



팔딱팔딱

SSGINC 스파로스 아카데미 K-디지털트레이닝 5차수



여보소 : Seahere

수산업체를 위한 수산 시장 특화 유통 / 재고 관리 시스템



01

주제 선정 배경

02

여보소 : SeaHere

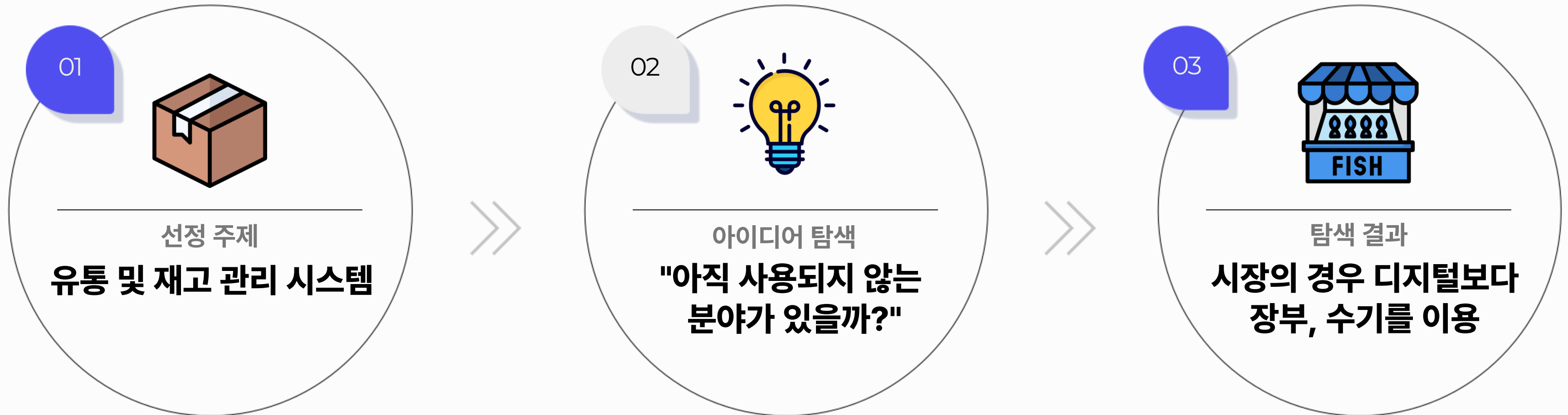
03

프로젝트 개발 과정

04

개발 결과







컴퓨터의 부재

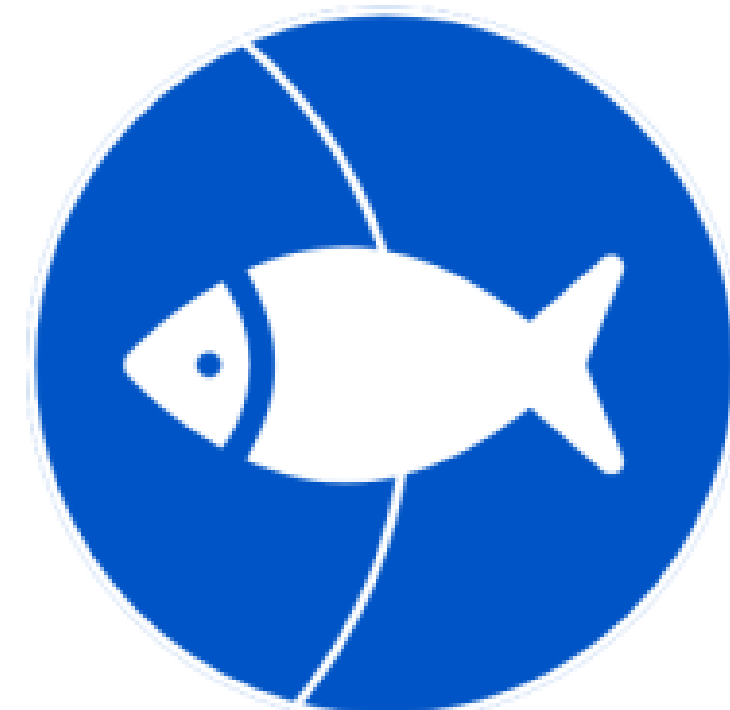


복잡한 유통 과정



결론

“수산 시장에 특화된 포터블 재고 관리 시스템”

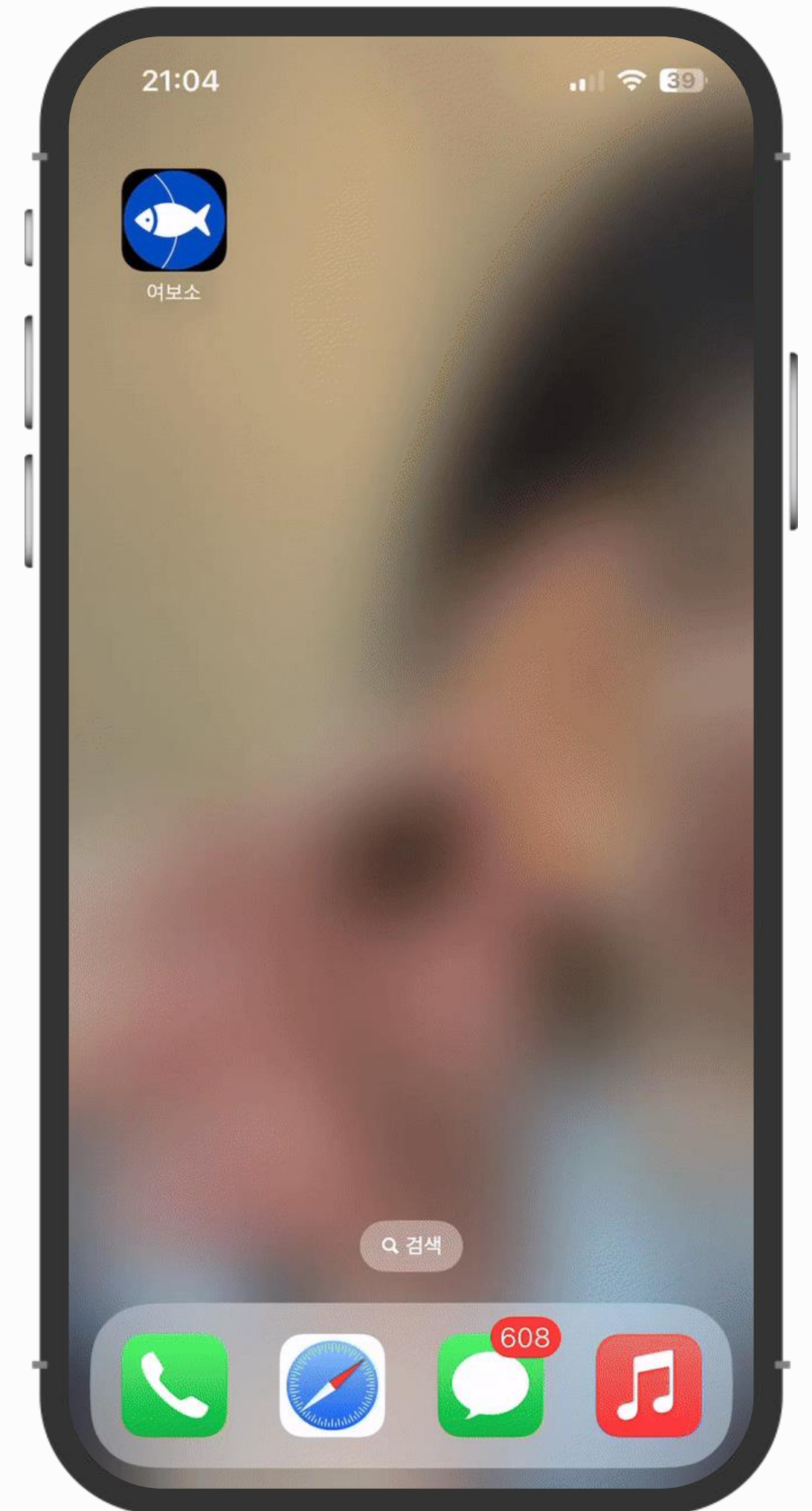


여보소
SEA HERE



여보소:Seahere

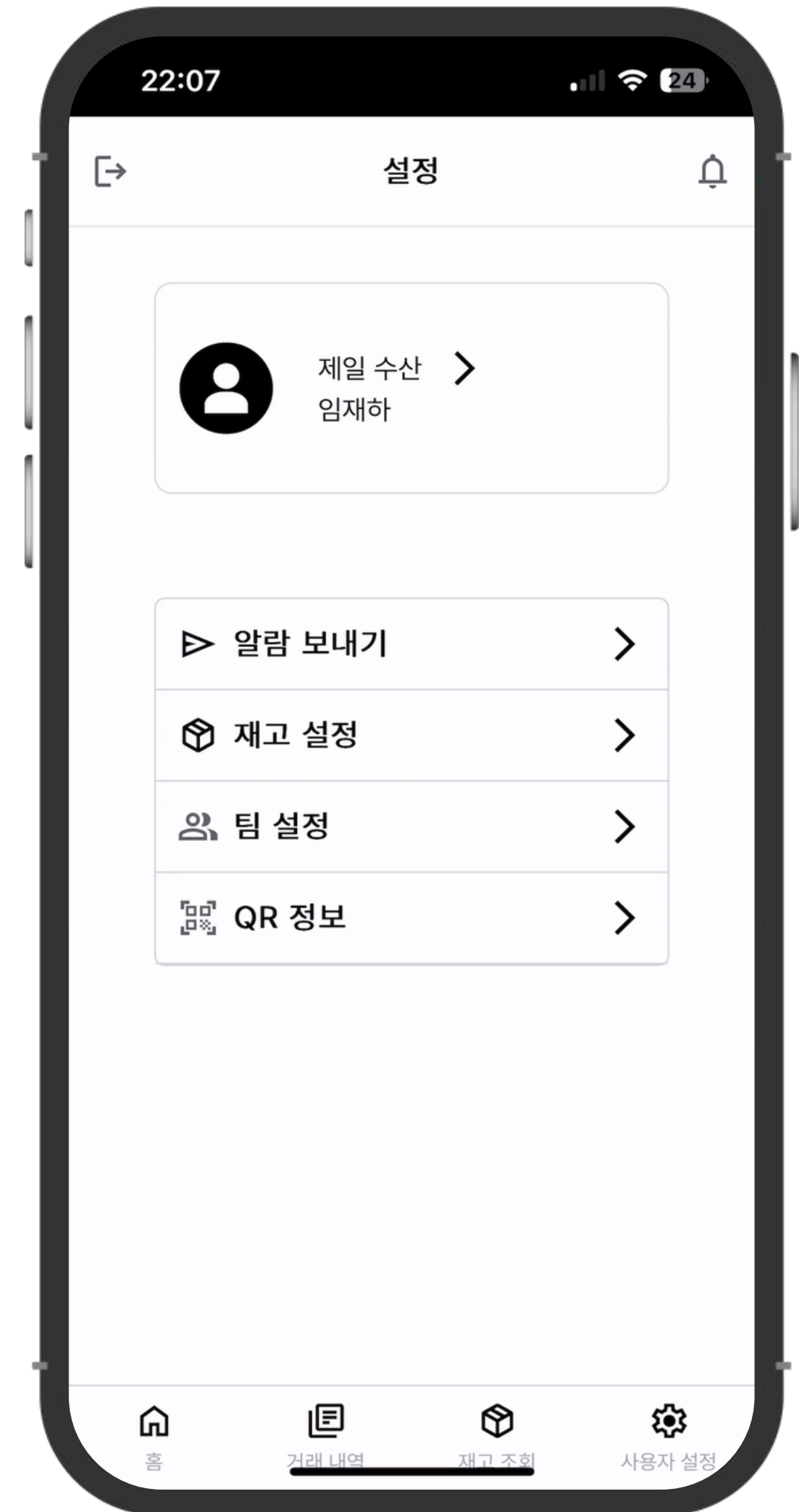
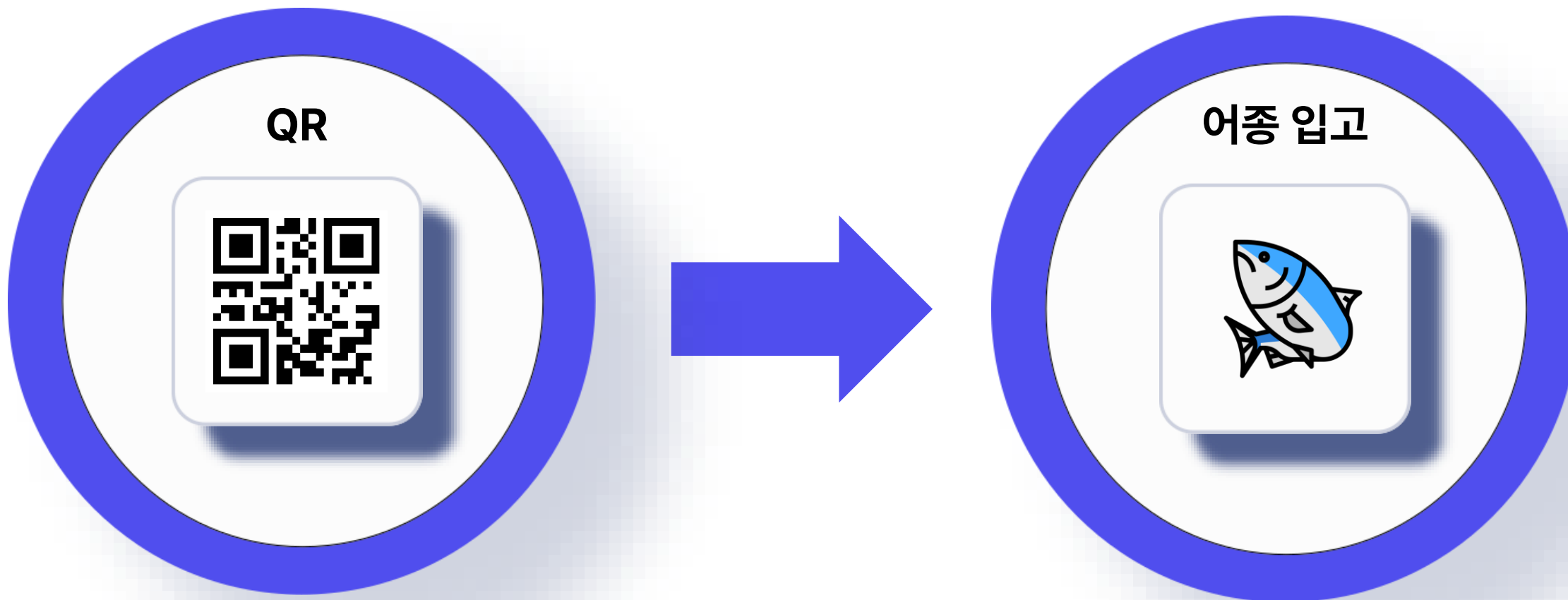
수산 업체를 위한 수산 시장 특화 유통/재고 관리 시스템





여보소:Seahere

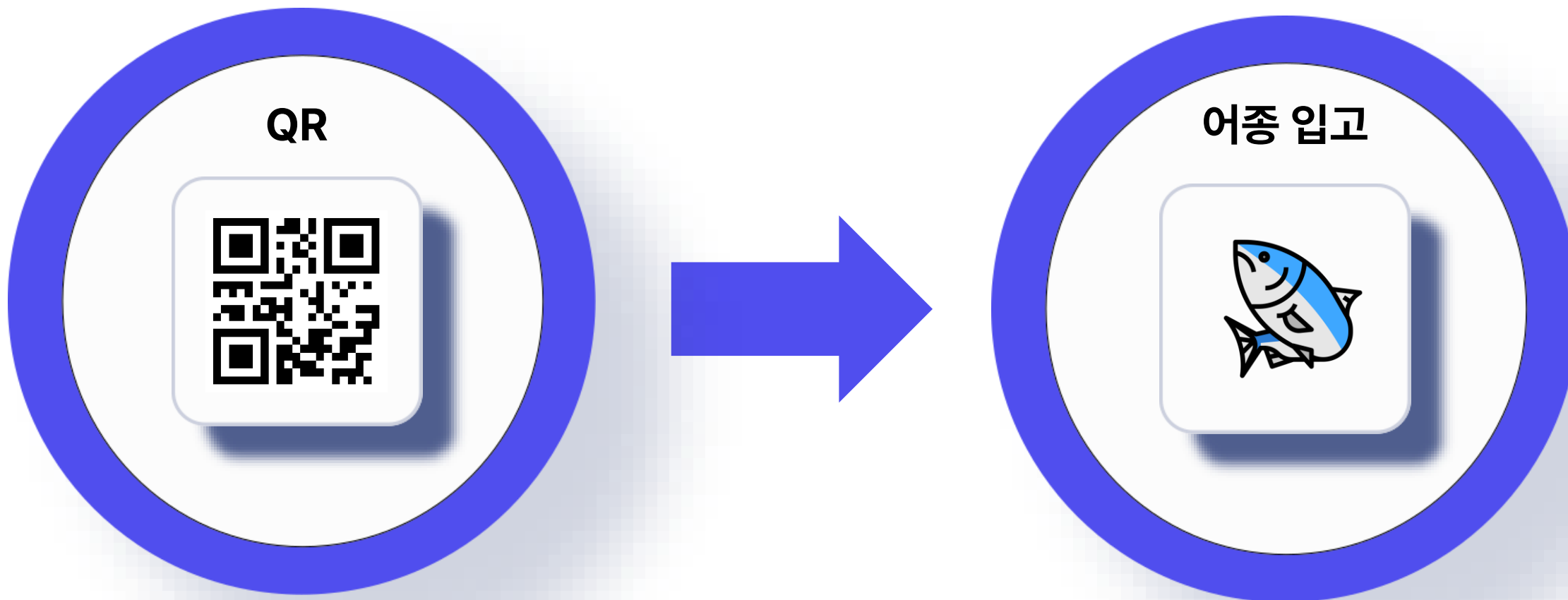
QR 코드를 통한 빠른 입고 처리





여보소:Seahere

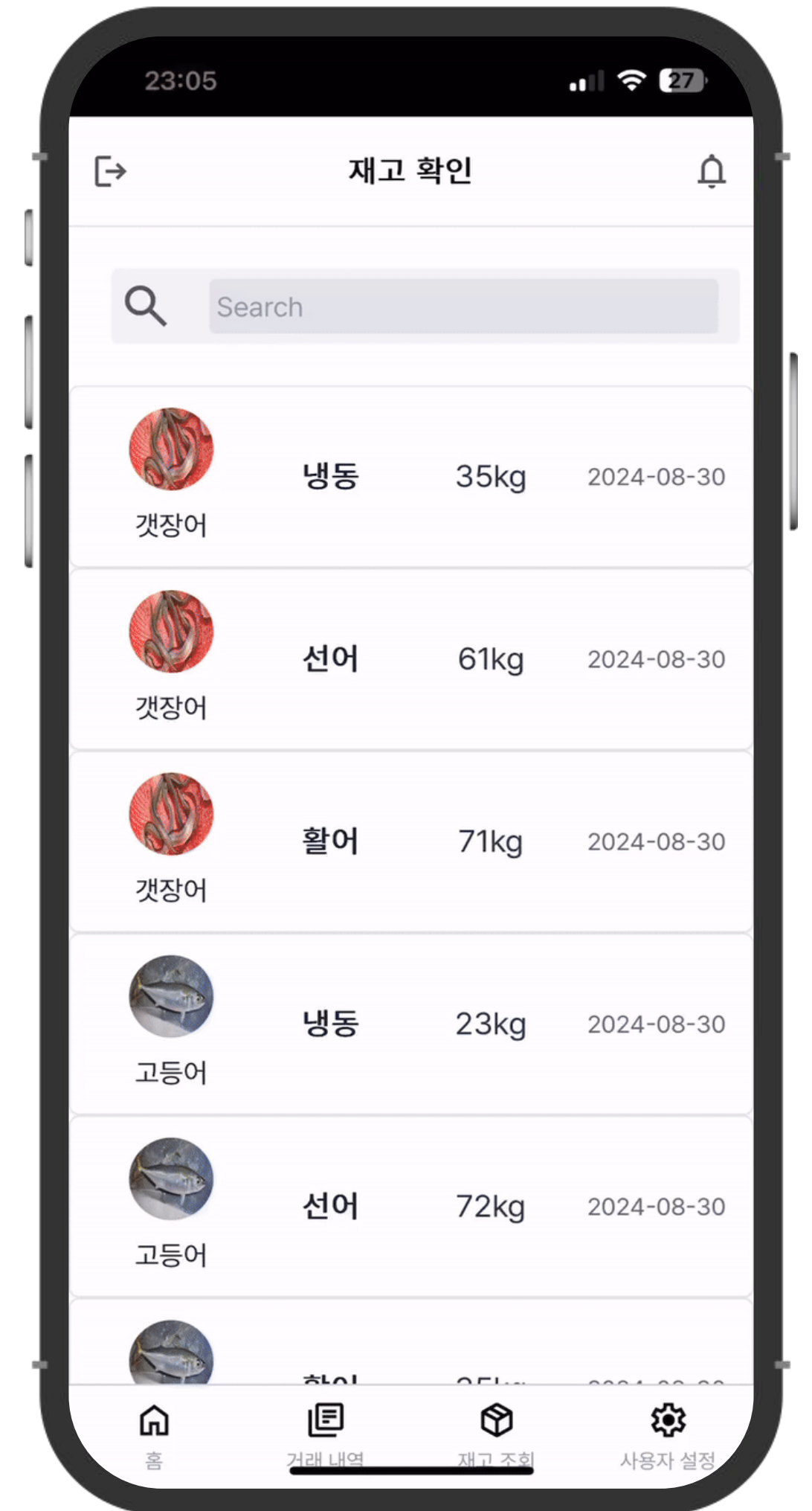
QR 코드를 통한 빠른 입고 처리





여보소:Seahere

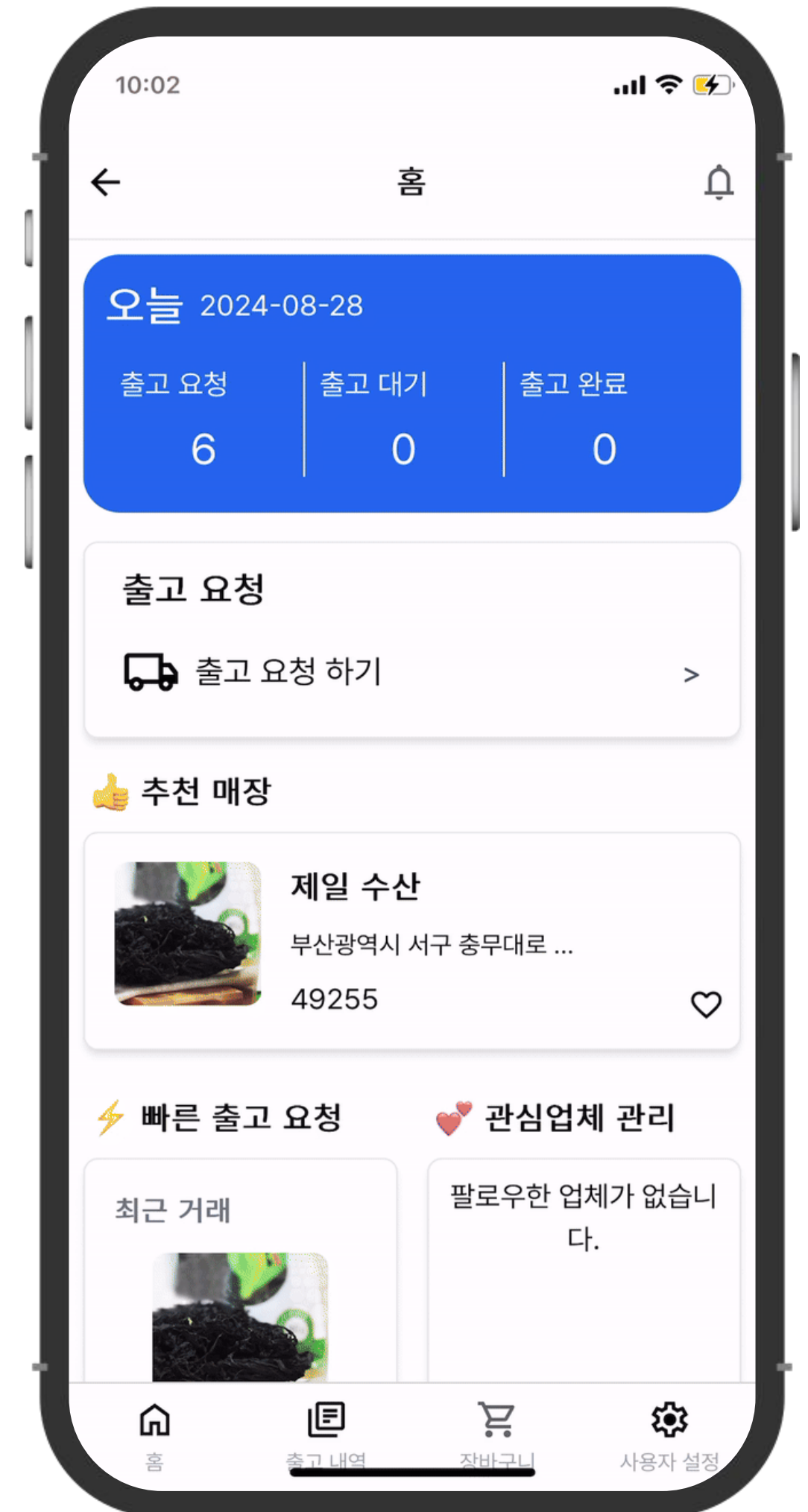
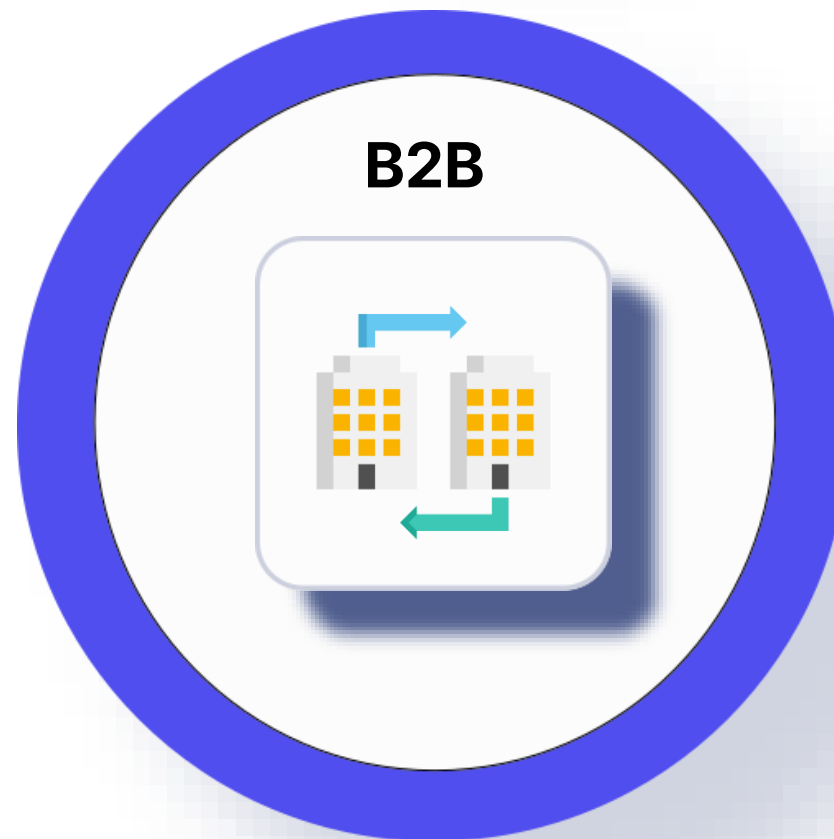
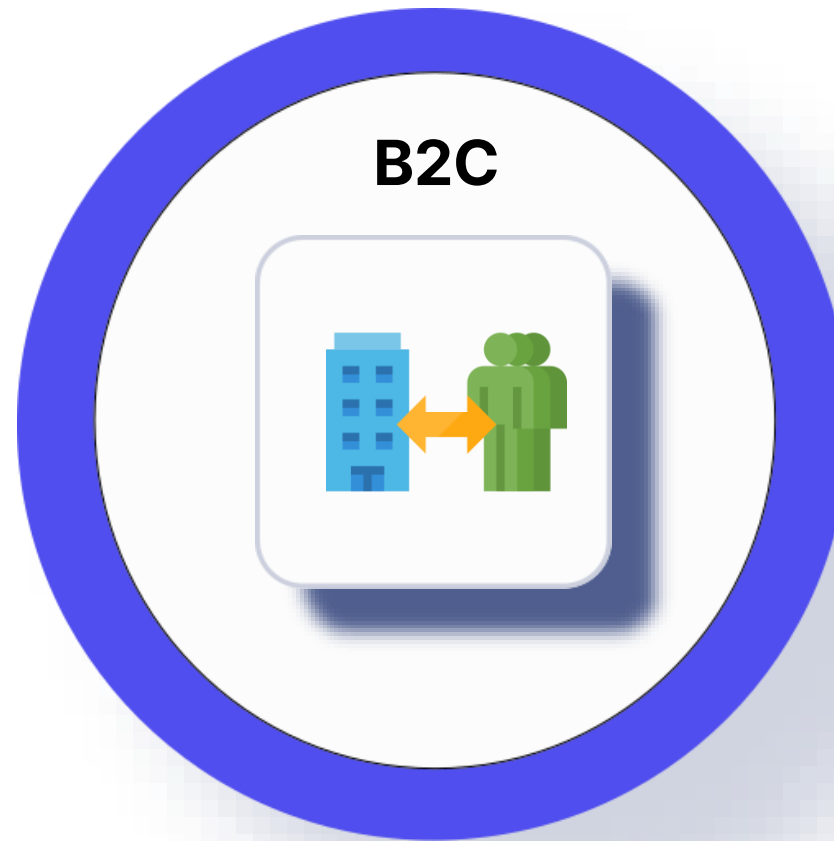
입고 / 출고 / 재고 관리





여보소:Seahere

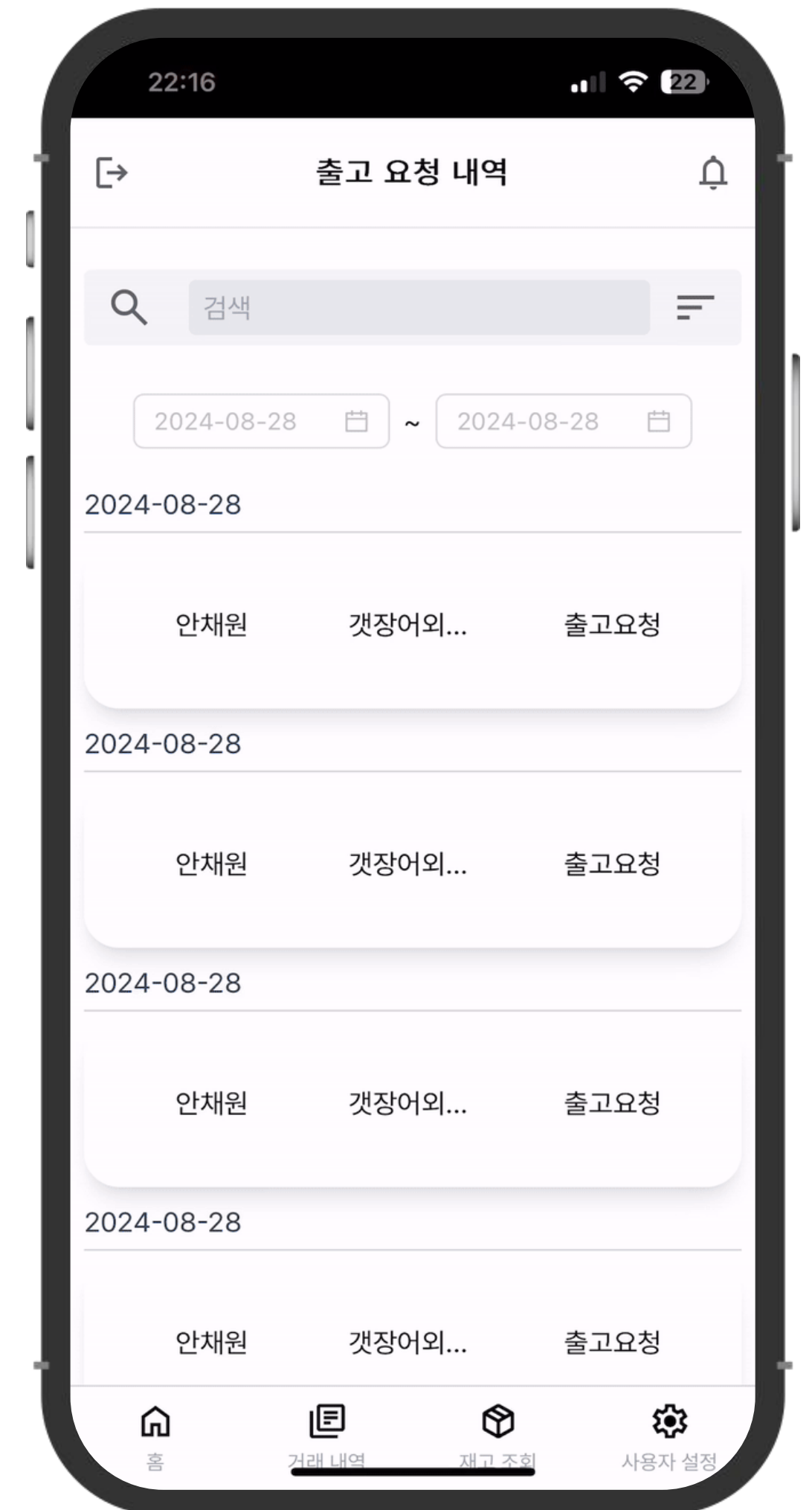
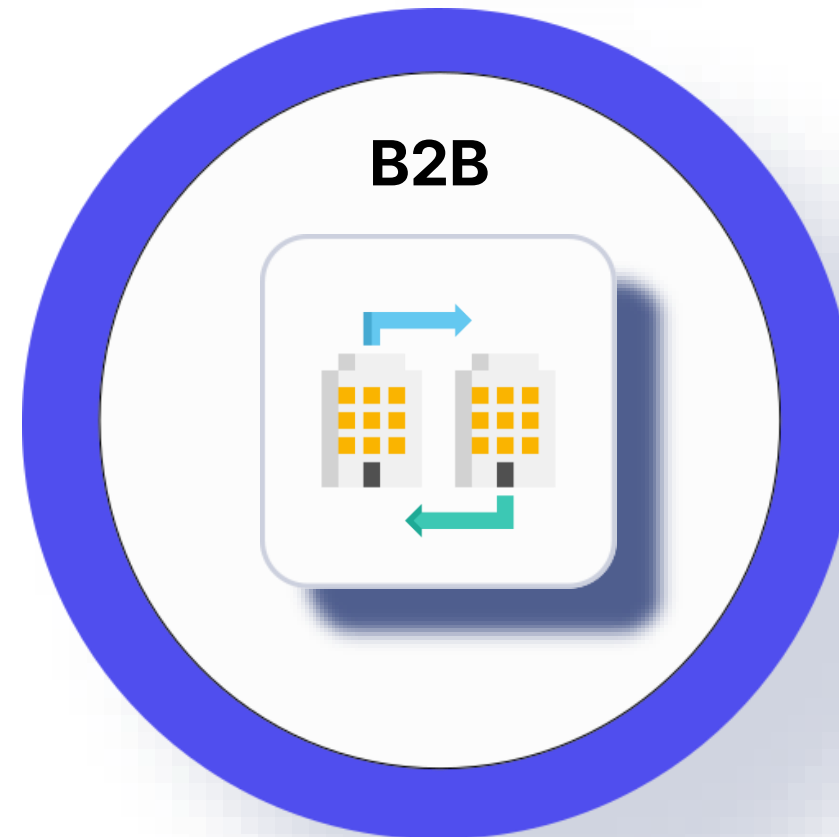
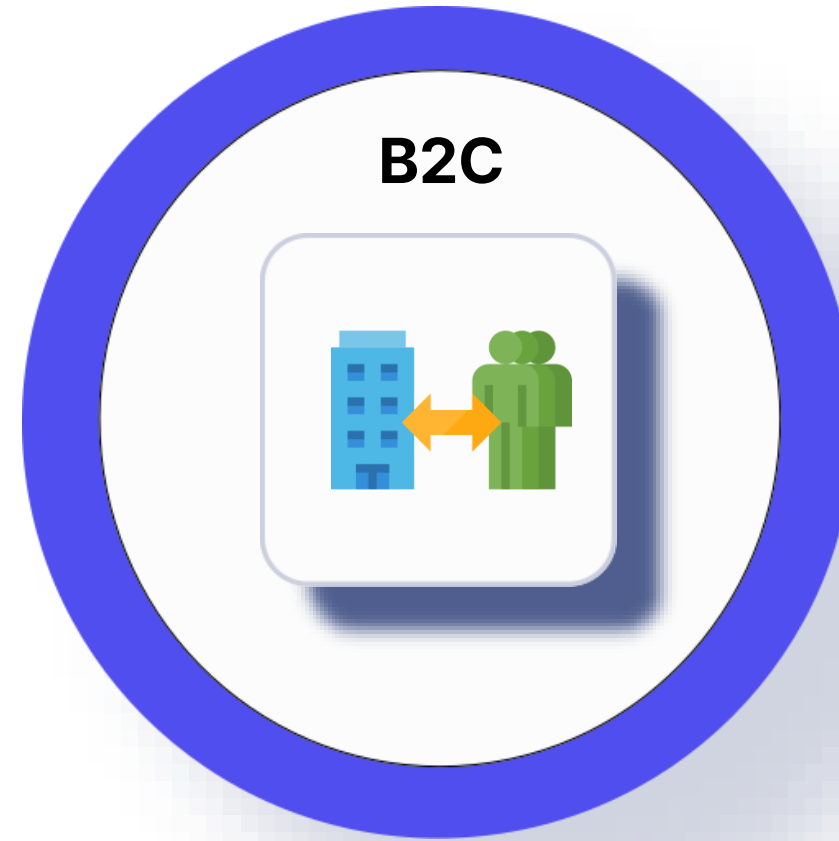
사용자 간 거래 요청





여보소:Seahere

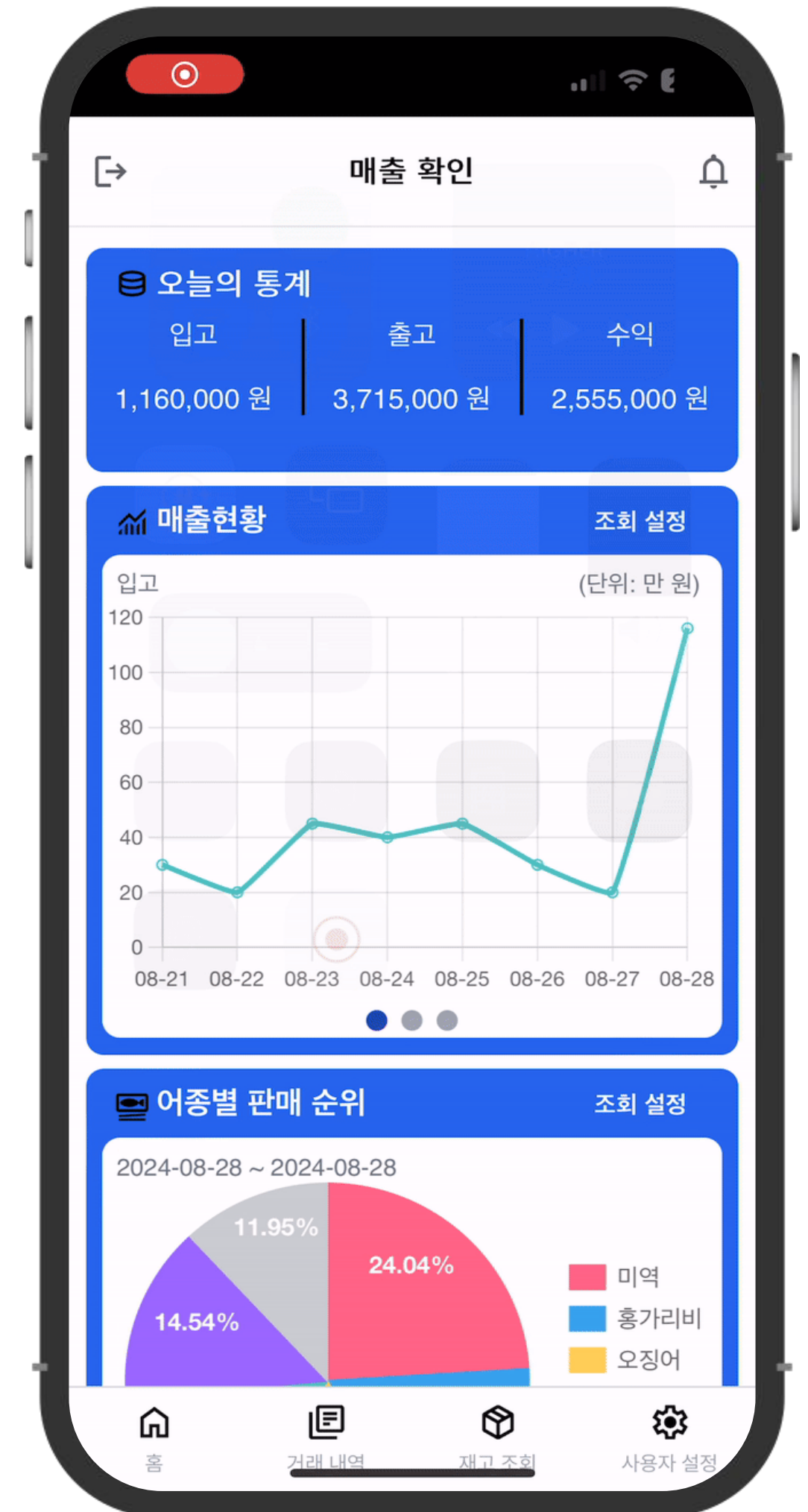
사용자 간 거래 요청





여보소 : Seahere

온라인 장부를 통한 실시간 통계

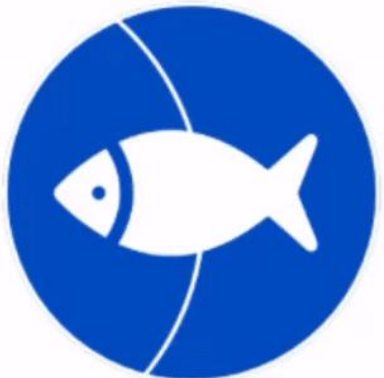





여보소:Seahere

사업자 등록증 스캔 & 푸시 알림

22:40 28



여보소
FISH BOX


카메라 인식

회사 이름

대표자 성명

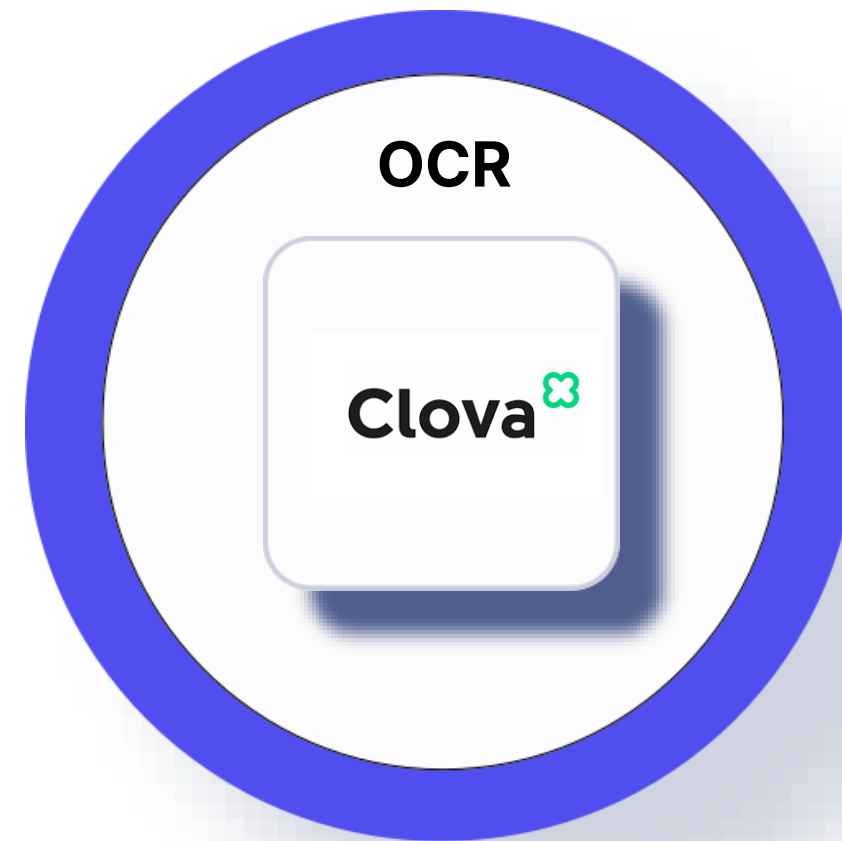
사업자 등록번호

주소 입력

상세 주소

개업 일자

진위 여부 확인



사업자 등록증 인증

Open API를 통한 진위 여부 확인

홈

오늘 2024-08-28

총거래	입고	출고
9	7	2

여데이~ 재고

🔍 검색

들어왔습니데이~ 입고

🔍 검색

🚚 나갑니데이~ 출고 요청 확인

도와달레이 도움 요청

다 팔렸데이 매출 확인

HELP

홈 거래 내역 재고 조회 사용자 설정

알림 서비스

FCM을 통한 사용자 알림





아날로그 전산화



시장 커뮤니티 활성화



QR코드를 통한 입고
프로세스 간소화



데이터 신뢰성 확보



효율적인 매출관리



김동

- FE/BE/Infra

로그인/회원가입
커스터머 출고
커스터머 설정
AWS 배포



김동

- FE/BE

수산물 입고
입/출고/수익 통계
어종 통계
트레이드



김동

- FE/BE

수산물 출고
브로커 알람
브로커 설정



김동

- FE/BE

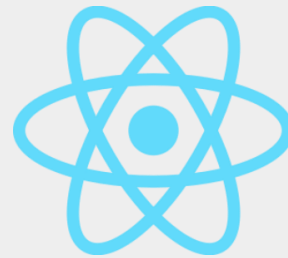
수산물 재고 관리
OCR
QR
회사 팔로우
S3 이미지 업로드



개발 진행-기술 스택 (Front - End)



JavaScript



React



Tanstack Query



Axios



Zustand



Firebase



Tailwind



WebPack



Vite



개발 진행-기술 스택 (Back - End)



Java 11



Gradle



Spring Boot



Spring Data JPA



Spring Security



JUnit 5



MySQL



Redis



H2



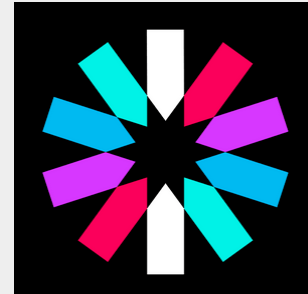
Slf4j



Query DSL



OAuth 2.0



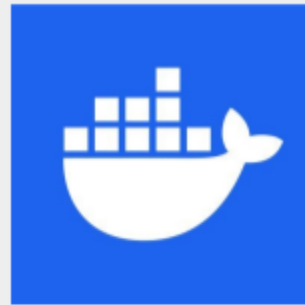
JWT



개발 진행-기술 스택 (CI/CD)



Nginx



docker



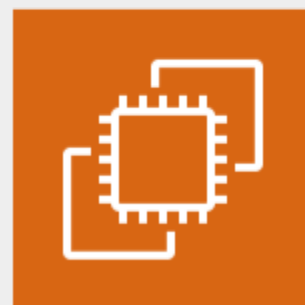
AWS ACM



Route 53



AWS S3



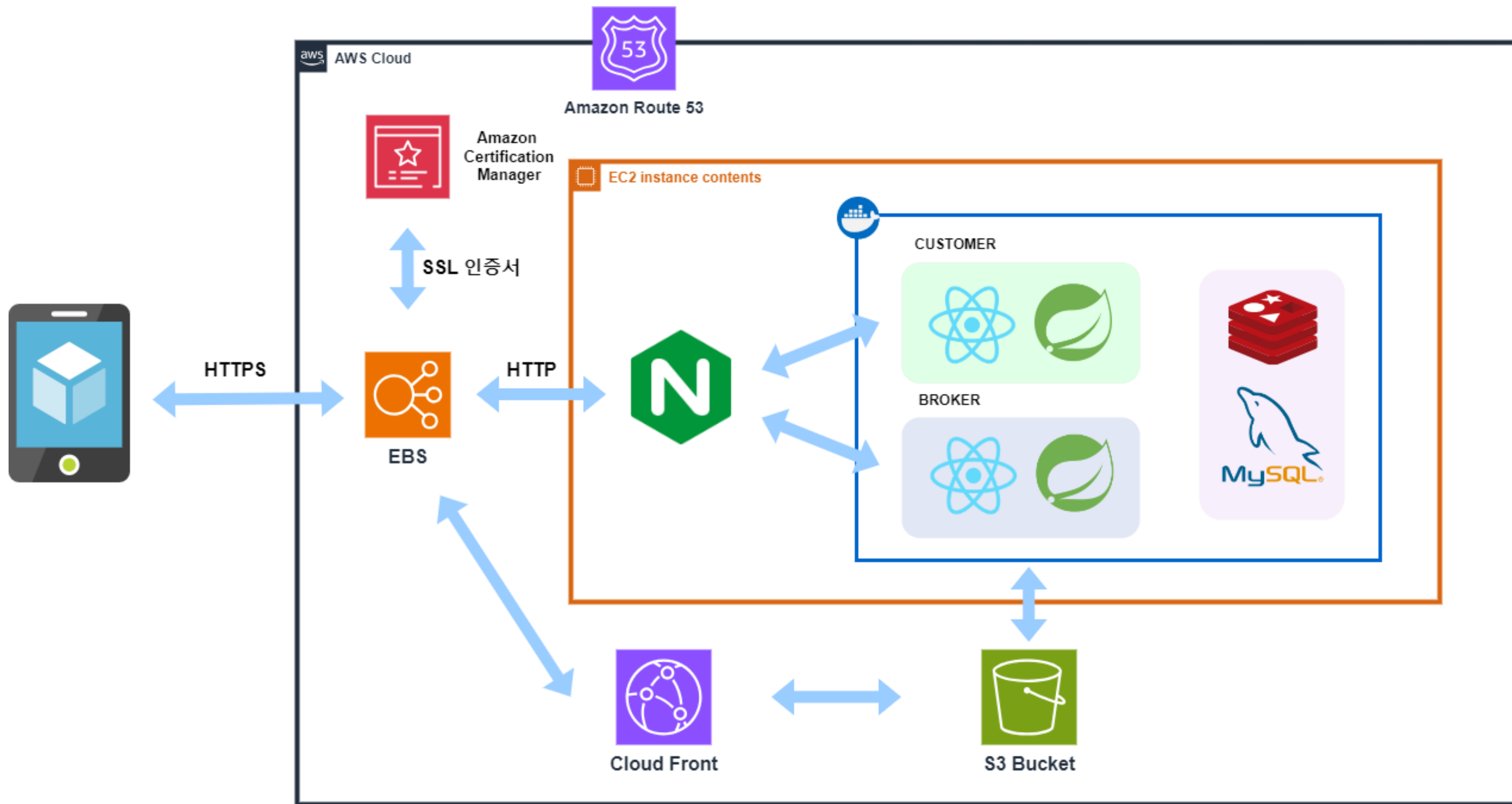
ASW EC2

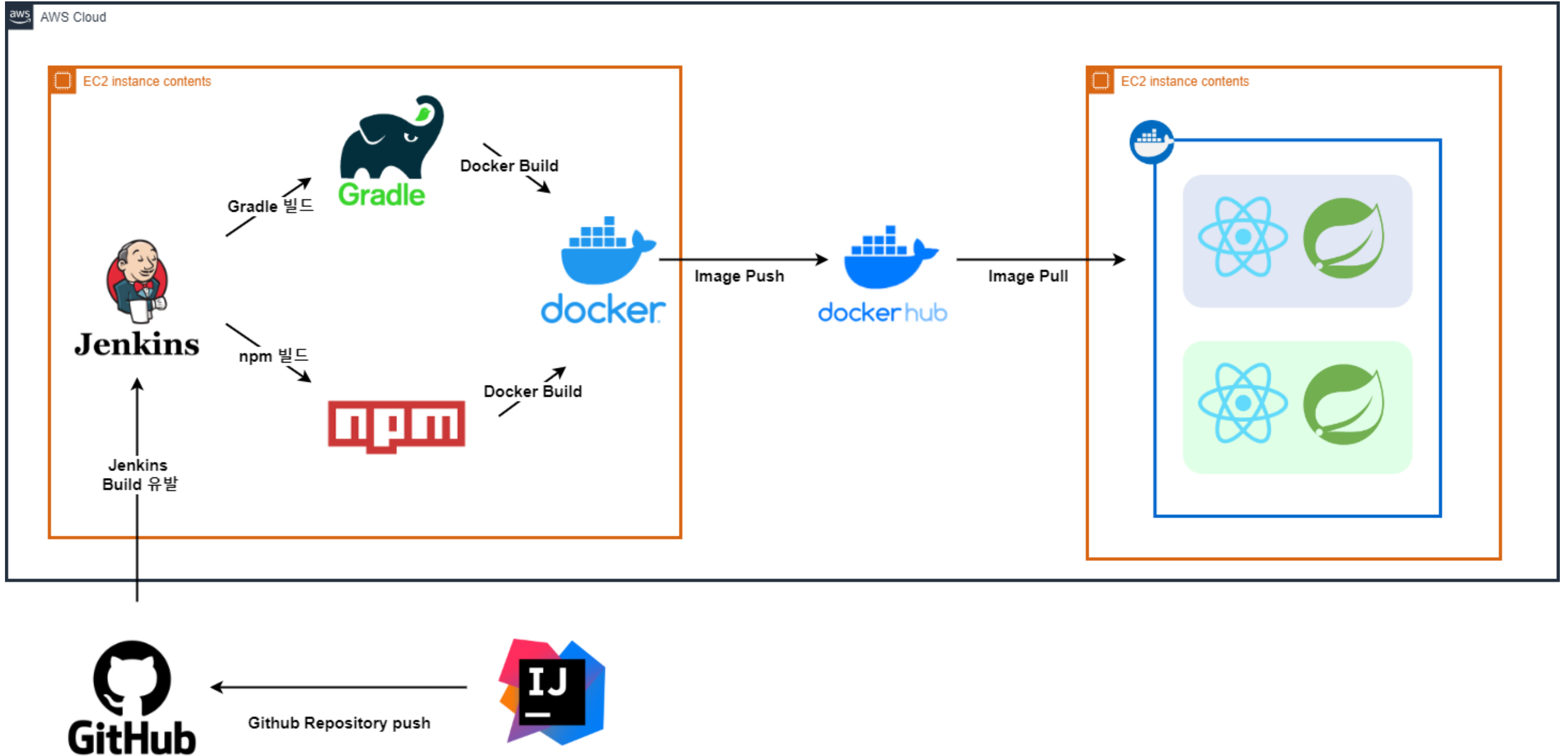


AWS ALB



Cloud Front







GROUND RULE

소유자 인증 태그 ...
 🧑 도영 김 Empty Empty

1. 지각 하지 않는다

지각하면 점심 커피 사기

2. 금요일 9 to 6 지키기

금요일은 야근 강요 없는 날

3. “아니 근데” 금지

반론은 다 듣고 하기

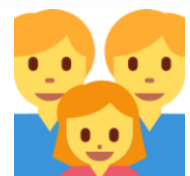
공감 후 반론 진행 하기

4. 36시간 전 개인 변동 사항 공유

최소 36시간 전에는 필요한 개인 일정 변동 사항 공유

5. Postive Emotion

긍정적인 마인드로 프로젝트에 임하기



Ground Rule

팔보채 팀 노션

☰ 홈 ▾

🔍 검색

TEAM

👤 MEMBER

💬 오늘의 ...

🔥 프론트 ...

🧐 백 지식 ...

📁 회의록

👥 GROUN...

🗣 팀 로고

PROJECT

👤 스프린트

⚠ 트러블 슈팅

🔴 Git Conven...

📄 Develop C...

🔍 전체 아키...

1 📄 페이지 1

2 📄 페이지 2

📈 중간 발표 ...

DOCS

♥ 깃허브

♥ ERDCloud

★ 요구 사항 ...

📖 API 명세

📖 API 명세서 ...

📄 제출 문서

🌈 figma

🔗 drive.google.com

🌈 8조(김도영, 정기 석, 안채원, 장호영) - Google Drive

33	+	//todo 추후에 user도 entity로 바꿀것
31	34	@Autowired
32	35	private OutgoingService outgoingService;
33	36	@Autowired
34	37	private OutgoingDetailJpaRepository outgoingDe
35	38	@Autowired
39	+	private CompanyRepository companyRepository;
40	+	@Autowired
36	41	private EntityManager em;
42	+	@Autowired
43	+	private OutgoingJpaRepository outgoingJpaRepos
37	44	
38	45	@PersistenceUnit
39	46	EntityManagerFactory emf;
47	+	
48	+	private Long companyId;
49	+	private Long outgoingId;
40	50	@BeforeEach
41	51	void init(){

kimdodo97 2 weeks ago
 초기 데이터 넣는 부분을 IntStream을 이용한 반복 처리 또는 SQL 파일을 활용해서 좀 더 가독성을 높이는건 어떨까요?
 😊
 Reply...



Code Review

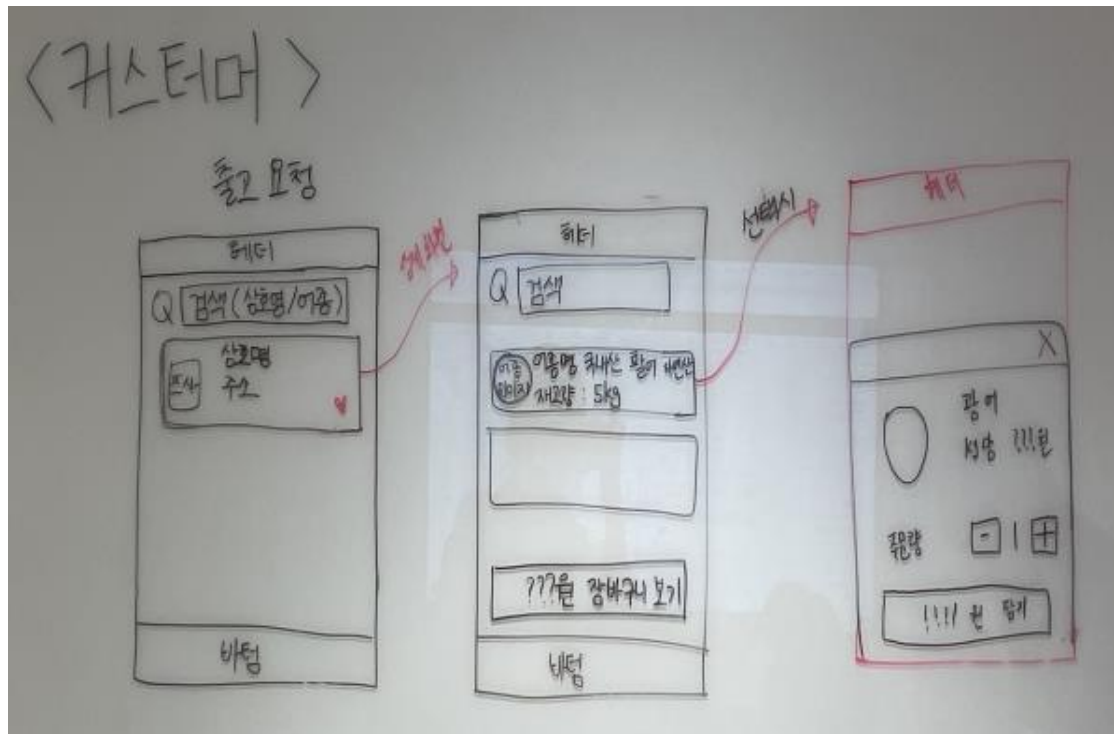


Notion & Slack



개발 진행-초기 회의

와이어 프레임

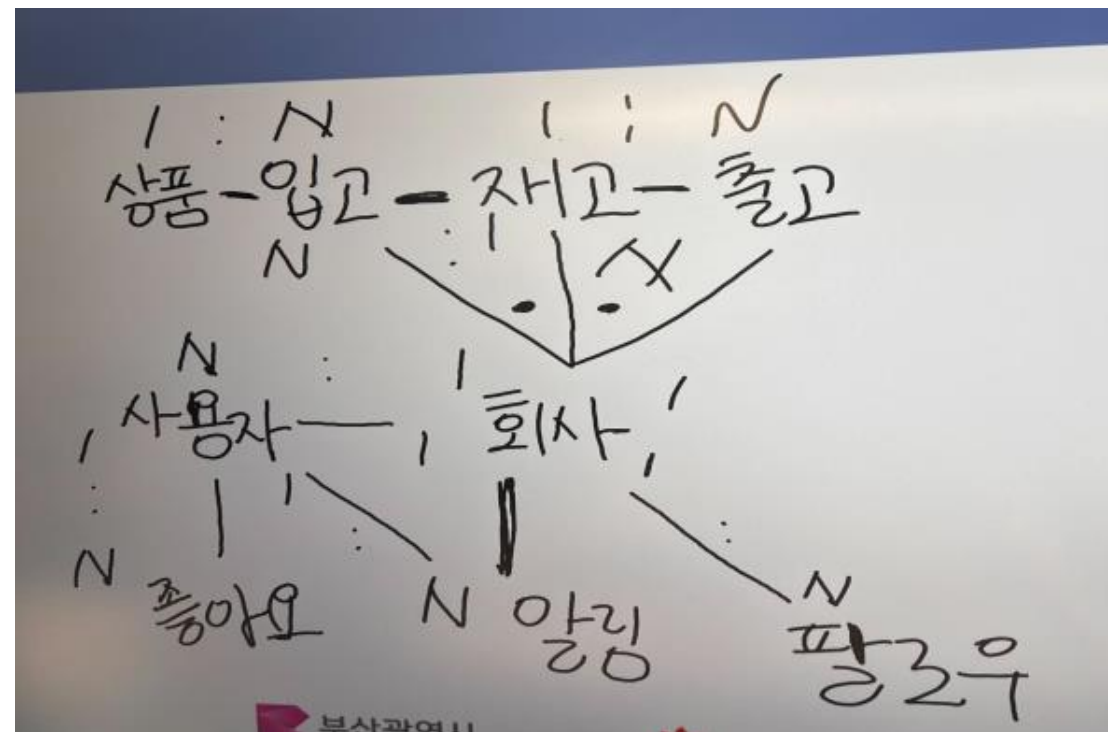
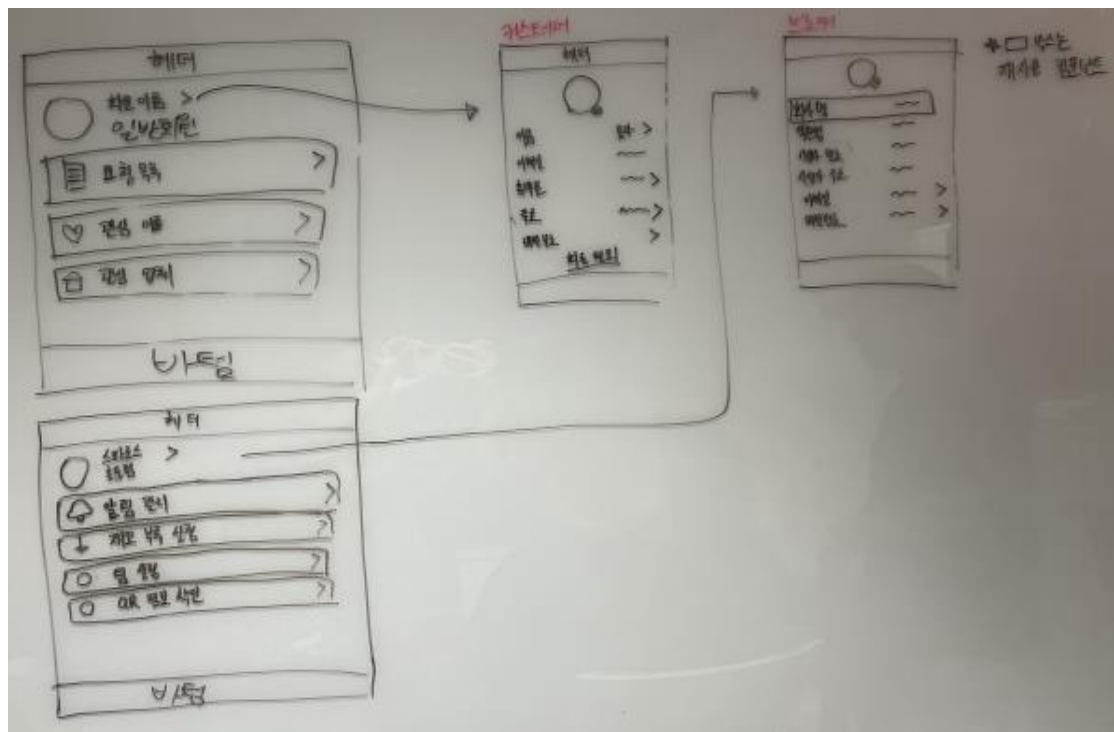


DB 설계

회사	사용자	상품	입고	재고
Company	USER	Product	Store	Inventory
ID	ID	ID	ID	ID
사명/자음	Email	Name	상품ID	상품ID
번호	PW	Category	유저ID	입고일
Name	Roll	Country	입고량	유통기한
	회사ID	Natural	입고일	재고량
	OAuth2.0	Qr	회사ID	입고금액
				재고금액
				회사ID

기능 설계

앱 : 로그인, 회원가입, 입고, 출고
재고, 팔로우, 히스토리, 알림내역,
B : 로그인, 회원가입, 직원등록, 입고
출고, 온라인장부, QR코드, 히스토리
재고, 알림내역, 팔로우,
직원등록, 히스토리

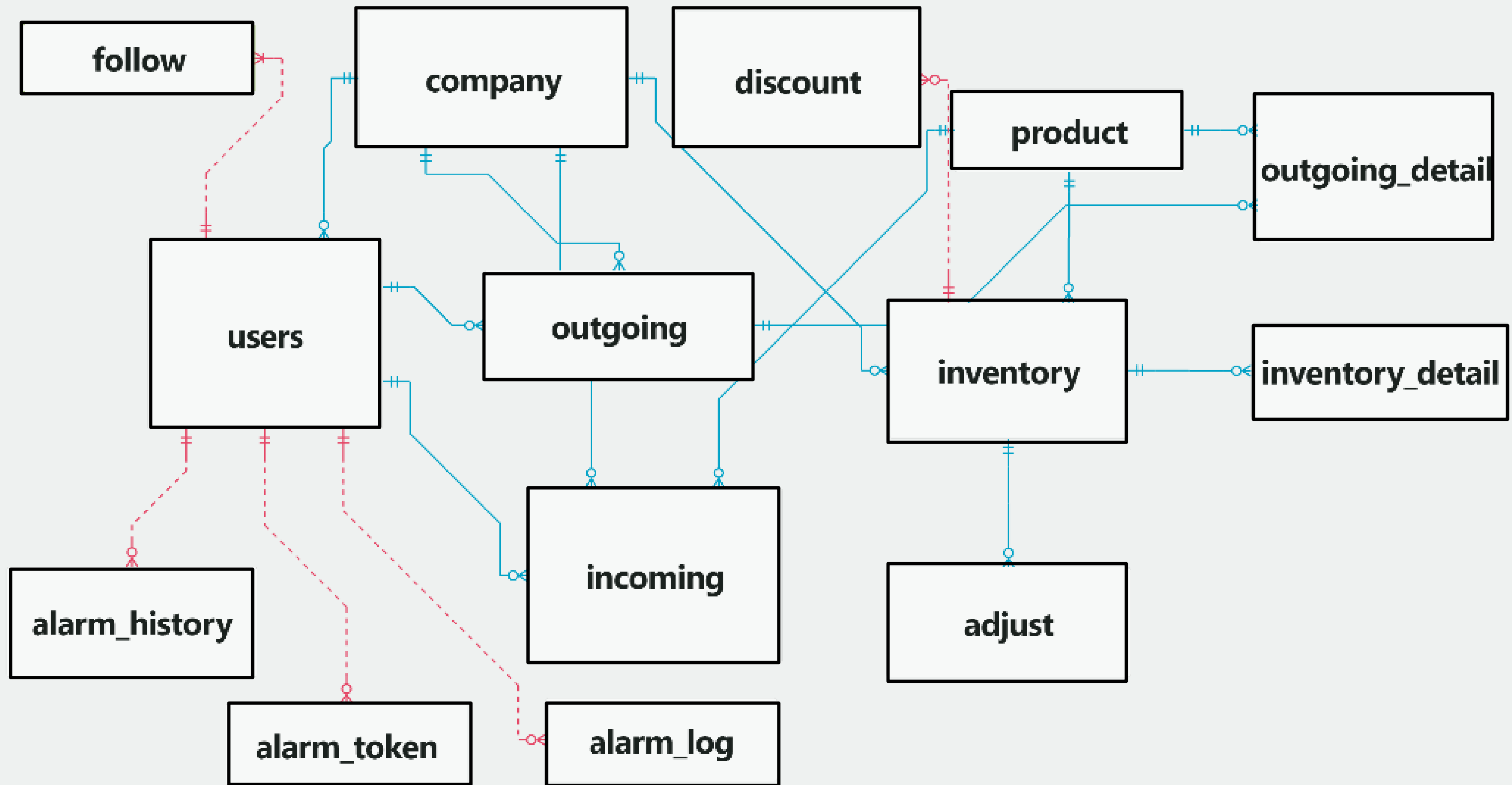


종양 : Broker
고객 : Customer
입고 : Incoming
출고 : Outgoing
재고 : Inventory
조정 : adjust
직원 : employee
회사 : company
알림 : message
상품 : product

종양 : like
팔로우 : follow
종양 : B2C
종양 : B2B
종양 : + like
종양 : find ~
종양 : add ~
종양 : create ~
종양 : delete
종양 : update

직원 역할 (ROLE) : Staff
관리자 역할 : Manager
종양 : Not Found (404)
종양 : Empty
종양 : reject
종양 : login
종양 : logout
종양 : Ready
종양 : Complete
종양 : Search

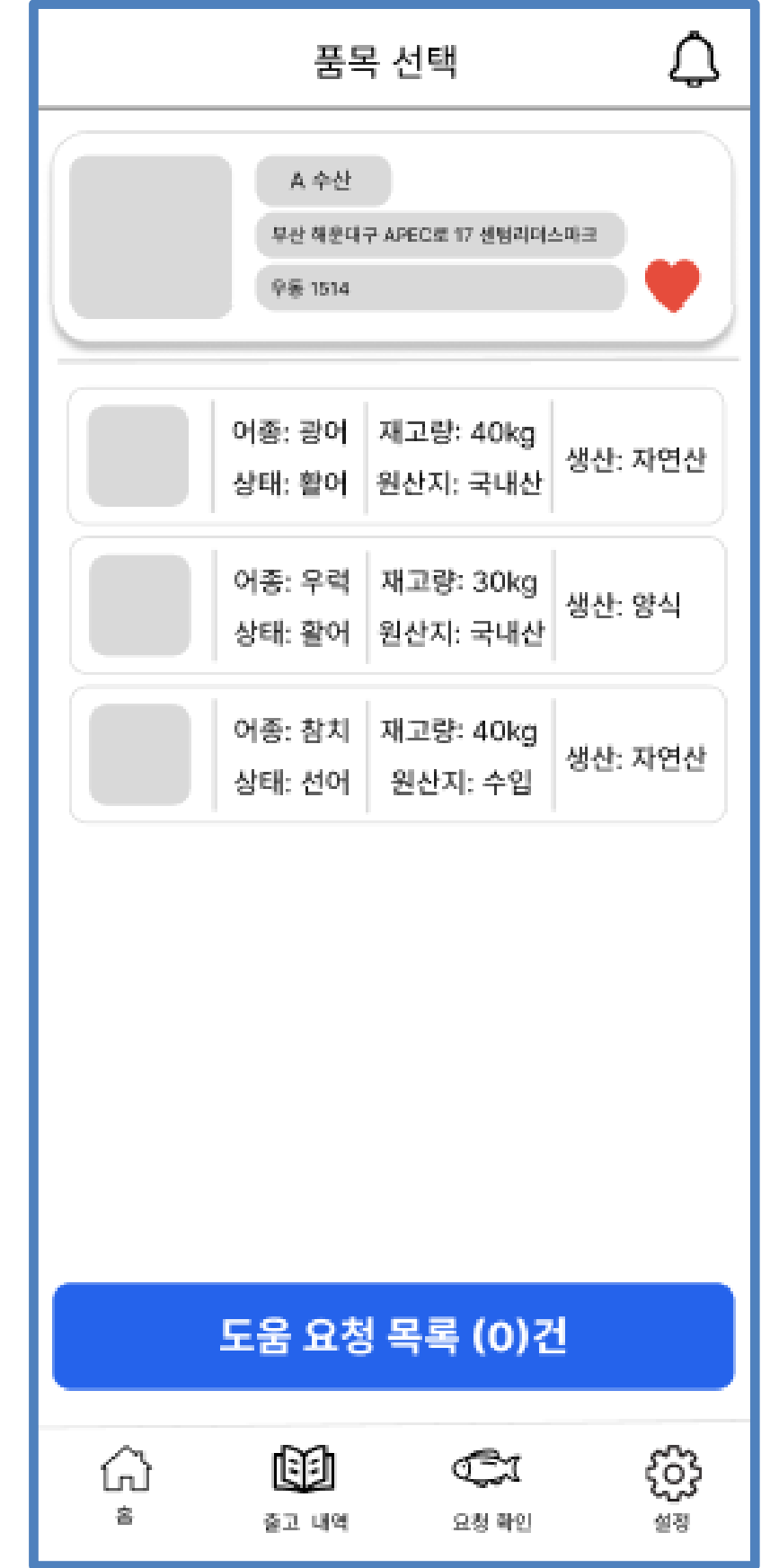
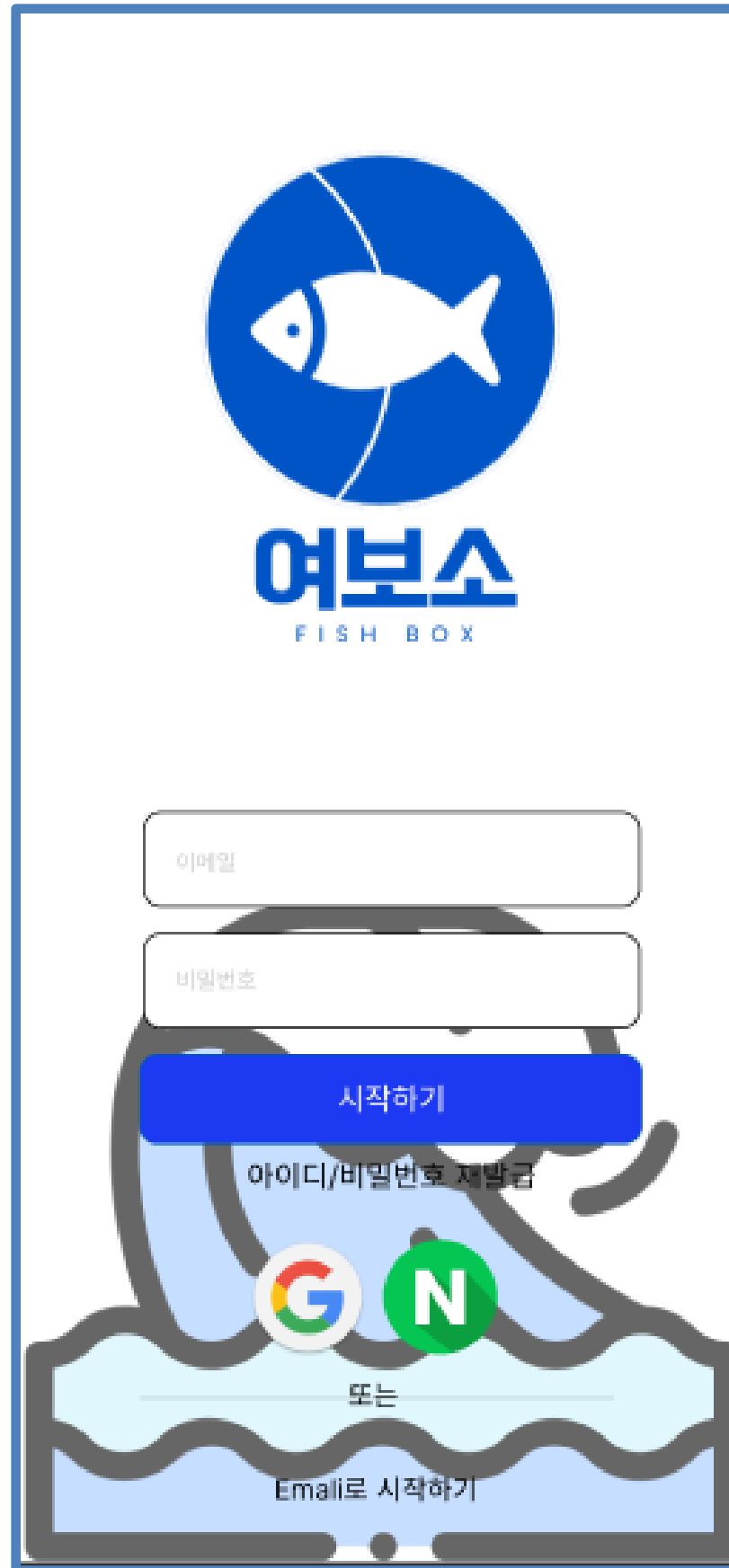
종양 : live (생존)
종양 : dead (사망)
종양 : frozen (냉동)
종양 : abroad (해외)
종양 : local (국내)







개발 진행 - Figma





01

"우리프로젝트의정체성"

수산업체를위한수산시장특화유통관리시스템

02

"프로젝트의확장가능성"

비수산업자들을 위한서비스확장

03

"배포및성능개선"

어플리케이션AWS배포및성능개선

페이즈 1

서비스 핵심 정체성을 보유한 MVP 개발
로그인 회원 인증, 입고/출고, 재고 수정 기능을 보유한 핵심 서비스 개발

페이즈 2

핵심 기능 기반 사업성 및 사용자 편의성 확장 개발
B2C, B2B 거래 기능, 온라인 장부, QR/OCR 등의 확장 기능 개발

페이즈 3

개발된 서비스 기반 AWS 배포 및 성능 개선 개발
Redis 동시성 및 Cache 기능, AWS를 통한 클라우드 배포, CI/CD 기능 개발



7월

07월 15일 ~ 07월 20일
- 기획 및 주제 선정
- 자료 조사
- 요구사항 작성

07월 22일 ~ 07월 24일
- 와이어 프레임 제작
- DB설계
- API 정의

07월 25일 ~ 08월 06일
- 페이즈 1

SUN	MON	TUE	WED	THU	FRI	SAT
	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	31			



8월

07월 25일 ~ 08월 06일
- 페이지 1

08월 07일 ~ 08월 19일
- 페이지 2

08월 20일 ~ 08월 27일
- 페이지 3

08월 28일 ~ 08월 31일
- 산출물 정리
- 최종 보고서 작성
- 최종 발표

SUN	MON	TUE	WED	THU	FRI	SAT
				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	31



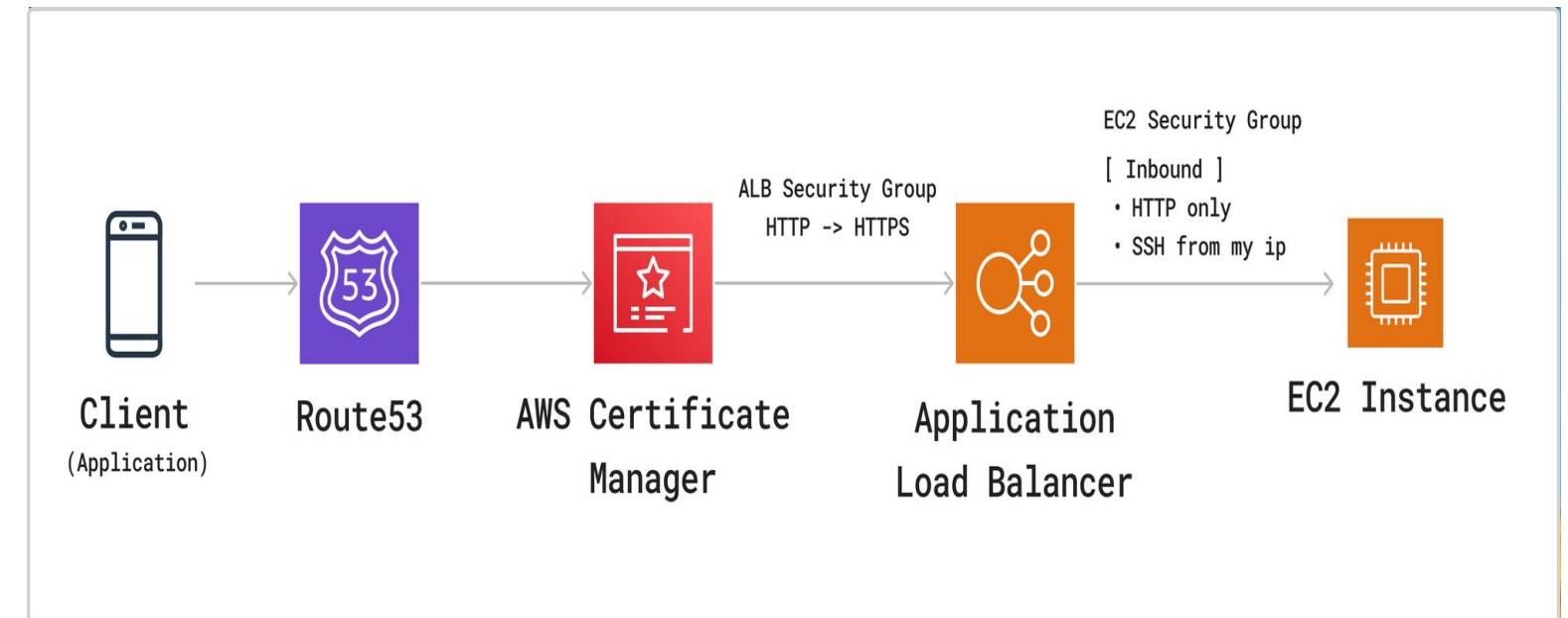
트러블 슈팅 1: HTTPS

❌ ▶ Mixed Content: The page at ' [brokerCertificationApi.js:18
https://cf45-58-235-119-39.ngrok-free.app/signup/broker](https://cf45-58-235-119-39.ngrok-free.app/signup/broker)' was loaded over HTTPS, but requested an insecure XMLHttpRequest endpoint ' <http://api.odcloud.kr/api/nts-businessman/v1/validate?servicekey=%2...>'. This request has been blocked; the content must be served over HTTPS.

배포 전 발생한 Mix Content 문제

- HTTPS 프론트에서 HTTP 백엔드 서버에 OCR 통신 요청
- > 백엔드 서버가 HTTP이기에 Mix Content 에러 발생
- > 임시로 Ngrok을 사용하여 HTTPS 사용

Mix Content : 웹 페이지가 HTTPS를 사용하면서 HTTP 리소스를 로드하려고 할 때 발생하는 문제



배포 후 문제 해결

- Route 53 를 통해서 도메인 구매 후 ACM 적용
- > AWS ALB를 통한 HTTPS 요청 수신
- > ALB 에서 Nginx 로 요청 수신 후 리버스 프록시
- > 컨테이너간 HTTP 통신 / 브라우저 - ALB HTTPS 통신

해결 방법 : 도메인에 ACM을 통한 SSL/TLS 인증서 적용 후 ALB를 통한 EC2 요청 전달로 해결



트러블 슈팅 2: 동시성

```
org.opentest4j.AssertionFailedError:  
expected: 20.0f  
but was: 24.0f  
Expected :20.0f  
Actual   :24.0f  
<Click to see difference>
```

동시성 제어 적용 전 동시 출고 테스트

커스터머들의 동시 출고 요청

- > 출고 감소 데이터와 남아 있는 재고 데이터가 불일치
- > 경쟁 상태가 발생하여 예상과 다른 결과가 나온다

경쟁상태 : 공유 자원에 대해 여러 프로세스가 동시에 접근을 시도할 때,
타이밍이나 순서 등이 결과값에 영향을 줄 수 있는 상태



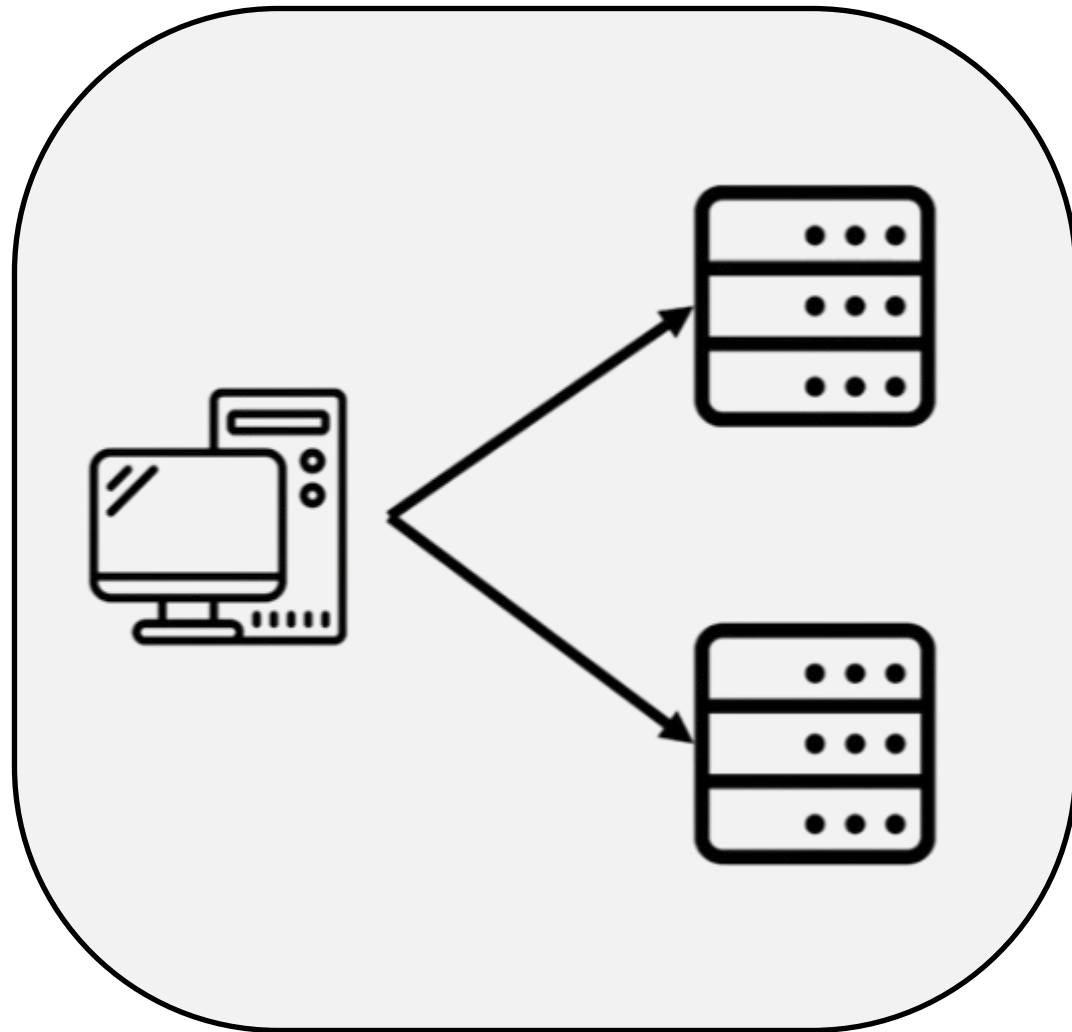
```
127.0.0.1:6379> keys *  
(empty array)  
127.0.0.1:6379> keys *  
1) "1outgoing"  
127.0.0.1:6379> keys *  
1) "1outgoing"  
127.0.0.1:6379> keys *  
(empty array)  
127.0.0.1:6379> keys *  
1) "1outgoing"  
127.0.0.1:6379> keys *  
(empty array)
```

동시성 제어 적용 후 SETNX를 이용 동시성 제어

Redis의 Lettuce 방식을 활용

- > 입,출고시에 동시성제어
- > Redis는 싱글 스레드를 사용하여 스레드 안전
- > 필요한 비즈니스 로직에 퍼사드 패턴을 적용

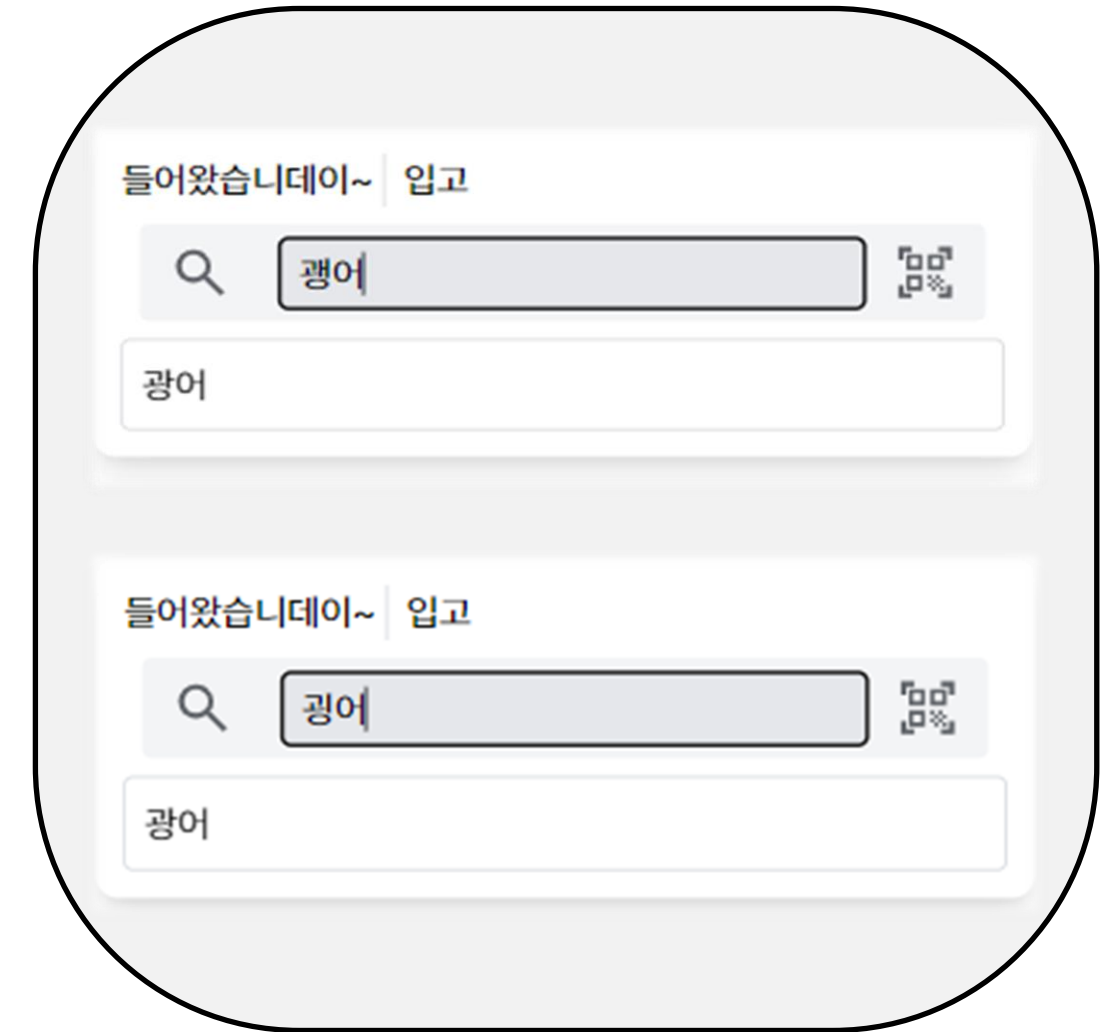
해결 방법 : 이미 사용하는 key가 있다면 대기하여 key를 생성 후 진행,
이후 키 삭제



무중단 배포



부하 테스트



Elasticsearch



개발 후기:느낀 점



김동

- 여보소 라는 프로젝트를 진행하면서 단순 개발 뿐만 아니라 기획과 인프라 구성을 해보면서 개발자로 조금 더 성장한 거 같습니다. 무중단 배포, 부하 테스트와 같은 다양한 시도를 해보려 했지만 하지 못한 부분이 아쉽기도 합니다. 2달이라는 시간 동안 좋은 팀원들과 웃으며 재미있게 개발을 한 거 같아서 팀원 분들에게 너무 감사드립니다. 그리고 항상 고생하시는 준범님 감사합니다!



김동

- 이번 프로젝트를 통해 기획부터 배포까지의 모든 과정을 직접 경험하면서 정말 제대로 된 프로젝트를 진행하는 것 같았습니다. 프론트엔드 지식이 전혀 없었던 저에게는 풀스택 개발이 큰 도전이었지만, 이를 통해 개발자로서 한걸음 더 성장할 수 있었습니다. 또한, 이번 프로젝트는 단순히 기술적인 측면뿐만 아니라, 팀원들과의 협업, 문제 해결 능력, 그리고 프로젝트 관리 역량을 키우는 데에도 큰 도움이 되었습니다. 살면서 이런 팀원들을 만날 수 있을까?라는 생각이 들 정도로 합이 너무 잘 맞아서 두 달 동안 힘들지만 행복했습니다.



김동

- 많은 것을 경험했던 의미 있는 시간이었습니다. 처음으로 API 통신을 사용하는 풀스택 개발을 진행하면서 프로젝트 초기에 어떻게 API를 정의해야 하는지를 실제로 경험할 수 있었습니다.또한 그동안 어렵קות이 알고만 있던 Querydsl과 Redis를 실제 프로젝트에 적용해 보면서 많은 공부가 되었습니다. 아쉬운 점으로는 일정이 촉박하다 보니 프로젝트의 퀄리티를 충분히 높이지 못한 점과 제가 맡은 역할을 완벽히 수행하지 못한 점이 특히 아쉽습니다. 그럼에도 불구하고 좋은 팀원들과 함께할 수 있어서 즐거운 경험이었습니다



김동

- 처음 프로젝트를 시작할 때만 해도 '과연 할 수 있을까? 내가 다른 팀원들에게 짐이 되지 않을까?' 하는 생각도 있었다. 하지만 팀원들 덕분에 경험해보지 못한 일에 대한 불안과 걱정을 덜어낼 수 있었고 개발자로서 한걸음 나아갈 수 있었다. 서로가 서로에게 웃음을 주는 팀. 어려운 일은 같이 해결하고자 하는 팀. 유난히 덥고 습한 여름이었지만 얼굴만 봐도 웃음이 나오는 팀 덕분에 건물에 들어서고 나면 팀원들 얼굴이 보고 싶다는 생각부터 들었다. 이보다 더 좋은 팀원들을 만날 수 있을까... 24년의 여름은 잊히지 못할 것 같다.



QnA

