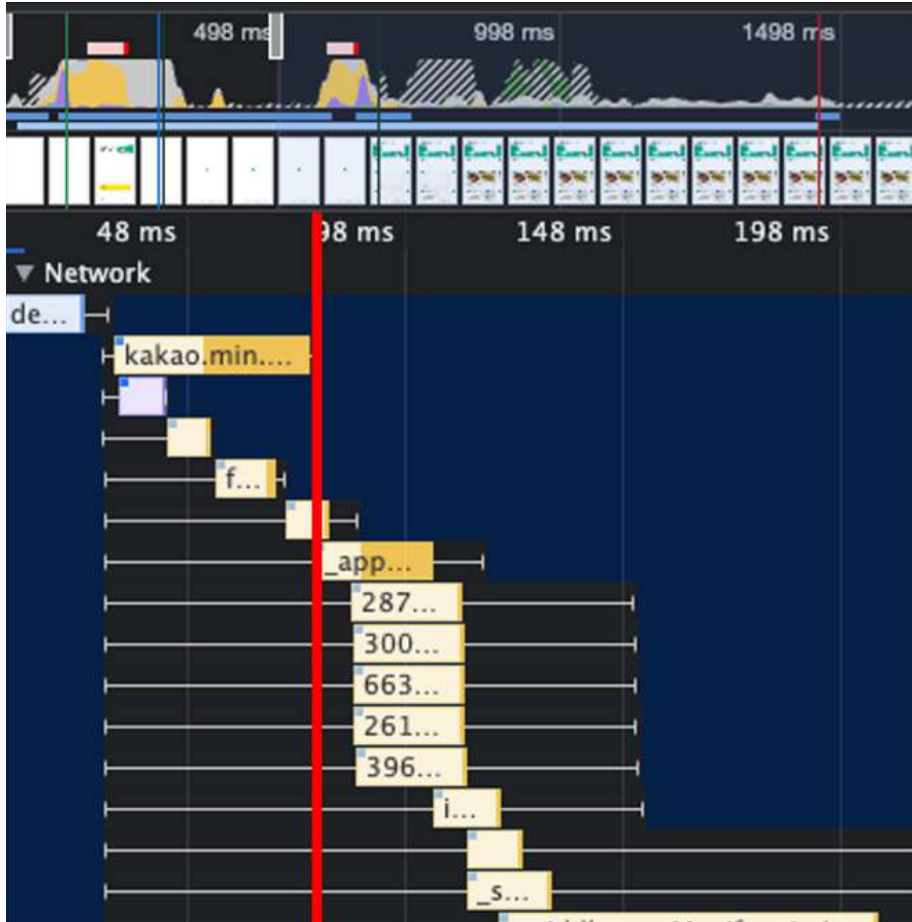
- 카카오 로그인 API script가 동기 처리되어 해당 script 실행이 끝나기 전까지 다른 script를 실행하지 못함

해결

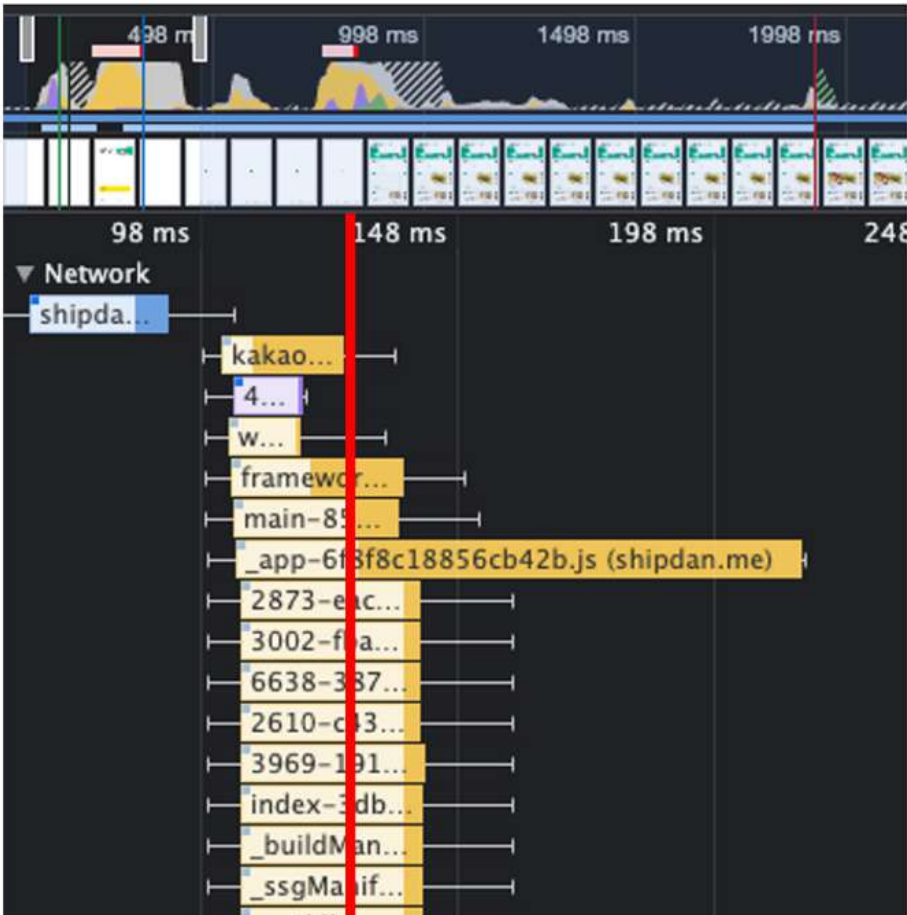
“script 실행을 비동기로 처리하여 다른 script와 함께 실행 시키자!”

01 script를 비동기 처리

ASIS



TOBE



회고

Keep

- 이미지 확장자 변경 및 리사이징을 위한 3가지 해결 방법(AWS Lambda 함수 활용, Sharp.js 활용, Python을 통한 확장자 변경 및 리사이징) 제시
- Chrome Lighthouse의 상세 네트워크 다운로드 상황을 분석하여 문제 원인 파악

Problem

- 이미지와 script 관련 지식 부족으로 인해 발생한 높은 LCP

Try

- 빠른 서비스 개발을 위한 LocalStorage Auth처리를 Cookie를 통한 처리로 리팩토링
- 이를 통해 SSR을 활용하여 클라이언트 사이드 상태에 따른 렌더링이 아닌, 서버사이드에서 상태에 따른 HTML을 다운로드 하는 것으로 수정하여 LCP 추가 개선

02



애니메이션 최적화를 통한 브라우저 렌더링 과정 개선

 [블로그 Link >](#)

문제/배경

- 브라우저의 렌더링 과정 이해
- 현재 애니메이션 구현 방식은 Reflow, Repaint를 유발한다는 것을 깨달음

해결

“불필요한 **Reflow, Repaint**를 유발하는 애니메이션을 최적화하자!”

01 width, left, right와 같이 Reflow/Repaint를 발생 시키는 기존 코드를 transform으로 변경

02 color와 같이 Repaint를 발생 시키는 기존 코드 중 opacity로 변경할 수 있는 부분 변경

ASIS

```
const Circle = styled.div<{ active: boolean }>`
  width: 20px;
  height: 20px;
  border-radius: 100px;
  background-color: #ffffff;
  transition: transform 400ms ease;
  transform: translateX(${active ? 'toActive' : 'toInactive'}) 0.4s
```

```
@keyframes toActive {
  0% {
    margin-left: 0;
  }
  100% {
    margin-left: 24px;
  }
}
@keyframes toInactive {
  0% {
    margin-left: 24px;
  }
  100% {
    margin-left: 0;
  }
}
```

TOBE

```
const Circle = styled.div<{ active: boolean }>`
  width: 20px;
  height: 20px;
  border-radius: 100px;
  background-color: #ffffff;
  transform-origin: left;
  transform: translateX(${active} => (active ? '24px' : 0));
  transition: transform 400ms ease;
```

```
transform-origin: left;
transform: translateX(${active} => (active ? '24px' : 0));
transition: transform 400ms ease;
```


회고

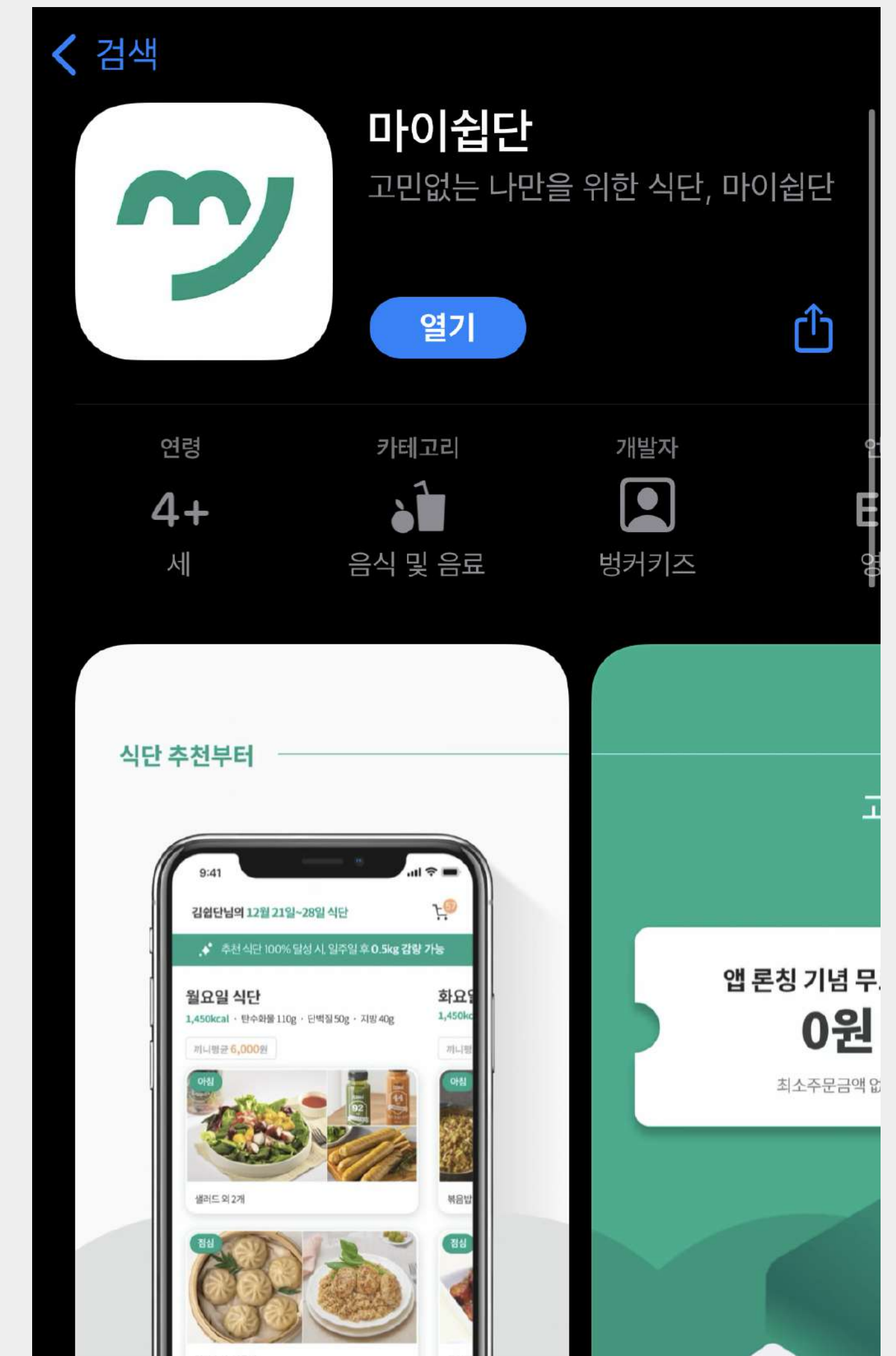
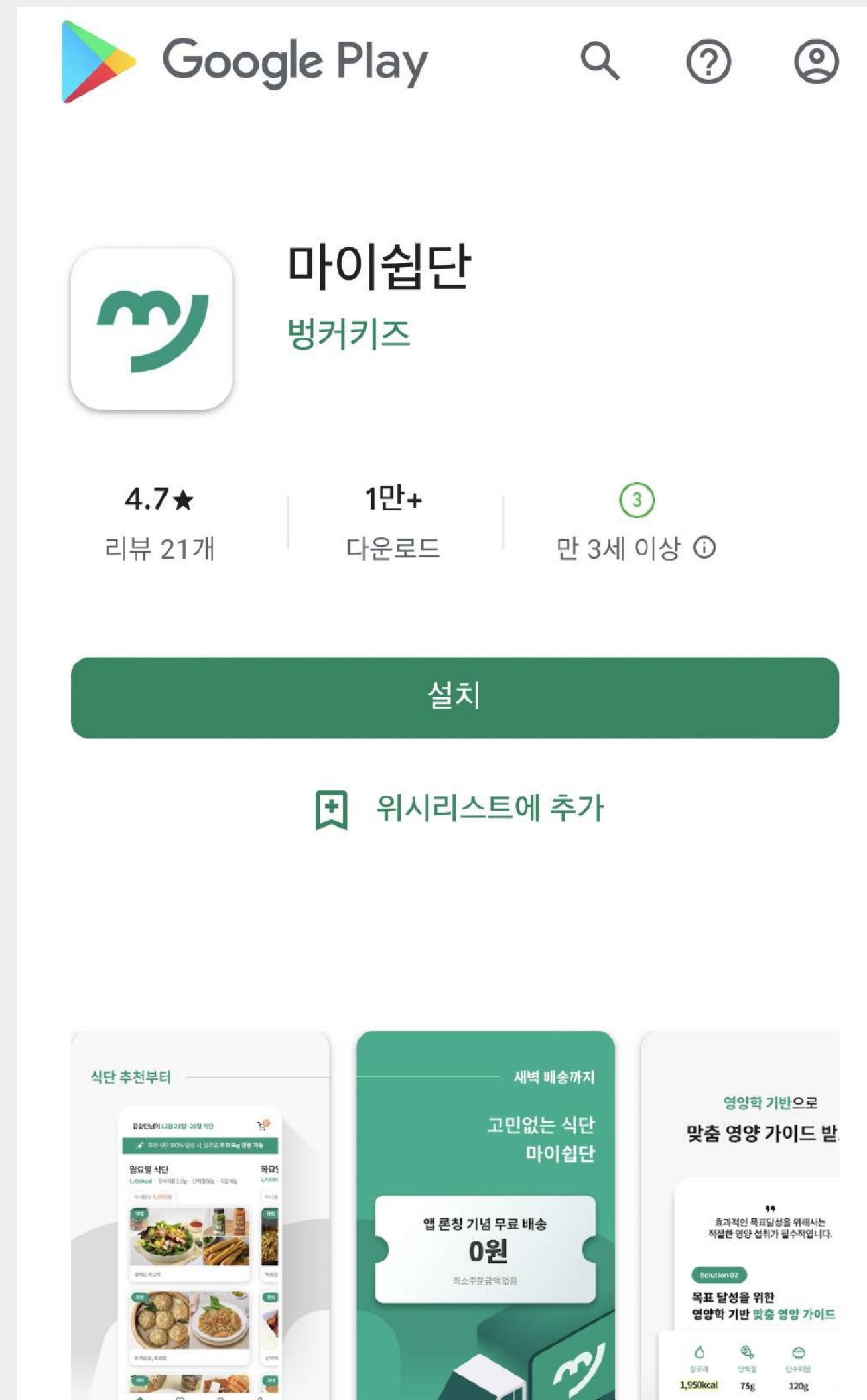
Keep

- 브라우저의 렌더링 과정을 이해하고 개선 가능한 애니메이션에 대해 최적화
- Reflow, Repaint가 꼭 필요한 애니메이션인지 고려하여 애니메이션을 개발

03

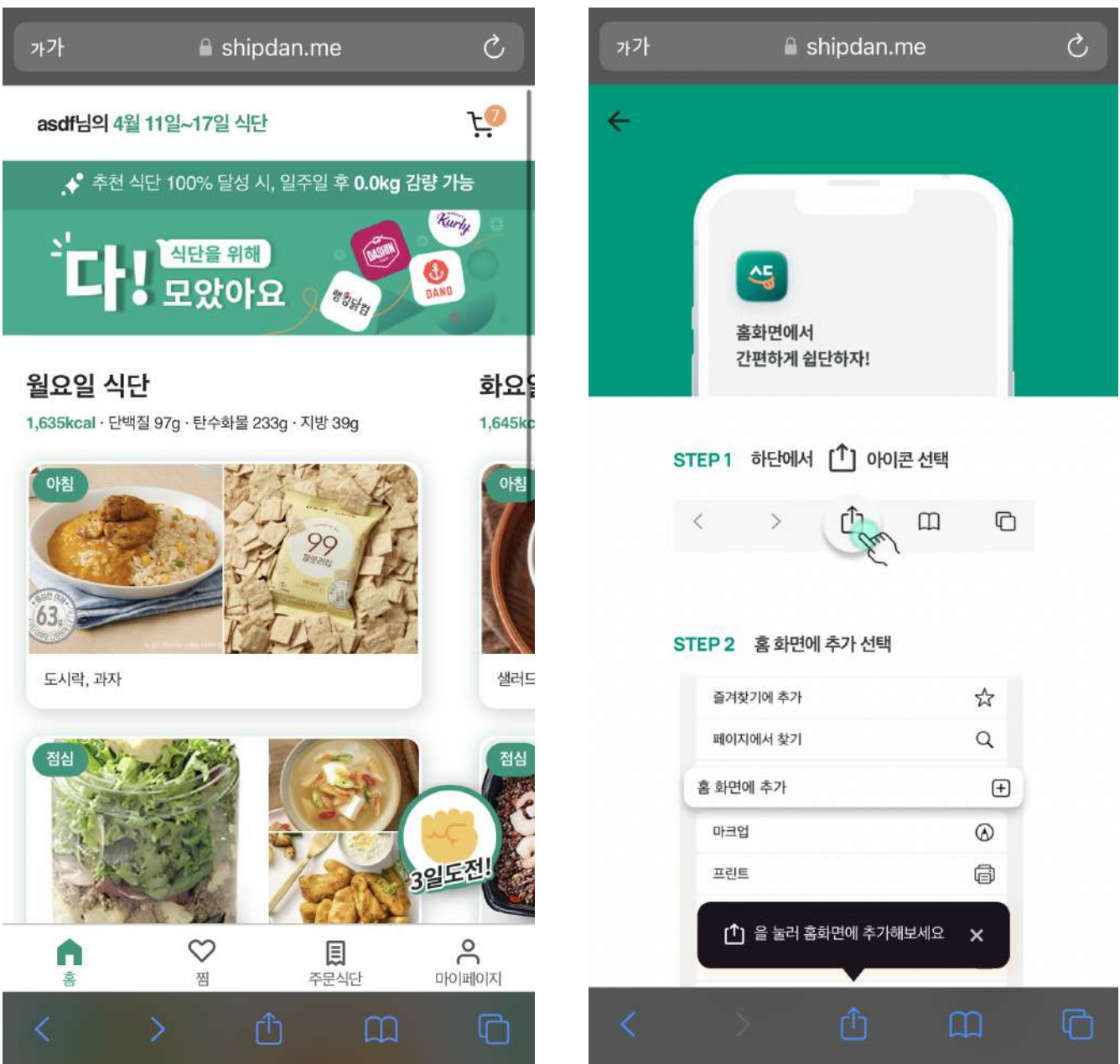
웹뷰 앱 개발/배포

[블로그 Link >](#)



문제/배경

- 웹으로 구현된 서비스 진입이 불편하다는 VOC
- 서비스 접근성 향상을 위해 PWA를 개발하였으나 복잡한 설치과정



기존의 웹 서비스와 PWA 설치화면

해결

“React Native 웹뷰 앱을 개발하자!”

- 01 해결 방안(네이티브 앱, React Native 웹뷰 앱, React Native 앱) 중 아래 이유로 웹뷰 앱을 선택하여 개발
 - 잦은 실험/수정/배포가 필요한 팀의 상황에 앱 검수가 필요 없는 장점
 - 기존 웹 프로덕트가 모바일 사이즈에 대응 중이기 때문에 모든 화면을 개발할 필요 없음
- 02 React Native Module이 필요한 경우에 웹 → 앱 / 앱 → 웹 이벤트를 발생시켜 웹/앱 각각에서 이벤트 핸들러를 통해 구현



22.05.13 안드로이드 배포 이후, 22.07.02 1만+ 다운로드

22.05.19 iOS 배포 이후, 22.06.27 1만+ 다운로드

22.06.28 기준 앱 스토어 음식 및 음료 무료 앱 인기 순위 17위(최고 순위 16위)

회고

Keep

- 문제 상황을 해결하기 위해 3가지(네이티브 앱, React Native 웹뷰 앱, React Native 앱) 해결 방안 중 팀의 상황에 맞는 선택을 통한 빠른 실험/검증 비즈니스에 기여
- 웹 → 앱 / 앱 → 웹 간의 이벤트명 기반 동작 방식 설계

Problem

- 앱에서 지정하지 않은 이벤트명에 대한 undefined 미처리로 인해 앱 크래시 발생 경험
 - “웹 배포 롤백 → 원인 파악 → 앱 작업 → 앱 검수 및 배포 → 웹 재배포”를 통해 해결
- 코드푸시를 설정이 되어있었다면 앱 검수 및 배포 과정 없이 해결할 수 있었을 것

Try

- Javascript로 해결할 수 있는 이슈를 앱 검수 및 배포 과정 없이 해결할 수 있도록 코드푸시 설정

이미지 다운로드 자동화를 통한 8시간 업무 → 0.15시간으로 단축

6월20일~6월25일

정지우 님의 20220616_0000000 주문

20일 월요일 식단

7개 제품

21일 화요일 식단

8개 제품

22일 수요일 식단

8개 제품

23일 목요일 식단

8개 제품

24일 금요일 식단

1,941kcal · 단백질 74g · 탄수화물 219g · 지방 56g

☼ 아침



[랭킹달콤] 잇메이트 닭가슴살 소떡소떡 매콤
270kcal
탄수화물 45g · 단백질 18g · 지방 2g
1개



[마이쉽단] 샐러드판다 발사믹 드레싱 유러피안 샐러드
147kcal
탄수화물 19g · 단백질 2g · 지방 10g
1개



☀ 점심



[마이쉽단] 브랜드닭 닭가슴살 볶음밥 버섯불고기
400kcal
탄수화물 65g · 단백질 20g · 지방 8g
1개



[마이쉽단] 브랜드닭 닭가슴살 핫바 숯불갈비
109kcal
탄수화물 2g · 단백질 14g · 지방 5g
1개



[마이쉽단] 서울우유 비요트 요거트 초코링
205kcal
탄수화물 27g · 단백질 6g · 지방 8g
1개



🌙 저녁

☼ 아침



[마켓컬리] Kurly Only 닭가슴살 불고기맛 곤약볶음밥
265kcal
탄수화물 43g · 단백질 16g · 지방 3g
1개



[마이쉽단] 호밀 이츠너츠
115kcal
탄수화물 6g · 단백질 3g · 지방 8g
1개



☀ 점심



[마이쉽단] 브랜드닭 닭가슴살 볶음밥 매콤야채
395kcal
탄수화물 70g · 단백질 20g · 지방 5g
1개



[랭킹달콤] 맛있는 닭가슴살 스틱 코코넛 가스
272kcal
탄수화물 15g · 단백질 18g · 지방 16g
1개



🌙 저녁



[마켓컬리] 남향푸드도또아 핫치킨 브리또
325kcal
탄수화물 34g · 단백질 16g · 지방 14g
1개



☼ 아침



[랭킹달콤] 잇메이트 닭가슴살 구운 주먹밥 김치
165kcal
탄수화물 29g · 단백질 6g · 지방 3g
1개



[마이쉽단] 샐러드판다 오리엔탈 두부면 샐러드
220kcal
탄수화물 16g · 단백질 20g · 지방 9g
1개



☀ 점심



[마이쉽단] 샐러드판다 메밀면 샐러드
310kcal
탄수화물 52g · 단백질 9g · 지방 6g
1개



[마이쉽단] 아침물 닭가슴살비엔나 오리지널
138kcal
탄수화물 4g · 단백질 17g · 지방 6g
1개



[마이쉽단] 서울우유 비요트 요거트 크런치볼
205kcal
탄수화물 26g · 단백질 7g · 지방 8g
1개



🌙 저녁



[마이쉽단] 라쿠치나 콤포포르마지오 파니니

☼ 아침



[랭킹달콤] 맛있닭 닭가슴살 한끼 소세지빵
290kcal
탄수화물 37g · 단백질 13g · 지방 10g
1개



[랭킹달콤] 잇메이트 닭가슴살 스테이크 오리지널
165kcal
탄수화물 12g · 단백질 20g · 지방 4g
1개



☀ 점심



[마켓컬리] 헛반 컵반 콩나물해장국밥
220kcal
탄수화물 49g · 단백질 5g · 지방 1g
1개



[마이쉽단] 샐러드판다 시저 샐러드랩
466kcal
탄수화물 55g · 단백질 21g · 지방 18g
1개



🌙 저녁



[마이쉽단] 라쿠치나 콤포포르마지오 파니니
350kcal
탄수화물 35g · 단백질 16g · 지방 16g
1개



☼ 아침



[랭킹달콤] 맛있닭 닭가슴살 한끼 소세지빵
290kcal
탄수화물 37g · 단백질 13g · 지방 10g
1개



[랭킹달콤] 잇메이트 닭가슴살 스테이크 오리지널
165kcal
탄수화물 12g · 단백질 20g · 지방 4g
1개



☀ 점심



[마켓컬리] 헛반 컵반 콩나물해장국밥
220kcal
탄수화물 49g · 단백질 5g · 지방 1g
1개



[마이쉽단] 샐러드판다 시저 샐러드랩
466kcal
탄수화물 55g · 단백질 21g · 지방 18g
1개



🌙 저녁



[마이쉽단] 라쿠치나 콤포포르마지오 파니니
350kcal
탄수화물 35g · 단백질 16g · 지방 16g
1개



문제/배경

- 식단 구매 유저들의 일주일 치 식단을 인쇄하여 함께 배송해주는 실험 진행
- 두 분의 팀원께서 매주 4시간씩 식단 스크린샷을 찍는 반복 작업 진행

해결

“개발을 통해 반복되는 워크플로우를 개선하자!!”

- 01 일주일 치 식단 조회 페이지에 html2canvas로 구현한 식단 이미지 다운로드 버튼 추가
- 02 해당 버튼을 클릭하는 셀레니움 스크래퍼 개발
- 03 식단 이미지 다운로드용 슈퍼 유저 계정 추가
- 04 해당 주차 주문 엑셀파일에서 주문 id를 추출하여 각 주문 id 별 식단 다운로드 자동화

**반복 작업을 통해 8시간 걸렸던 작업을 단 0.15시간의 작업으로 32배의 효율화
(400건 식단 다운로드 기준)**

회고

Keep

- 반복되는 작업의 프로세스를 이해하고 개발을 통한 해결
- 스스로 일의 범위를 확장하여 실험을 위한 워크플로우 효율화
- 효율화를 통해 팀의 생산성 향상

딥링크 URL 생성 실행 파일 개발을 통한 반복 과정 개선

```
jeongjiwoo — /Users/jeongjiwoo/PycharmProjects/deepLinkUrlGenerator/dist/main; exit — /Users/jeongjiwoo/PycharmProjects/deepLinkUrlGenerator/...  
Adjust Token 값을 입력하세요: temp  
temp  
앱 내 페이지를 입력하세요.(INTRO | MAIN | PAYMENT): MAIN  
MAIN  
웹뷰 내 페이지 URL을 입력하세요.(https://shipdan.me/ 제외): home  
home  
로그인이 필요합니까?(Y | N): Y  
Y  
https://jmgn.adj.st/?adjust_t=temp&screen%5Bpathname%5D=MAIN&webView%5Bpathname%5D=home&needLogin=true  
위 url을 복사해서 붙여넣으세요.  
Adjust Token 값을 입력하세요: █
```

문제/배경

- 마케팅 팀에서 CRM 혹은 광고 소재에서 클릭 시, 딥링크 URL 필요
- 딥링크 URL을 매번 개발팀에 요청

해결

“개발을 통해
반복되는 워크플로우를 개선하자!!”

- 01 앱 딥링크의 데이터 구조에 맞는 URL을 생성할 수 있는 실행 파일을 Python/Pyinstaller를 통해 개발
- 02 마케팅 팀에 배포

실행 파일 배포 이후 마케팅 팀에서 개발 팀에 딥링크 URL 요청 0건

회고

Keep

- 스스로 일의 범위를 확장하여 실험을 위한 워크플로우 효율화
- 반복되는 작업의 프로세스를 이해하여 마케팅 워크플로우 개선

Problem

- 어드민이 있었다면 URL 생성 기능을 추가하여 실행 파일 배포 없이 모든 팀원이 접근 가능하도록 개발했을 것

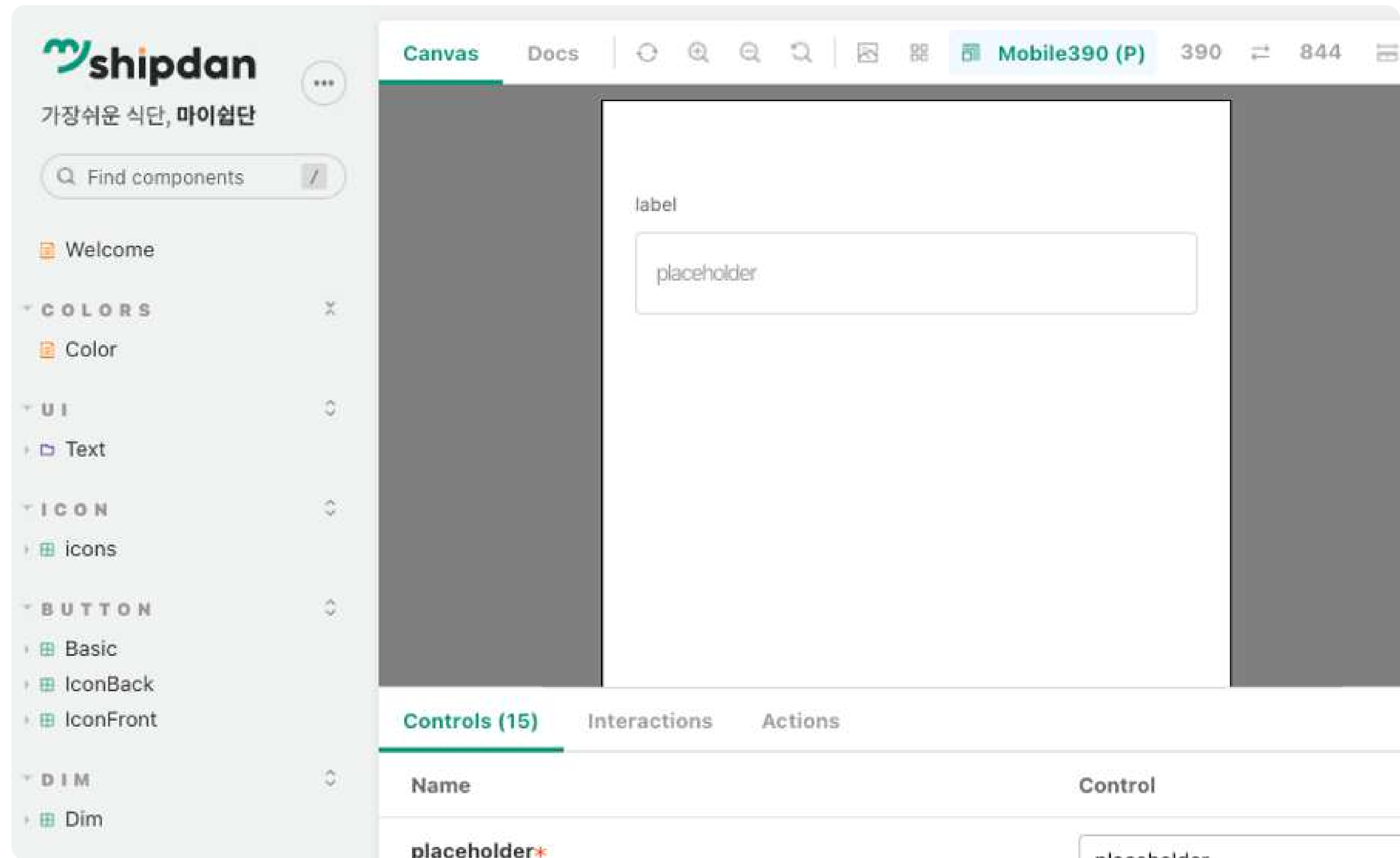
Try

- 실행파일을 배포하기 보다는 팀원들에게 효율화를 제공할 수 있는 것들은 어드민에 배포하여 많은 팀원들이 접근 가능하도록 하기

06

스토리북 도입 및 웹 배포를 통한 커뮤니케이션 비용 절감

[스토리북 Link >](#)



문제/배경

- 프로젝트 팀 내, 컴포넌트 관련 커뮤니케이션 시 각자 다른 명칭을 사용하여 미스커뮤니케이션 발생
- 추후 프론트엔드 협업 시, 디자인시스템을 통해 UI 개발 생산성 향상 기대

해결

“컴포넌트명을 통일하여 커뮤니케이션 비용을 줄이자!!”

“디자인시스템을 통해 UI 개발 생산성을 향상 시키자!!”

01 프로젝트 디자이너와 함께 디자인시스템 구축

02 스토리북 웹 배포

스토리북에 사용한 컴포넌트 명칭으로 통일하여 커뮤니케이션 비용 절감

프론트엔드 UI 개발 생산성 향상

회고

Keep

- 스토리북을 만드는 것에서 그치지 않고 배포하여 팀원들끼리 공유하여 컴포넌트 명칭 통일

Problem

- 디자인 예외 발생 시, 스타일 관련 파라미터를 확장해두지 않아 새롭게 개발해야하는 불편함 발생

Try

- 스타일 관련 파라미터를 컴포넌트에서 전달 받을 수 있도록 확장하여 개발

오픈소스 기여



01 **react-simple-star-rating**

02 **react-native-health**

01



react-simple-star-rating

Next.js 관련 Contribute

 [Pull Request Link >](#)

문제/배경

- 기존 코드에 window와 navigator 객체 내 값을 통해 isTouchDevice 값을 결정
- Next.js에서는 해당 객체들이 init 되기 전에 참조하여 undefined 상태에서 에러 발생

```
const isTouchDevice = useMemo(() =>
  'ontouchstart' in window || navigator.maxTouchPoints > 0, [])
```

ASIS 코드

해결

“Next.js의 동작방식에 맞게 수정하여 기여하자!!”

- 01 window와 navigator 객체가 undefined인 상황에 맞추어 falsy check를 통해 값을 결정하도록 수정

```
const isTouchDevice = useMemo(() =>
  !!window?.ontouchstart ||
  (typeof navigator !== 'undefined' && navigator.maxTouchPoints > 0), [])
```

TOBE 수정한 코드

회고

Keep

- 근본적인 해결책을 찾기 위한 패키지 코드 분석
- Next.js의 동작방식을 이해하여 해결
- Pull Request를 통한 오픈소스 기여

02



react-native-health

타입 관련 Pull Request

 [Pull Request Link >](#)

문제/배경

- 기존 코드에서 제공하는 initHealthKit의 타입의 가독성이 떨어진다고 생각
- callback에서 제공하는 props result의 타입인 HealthValue의 의미 모호

```
initHealthKit(  
  permissions: HealthKitPermissions,  
  callback: (error: string, result: HealthValue) => void,  
): void  
  
interface HealthValue = number;
```

ASIS 코드

해결

“새로운 타입을 만들어 가독성을 높이자!!”

- 01 iOS 건강 정보 정책 상, initHelathKit 함수는 권한 요청 여부에 대해 반환
- 02 InitHealthKitResultCode enum을 추가
- 03 initHealthKit callback 제공 result 타입을 InitHealthKitResultCode로 변경

```
export enum InitHealthKitResultCode {  
  NotDetermined = 0, // 미요청  
  Determined = 1, // 요청  
}  
  
initHealthKit(  
  permissions: HealthKitPermissions,  
  callback: (error: string, result: InitHealthKitResultCode) => void,  
): void
```

TOBE 수정한 코드

회고

Try

- 같은 원시 타입이라도 가독성을 높이기 위해 enum 혹은 constant 활용