

# SROA

Schedule Reserve Optimization Algorithm

팀명 : 한잔하조

발표 : 신재영

멘토 : 서영종

팀원 : 박상환 서상범 신재영 장주언

# 목차

01

팀원 소개

02

프로젝트 개요

03

프로젝트 구성

04

시연영상

05

후기

06

Q&A

# 01

## 팀원 소개



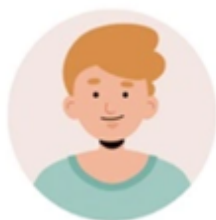
## 팀원소개



서영종  
멘토



박상환  
고객 어플리케  
이션



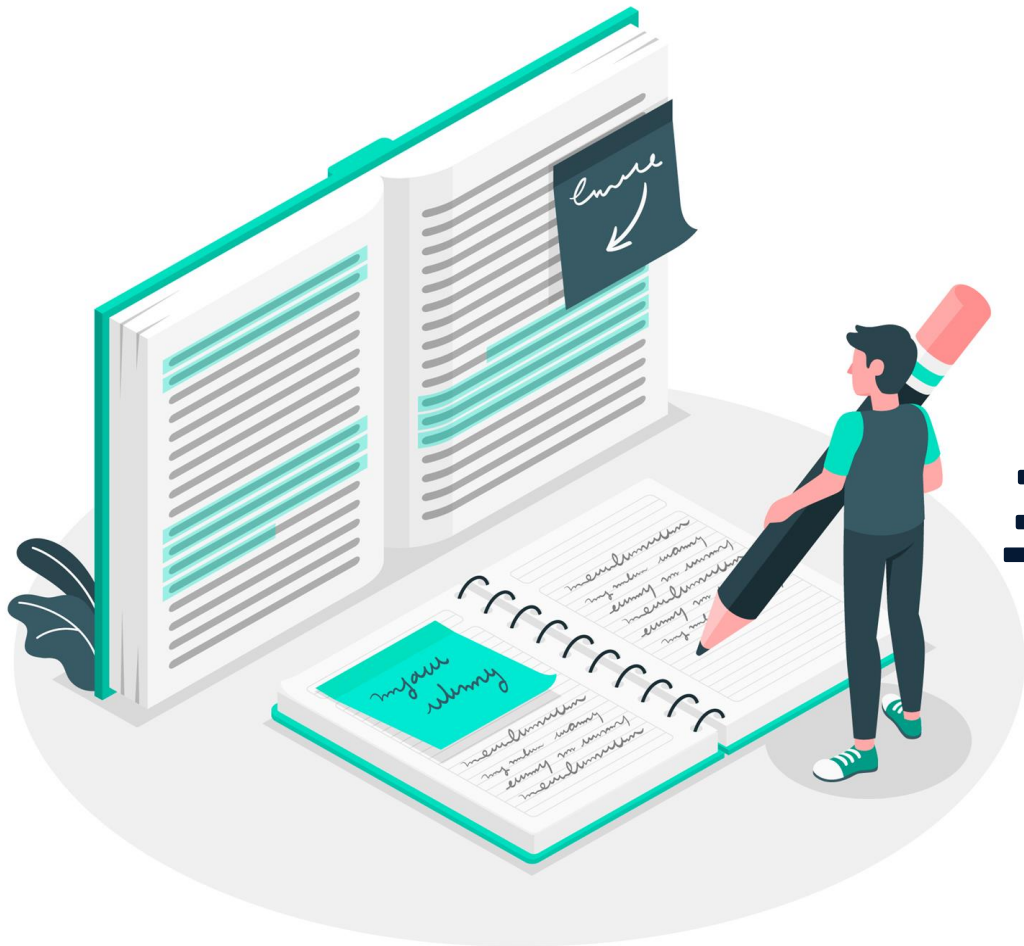
서상범  
엔지니어 어플  
리케이션



신재영  
AWS 인프라  
설계/구축



장주언  
MSA  
서버개발



# 02

## 프로젝트 개요

# 점검·수리·응대...하얏게 태워도 남는 건 실적 압박

◇실적 압박, 감정노동도 심해 = 과노동에 시달리는 이유는 '실적 압박' 때문이다. 응답자 40.7%가 목표량 달성 압박이 매우 심하다고 답했으며, 42.5%가 약간 심하다고 했다. 절반가량이 목표량을 채워도 고용이 불안한 처지에 놓여 있다고 스스로를 인식했다. 소득 수준도 낮다. 생활이 불가능한 소득이라는 데 89.6%가 동의했다.

[출처] <http://www.idomin.com/news/articleView.html?idxno=777841>

엔지니어의 업무에 대한 **실적 압박**

# 현재시장

본사에서 평가하는 서비스 실적을 채워야 하는데, 원하는 고객이 없을 경우엔 고의로 고장을 내서라도 가야만 했습니다.

<녹취> "(고장나면) 고객님들이 전화를 해요. 바로 방문하면 의심할 수 있어서 한 30분 있다가 (가는 거죠.)"

수리기사들은 실적을 못채우면 월급이 깎이기 때문에 이런 일이 전국 협력업체마다 공공연하다고 말합니다.

<녹취> "(실적 못 맞추면) 한달에 차감되는 금액이 적게는 10-20만 원에서 많게는 50-60만 원..."

[출처] <https://news.kbs.co.kr/news/view.do?ncd=3022409>



금속노조 삼성전자서비스지회 조합원들이 9월 30일 오전 서울 중구 황동 금속노조 회의실에서 연 '삼성전자서비스 가전수리 노동자 사망사고 규탄 기자회견'에 참석해 손팻말을 들고 한 노동자를 사망하게 한 삼성전자서비스를 규탄하고 있다. 연합뉴스

금속노조는 사고 주요 원인으로 삼성전자서비스 사측의 실적 압박을 꼽았다. 삼성전자서비스는 처리 건수와 수리 완료 정도를 따져 노동자마다 등급을 매기고 전급 여부를 결정한다. 윤 조합원과 같은 센터 소속인 김문석 삼성전자서비스서울지회 양천분회장은 "센터는 노동자 건강과 안전은 뒷전이고 항상 처리 건수를 빨리, 많이 높이는 일만 신경 쓴다"라며 "삼성이 만든 시스템 때문에 승환이가 죽었다"라며 울먹였다.

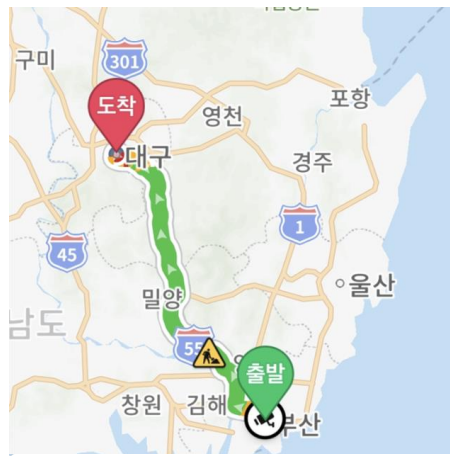
[출처]  
<http://worknworld.kctu.org/news/articleView.html?idxno=404222>

# 현재시장

## 하루 평균 이동 거리



[출처] <http://www.idomin.com/news/articleView.html?idxno=777841>



5.5km 짧은

1시간 48분 | 106km

요금정보 택시비 98,830원  
동행료 5,800원 | 연료비 16,591원  
중앙고속도로(부산-대구)



고객중심의 서비스로 인한 **비효율적인 엔지니어 동선**



# 해결 방법 및 기대효과

01

최근 한 달 동안의 업무 수를 비교하여 업무 수가 적은 엔지니어에게 우선순위를 높게 부여

02

담당 지역 내 엔지니어들의 빈 일정에 고객이 신청, 일정 간 위치 반영

기대효과

## 다양한 방문 서비스 예약에 대한 최적화 솔루션

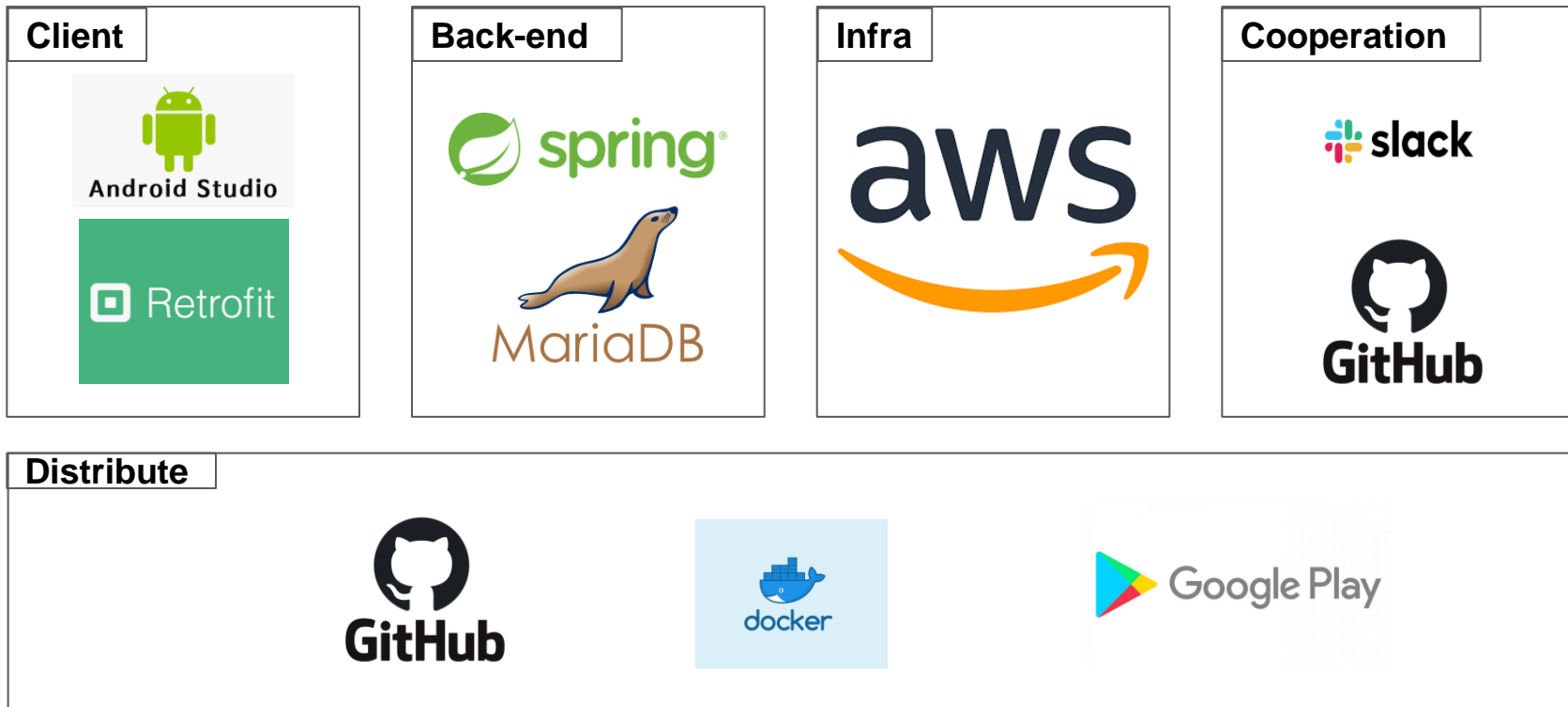
1. 균등한 업무 분배
2. 엔지니어들의 감정노동 감소
3. 효율적인 업무 배정
4. 기업의 인력 관리 및 이윤 최대화

# 03

## 프로젝트 구성



# 기술 스택



# 협업



Sinje ▾

Slack Connect

Slack 찾아보기

채널

# 랜덤

# 일반

# 최종프로젝트

+ 채널 추가

다이렉트 메시지

신재영 나

상환

서상범

장주언

+ 팀원 추가

# 최종프로젝트 ▾

주언

장주언 오후 12:00  
PPT 알고리즘 설명, 기존에 뭐 때문에 생각했는지 구체적으로

상환

상환 오후 12:01  
시나리오 백그라운드보완  
누구를 타겟으로 하는 서비스인지 정확한 설명이 필요

상환

상환 오후 12:09  
취지 목적 차별화 왜?

신재영

신재영 오전 11:24  
2021.10.26  
-----  
고객 app  
예약페이지 유저정보 받아와서 바로 출력  
예약 시간 클릭 이벤트 설정  
예약시 예약 데이터 받아오기 코딩  
현황조회 데이터 받아오기 코딩  
아이디 비밀번호 조회  
-----  
엔지니어 app  
홈 익점 클릭시 페이지 이동 구현



SROA-busan

Overview

Repositories

Packages

People

Teams

Projects

Type ▾

Language ▾

Sort ▾

New repository

SROA\_Account\_MSA (Public)

java 0 0 0 0 Updated 2 days ago

SROA\_ReservationSchedule\_MSA (Public)

java 0 0 0 0 Updated 2 days ago

SROA\_Repair\_MSA (Public)

java 0 0 0 0 Updated 2 days ago

SROA\_InquirySchedule\_MSA (Public)

java 0 0 0 0 Updated 2 days ago

SROA\_Evaluation\_MSA (Public)

java 0 0 0 0 Updated 2 days ago

SROA\_Manage\_MSA (Public)

java 0 0 0 0 Updated 2 days ago

Engineer-client (Public)

Kotlin 3 0 0 1 Updated 2 days ago

User-client (Public)

Kotlin 2 0 0 0 Updated 6 days ago

SROA-Server (Public)

java 1 0 0 0 Updated 16 days ago

# MSA Server



## MSA (MicroService Architecture)

각각의 서비스가 모듈화 되어있고 이러한 모듈끼리 API를 이용하여 통신

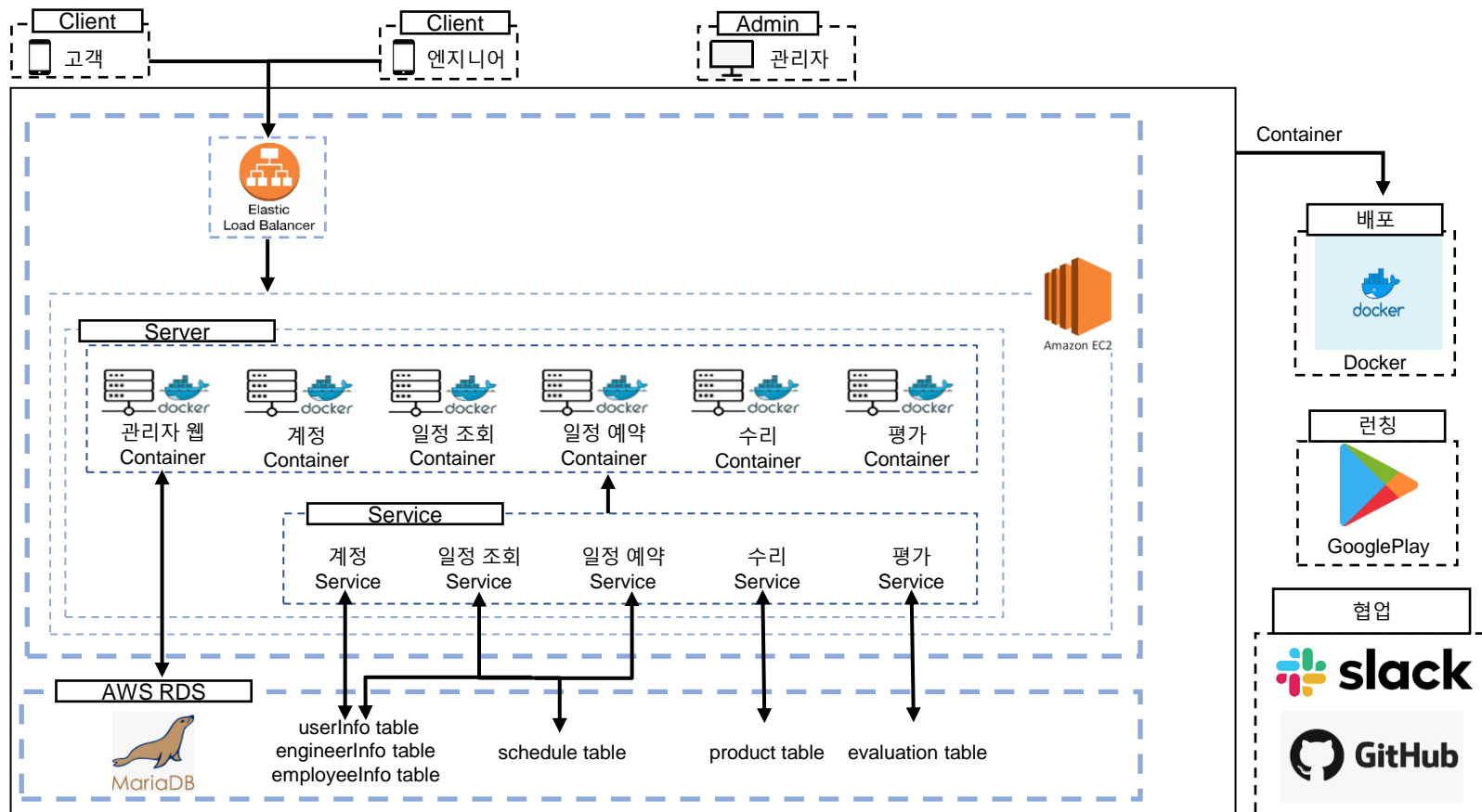
각각 개별의 서비스 개발을 빠르게 하며 유지보수가 쉬움

서비스별로 독립적 배포가 가능하며 모놀로식에 비해 지속적 배포에 유리

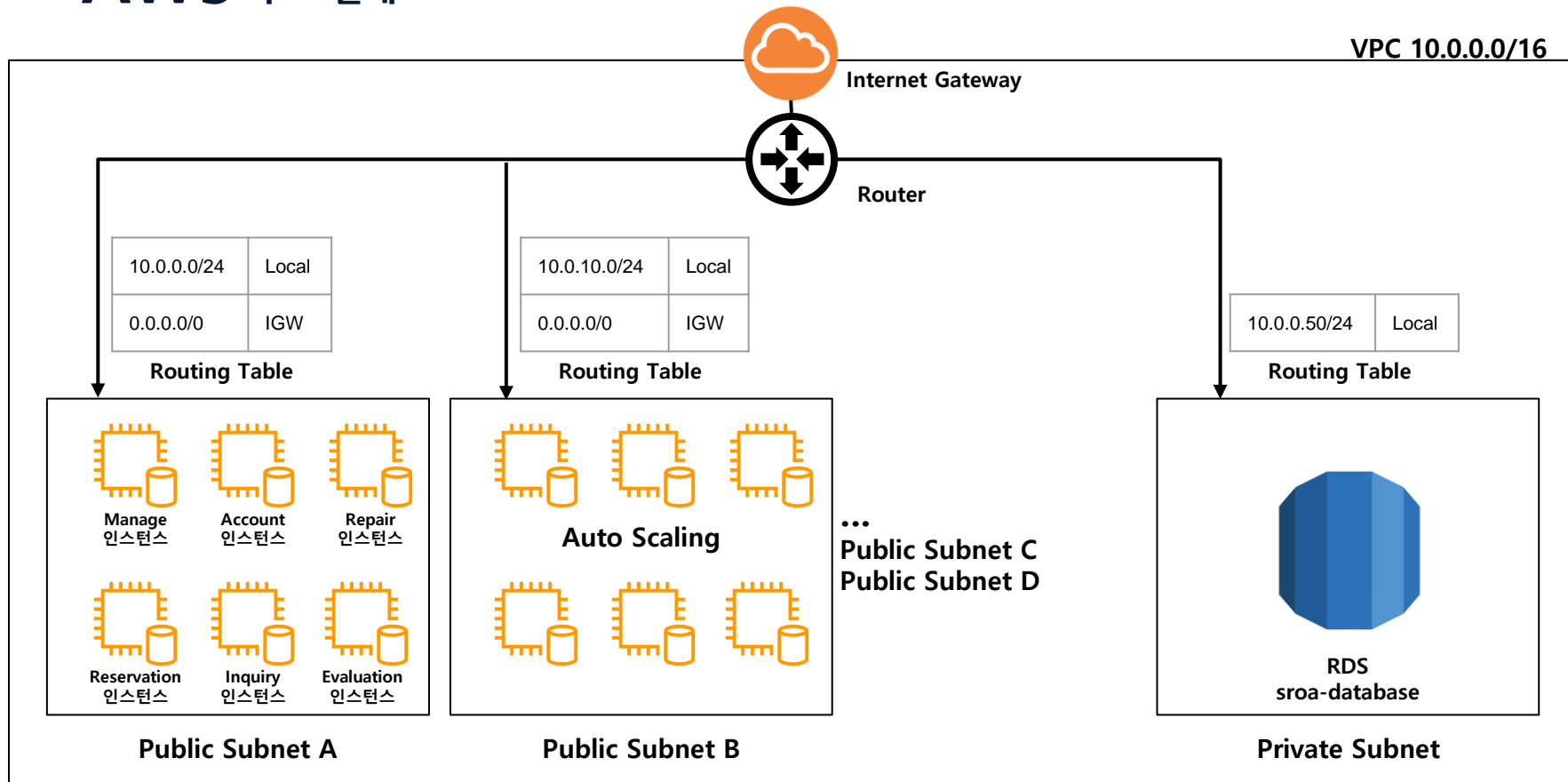
모놀로식에 비해 상대적으로 많이 복잡함

관리자 웹, 계정, 조회, 예약, 수리, 평가 6가지 서비스를 **독립적으로 기능**할 수 있도록 구성

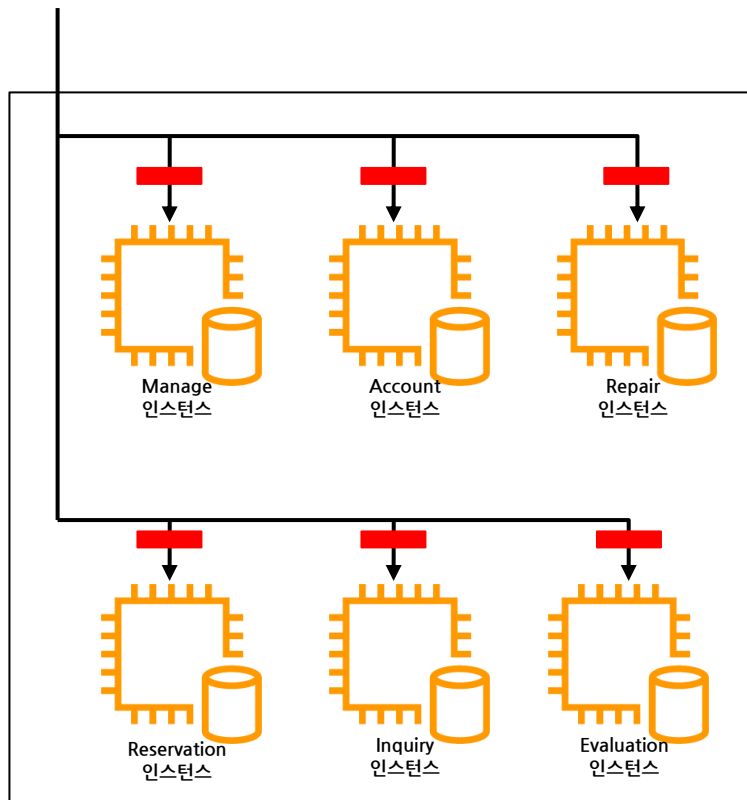
# 아키텍처 설계



# AWS 구조 설계



# AWS 구조 설계

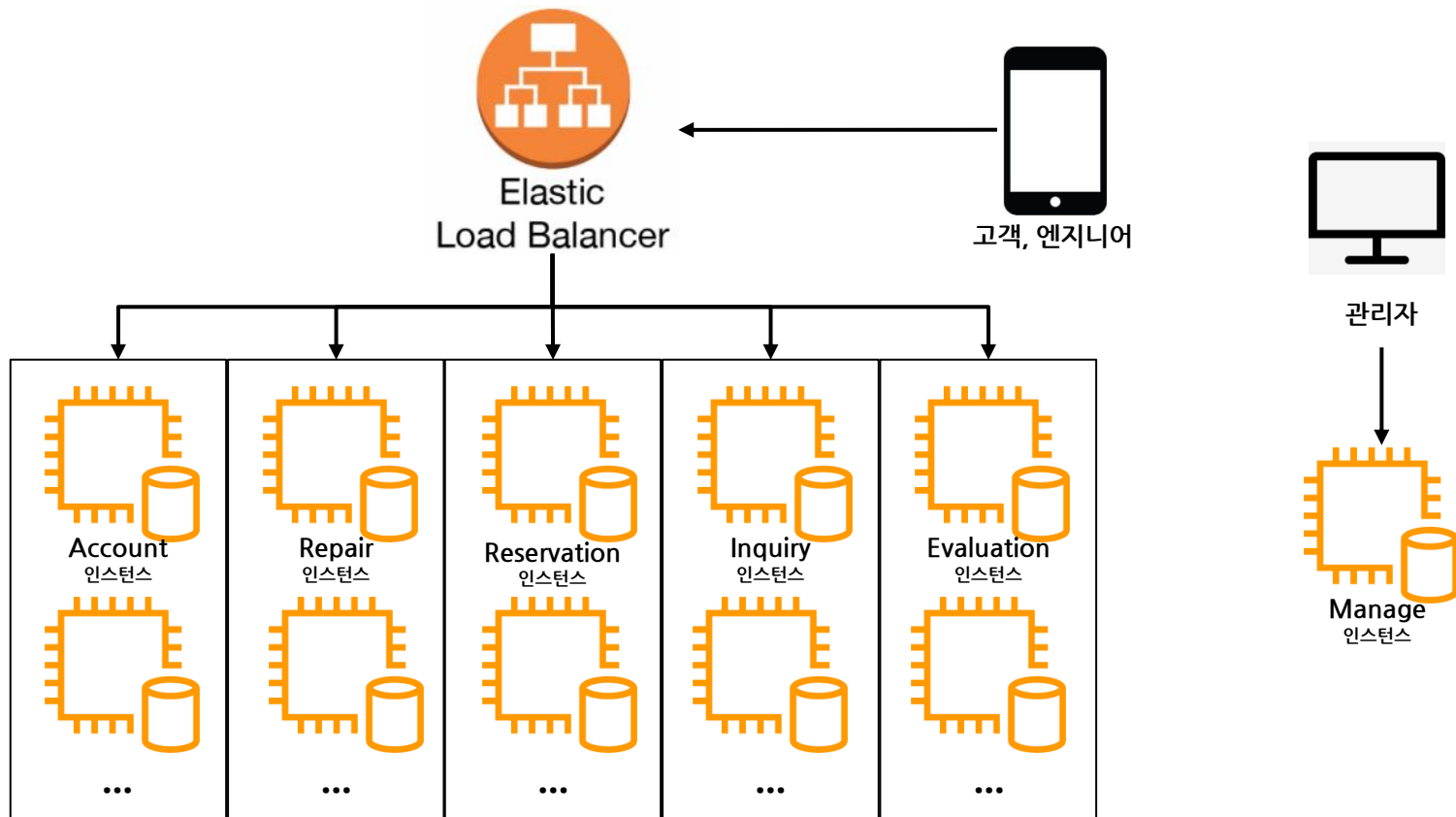


: 보안 그룹

각각의 인스턴스에 해당되는  
인바운드,아웃바운드 규칙



# AWS 구조 설계



# AWS

VPC, 서브넷, 보안그룹 생성

AMI, 시작 템플릿 생성

Auto Scaling 그룹 생성

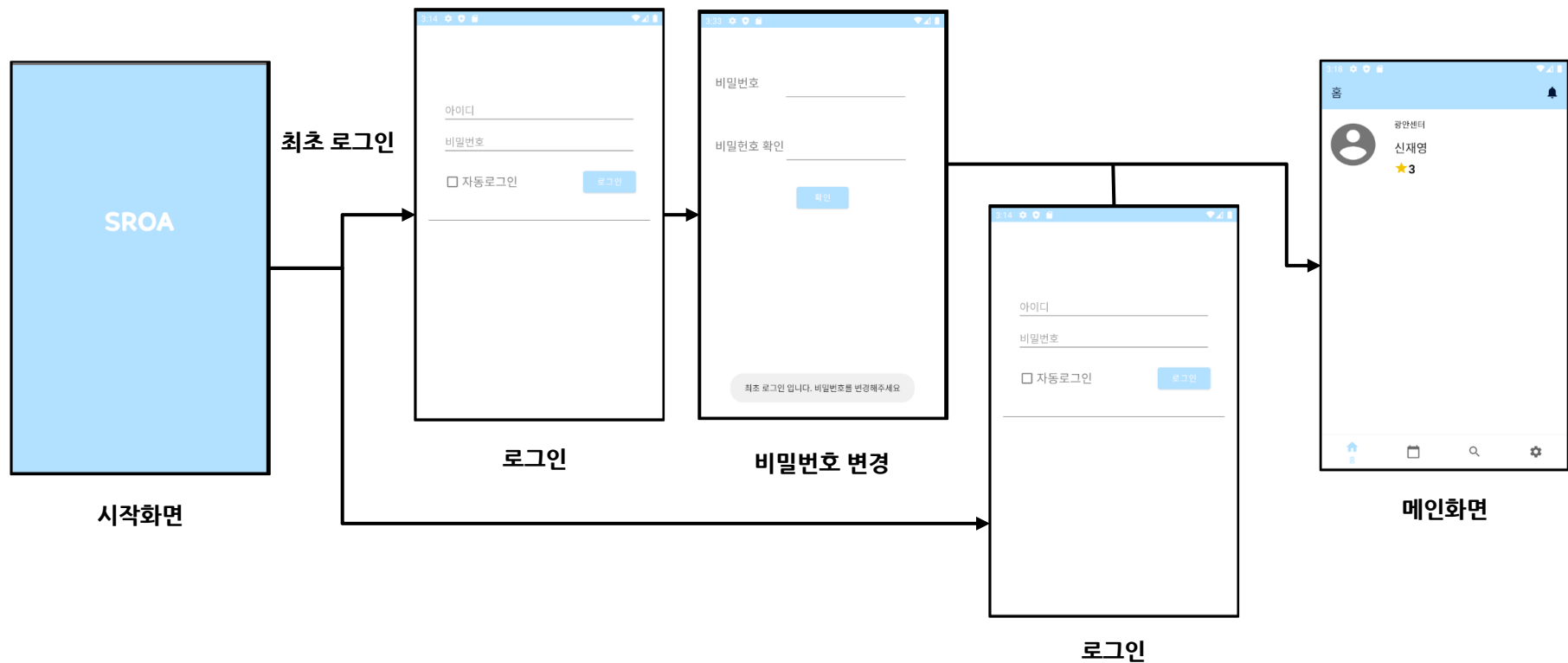
RDS, 서비스 별 인스턴스 생성

대상 그룹, 로드밸런서 생성

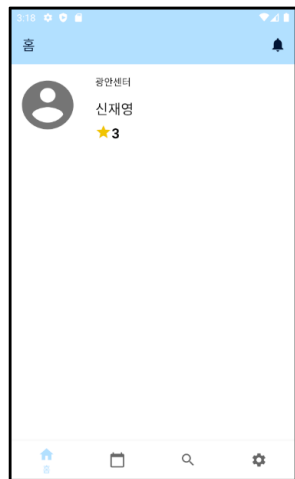
최종테스트



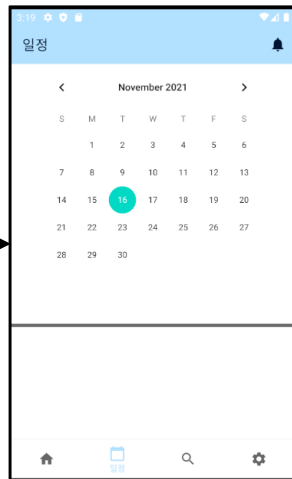
# 어플리케이션 엔지니어 - 계정



# 어플리케이션 엔지니어 - 조회



엔지니어 정보  
당일 일정

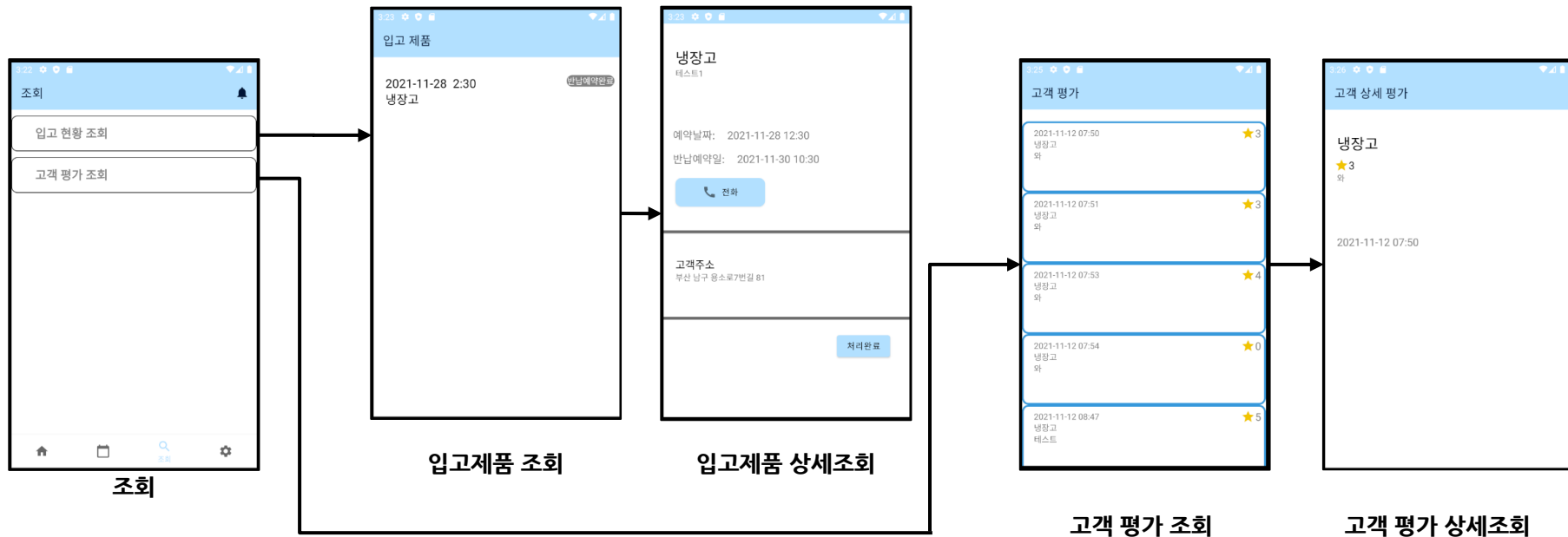


월간 일정



상세일정

# 어플리케이션 엔지니어 - 조회, 수리



# 어플리케이션 고객 - 계정



# 어플리케이션 고객 - 예약

예약

신철민

부산 남구 용소로7번길 81

010-5467-2798

냉장고

스타일러에서 심한 소리가 납

다음

예약정보입력

← 예약

스타일러  
010-5467-2798  
부산 남구 용소로7번길 81

< November 2021 >

S	M	T	W	T	F	S
	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				

09:00 10:30 12:30  
14:00 15:30 17:00

예약시간선택

← 예약

예약확인

부산 남구 용소로7번길 81

전화번호  
010-5467-2798

스타일러

스타일러에서 심한 소리가 납

2021-11-17 12:30

예약하기

예약

← 예약

예약이 완료되었습니다!!

스타일러  
스타일러에서 심한 소리가 납

예약일시  
2021-11-17 12:30

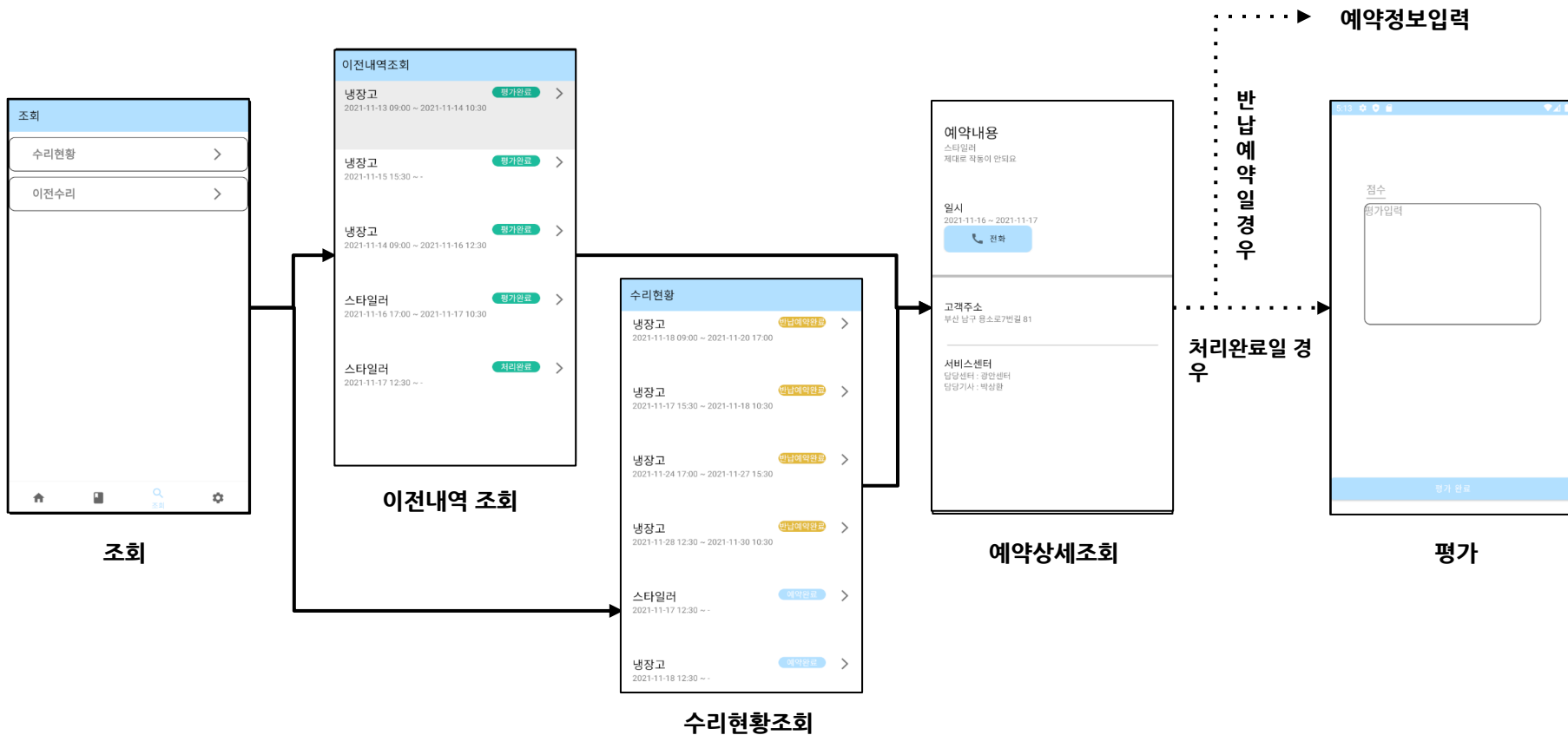
전화

고객주소  
부산 남구 용소로7번길 81

서비스센터  
담당센터: 광안센터  
담당기사: 박성환

예약정보확인

# 어플리케이션 고객 - 조회, 평가





# 관리자 웹 페이지

SROA Home [사원정보](#) [서비스센터별 일정 확인](#) [고객 정보](#) [제품 정보](#)

## 광안센터의 사원정보

사원추가

사원번호	비밀번호	엔지니어 번호	사원명	월간 작업량	평점
20211101	1	6	서상범	1	0
20211102	2	7	신재영	1	0
20211103	3	8	장주연	2	3

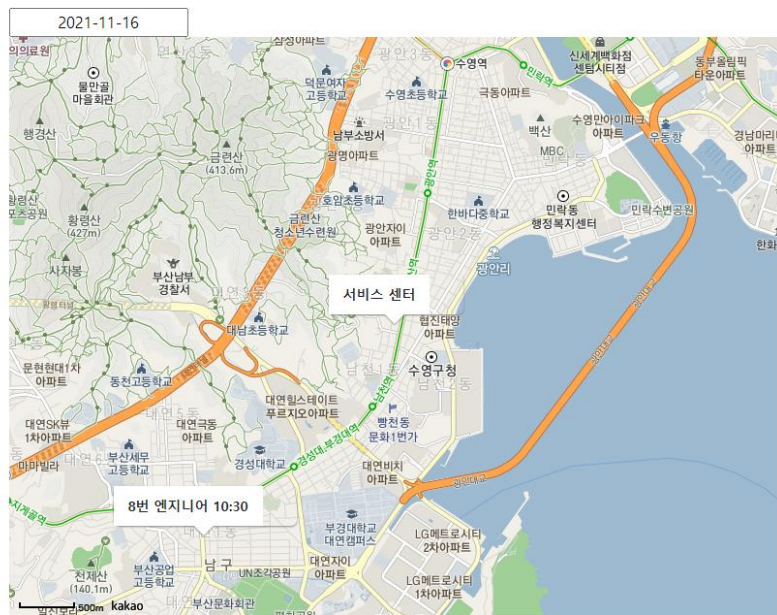
SROA Home [사원정보](#) [서비스센터별 일정 확인](#) [고객 정보](#) [제품 정보](#)

## 고객 정보

고객 번호	이름	ID	PW	주소	연락처
44	박상환	hwan4789	1q2w3e4r	부산광역시 남구 유엔평화로 29번길	부산광역시 남구 유엔평화로 29번길
45	박상환	psh0101	1234	부산 남구 유엔평화로 29번길	부산 남구 유엔평화로 29번길

ome [사원정보](#) [서비스센터별 일정 확인](#) [고객 정보](#) [제품 정보](#)

## 광안센터 일정



# MSA Server API Account Service Server



구분	기능	Method	API
공통	계정 생성시 ID의 중복 여부를 체크	GET	/account/duplicateCheck/{ID}
	어플 로그인		/account/login/{ID}/{PW}
	비밀번호 변경		/account/settingPw/{ID}/{PW}
고객	고객 회원가입	POST	/account/customer/signUp
관리자	엔지니어 계정 생성 관리자가 사원번호를 부여하면서 ID: 사원번호 PW : 00000000로 할당 엔지니어 최초 로그인 시 비밀번호 변경 페이지로 이동하여 변경	POST	/account/createNewEmployee

# MSA Server API InquirySchedule Service Server - 고객



구분	기능	Method	api
고객	메인 페이지 접근/ 메인페이지에서 월 변경 시 해당 달에 예약한 일정 정보를 요청	GET	/schedule/customer/MainPage/{id}/{date}
	메인 페이지에서 날짜 클릭 시 해당 날짜에 대한 일정정보를 요청		/schedule/customer/inquiryDateSchedules/{id}/{date}
	조회 페이지에서 수리현황 클릭 시 서비스 센터에 입고된 물품 정보 요청		/schedule/customer/inquiryCurrentRepair/{id}
	조회 페이지에서 지난 수리 현황 클릭 시 처리 완료된 일정 요청		/schedule/customer/inquiryLastRepair/{id}
	일정 리스트 중 한 개를 클릭하면 해당 일정에 대한 자세한 정보 요청		/schedule/customer/inquiryDetailSchedule/{scheduleNum}
	고객이 예약페이지에 접근하면 저장된 고객의 이름, 주소 등의 정보가 기본적으로 작성		/schedule/customer/inquiryReservationPage/{id}

# MSA Server API InquirySchedule Service Server - 엔지니어

계정 — 조회 — 예약 — 수리 — 평가

구분	기능	Method	api
엔지니어	메인 페이지 접근 시, 현재 일 기준 일정 리스트 요청	GET	/schedule/Engineer/MainPage/{id}
	달력에서 현재 월의 날짜마다 할당된 일정 수를 요청		/schedule/Engineer/inquiryWorkOfMonth/{id}/{month}
	달력에서 날짜 선택 시 해당 날짜에 할당된 일정 정보들을 요청		/schedule/Engineer/inquiryWorkOfDate/{id}/{date}
	현재 자신에게 배정된 일정 중 서비스 센터에 입고된 물품 정보를 요청		/schedule/Engineer/inquiryWarehousingProduct/{id}
	자신의 일정 리스트 중 하나의 일정에 대해 상세 정보 요청		/schedule/Engineer/inquiryDetailSchedule/{scheduleNum}

# MSA Server API

Reservation Service Server



구분	기능	Method	API
고객	수리 예약 신청, 날짜 선택 시 이용 가능한 시간 조회	GET	/schedule/findAvailableTime/{date}/{address}
	예약 신청, {예약 시간, 위치}를 이용하여 최적의 엔지니어에게 일정을 할당		/schedule/findAvailableTimeForReturn/{scheduleNum}/{date}
	입고된 물품 반납 신청, 날짜 선택 시 이용 가능한 시간 조회	POST	/schedule/allocateEngineer
	입고된 물품 반납 신청, 입고 처리를 한 엔지니어에게 할당		/schedule/allocateReturn/{scheduleNum}/{dateTime}

# MSA Server API

Repair Service Server



구분	기능	Method	API
엔지니어	실시간으로 고치지 못하고 서비스센터로 물품을 가지고 가는 입고 처리	GET	/repair/engineer/requestWarehousing/{scheduleNum}
	수리가 완료되었거나 고객에게 수리된 물품을 반납했을 때에 대한 처리완료 처리		/repair/engineer/requestComplete/{scheduleNum}
	입고된 물품에 대해 수리 완료 처리		/repair/engineer/requestRepair/{scheduleNum}

# MSA Server API

Evaluation Service Server



구분	기능	Method	API
엔지니어	자신에게 작성된 평가 정보를 조회	GET	/evaluation/engineer/inquiryEvaluation/{id}
고객	처리 완료된 서비스에 대해 엔지니어 평가	POST	/evaluation/customer/writeEvaluation

# 최적화 알고리즘 일정 배치 시나리오

해당 날짜에 처음으로 할당되는 경우

Service Center



13:00  
예약 요청



일정 사이에 할당되는 경우

09:00



Service Center



11:00  
예약 요청



13:00



해당 날짜의 마지막 일정으로 할당되는 경우

Service Center



11:00



13:00  
예약 요청



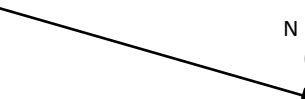


# 최적화 알고리즘 일정 간 거리 계산

N-1 번째 일정  
( $\phi_1, \lambda_1$ )



N 번째 일정  
( $\phi_2, \lambda_2$ )



## HarverSine Fomula

$$\text{hav}(\Theta) = \text{hav}(\varphi_2 - \varphi_1) + \cos(\varphi_1) \cos(\varphi_2) \text{hav}(\lambda_2 - \lambda_1)$$

- $\varphi_1, \varphi_2$ : 1지점과 2지점의 위도 (라디안 단위),
- $\lambda_1, \lambda_2$ : 1지점과 2지점의 경도 (라디안 단위).

$$\text{hav}(\theta) = \sin^2\left(\frac{\theta}{2}\right) = \frac{1 - \cos(\theta)}{2}$$

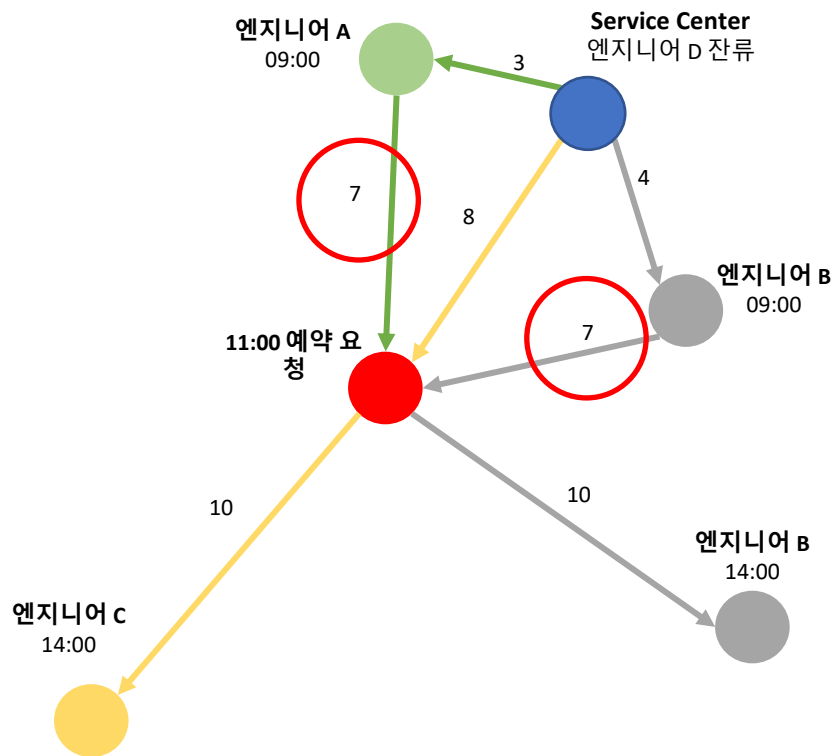
- 거리 계산을 위해 역함수인 arcHarversine을 이용

$$d = r \text{ archav}(h) = 2r \arcsin(\sqrt{h}) \quad * r \text{은 지구 반지름}$$

$$\begin{aligned} d &= 2r \arcsin\left(\sqrt{\text{hav}(\varphi_2 - \varphi_1) + \cos(\varphi_1) \cos(\varphi_2) \text{hav}(\lambda_2 - \lambda_1)}\right) \\ &= 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \end{aligned}$$

# 최적화 알고리즘

## 일정 간 거리를 이용한 판단의 문제점



### 이전 일정과의 거리

#### 엔지니어 A

현재 할당 일정이 당일 마지막 일정, 거리 = 7

#### 엔지니어 B

일정 사이에 할당되는 경우, 거리 = 7

#### 엔지니어 c

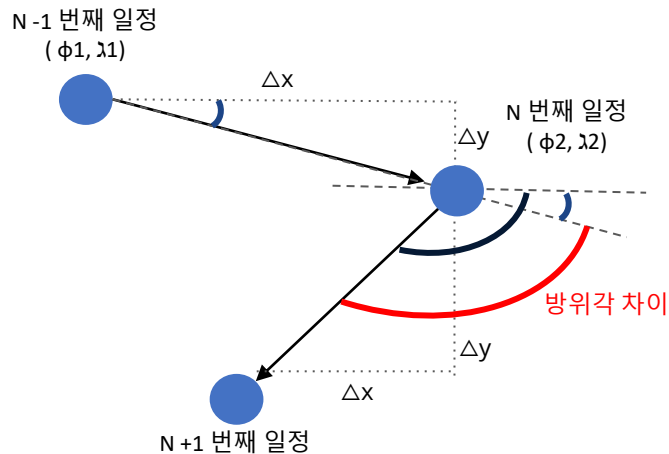
현재 할당 일정이 첫 번째 일정, 거리 = 8

#### 엔지니어 D

해당 날짜의 처음으로 할당되는 일정, 거리 = 8

**방향성 필요 !!**

# 최적화 알고리즘 일정 간 방위각 계산



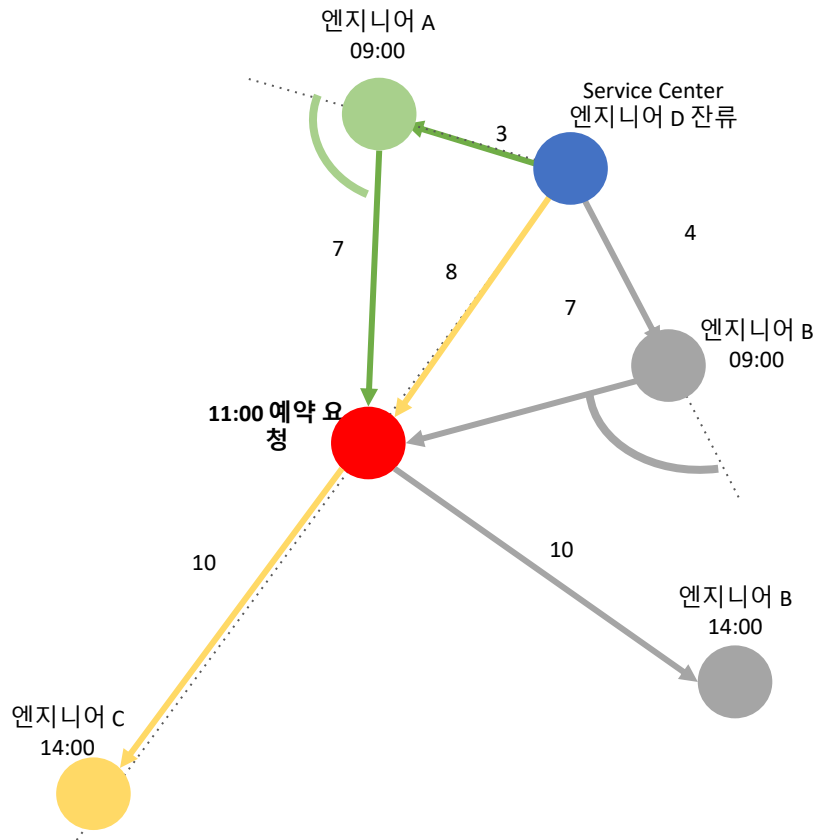
$$\text{방위각} = \text{atan2}(\Delta y, \Delta x)$$

$$\Delta y = \sin(\lambda_2 - \lambda_1) \times \cos(\phi_2)$$

$$\Delta x = \{ \cos(\phi_1) \times \sin(\phi_2) \} - \{ \sin(\phi_1) \times \cos(\phi_2) \times \cos(\lambda_2 - \lambda_1) \}$$

$$\text{atan2}(y, x) = \begin{cases} \arctan\left(\frac{y}{x}\right) & \text{if } x > 0 \\ \arctan\left(\frac{y}{x}\right) + \pi & \text{if } x < 0 \text{ and } y \geq 0 \\ \arctan\left(\frac{y}{x}\right) - \pi & \text{if } x < 0 \text{ and } y < 0 \\ +\frac{\pi}{2} & \text{if } x = 0 \text{ and } y > 0 \\ -\frac{\pi}{2} & \text{if } x = 0 \text{ and } y < 0 \\ \text{undefined} & \text{if } x = 0 \text{ and } y = 0 \end{cases}$$

# 최적화 알고리즘 방향성을 적용한 비교



## 전후 일정과의 방향성 차이

엔지니어 A

거리 = 7

방위각 차이 = 약 120°

엔지니어 B

거리 = 7

방위각 차이 = 약 100°

엔지니어 C

거리 = 8

방위각 차이 = 약 0°

엔지니어 D

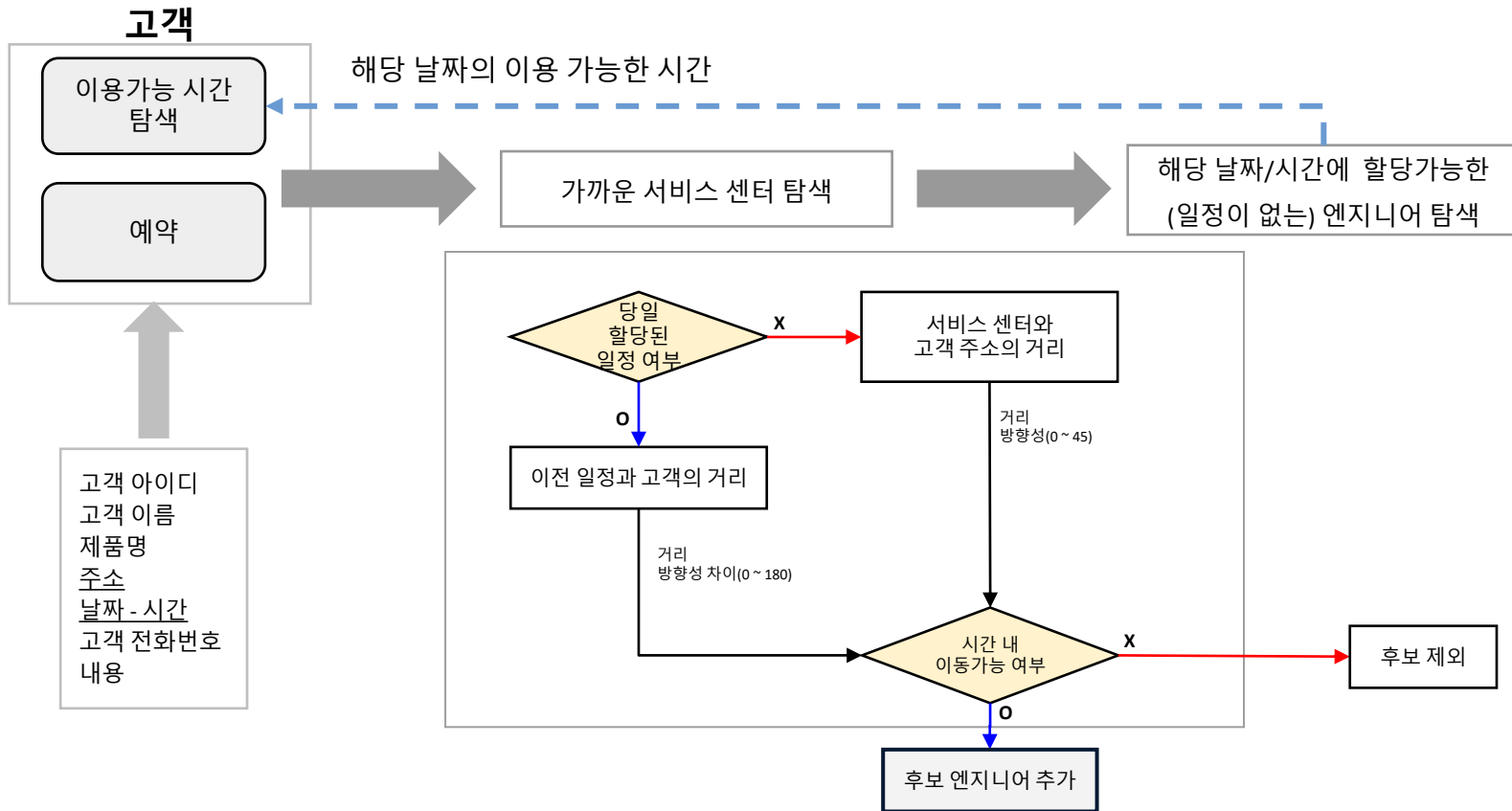
거리 = 8

방위각 = 약 45°

\* 해당 날짜의 일정이 없기 때문에 방위각으로 계산 판단

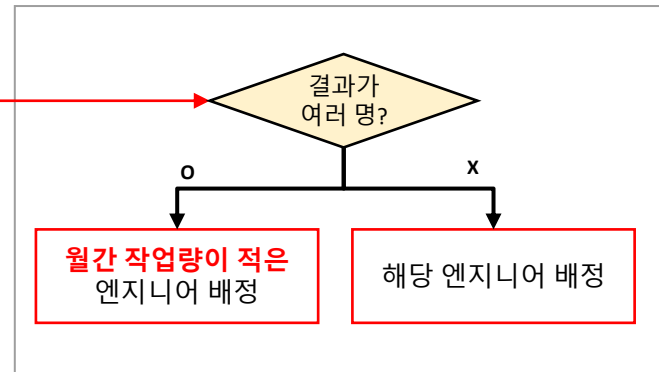
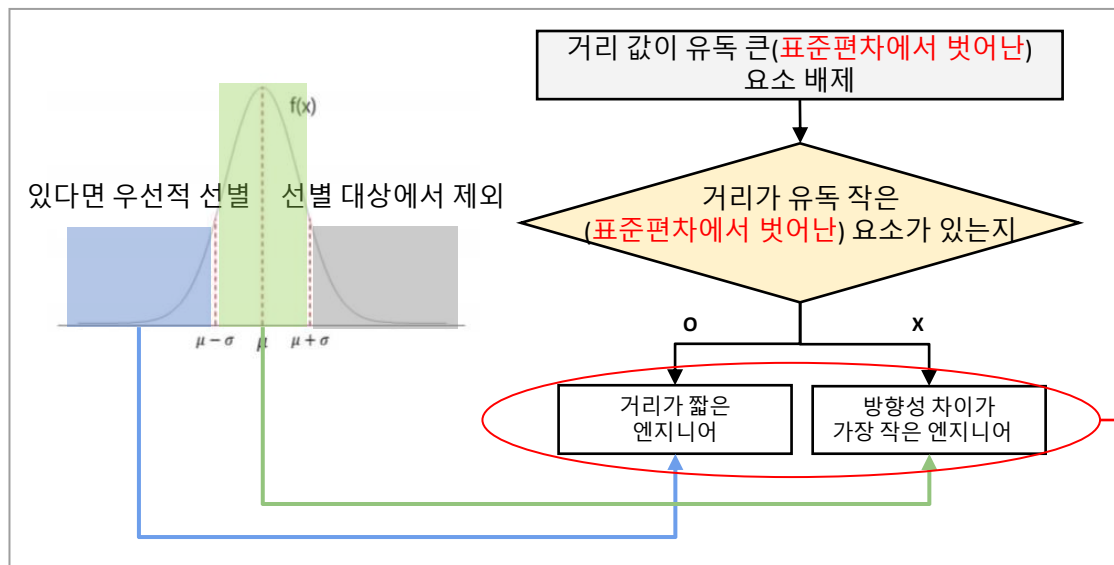
거리가 유사하다면 방위각 차이를 이용해 판단

# 최적화 알고리즘 거리비교



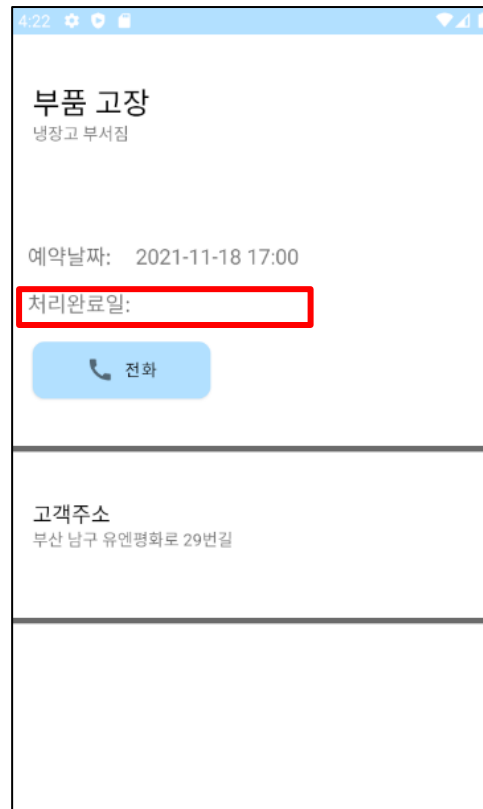
# 최적화 알고리즘

최적 엔지니어 선정





# MSA 유지보수/버전 업데이트





# MSA 유지보수/버전 업데이트

## 인스턴스 (3) 정보

Q 인스턴스 필터링

search: Repair X

필터 지우기

<input type="checkbox"/>	Name	인스턴스 ID
<input type="checkbox"/>	Repair	i-0a5ee897b...
<input type="checkbox"/>	Repair v1.0.1	i-09056a458...

템플릿으로 인스턴스 시작

템플릿 수정(새 버전 생성)

템플릿 삭제

템플릿 버전 삭제

기본 버전 설정

태그 관리

<input type="checkbox"/>	Name	AMI 이름	AMI ID
<input type="checkbox"/>	repair-img	sroa-repair-ami	ami-0e706cccb6e20af7d



lt-064f9cb077e386874

sroa-repair-st

## 시작 템플릿 버전 세부 정보

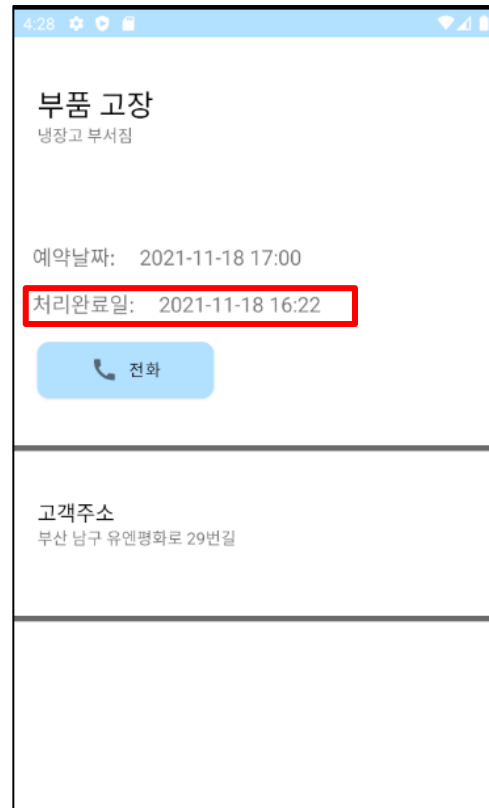
버전

2(기본값)

설명

v1.0.1

# MSA 유지보수/버전 업데이트

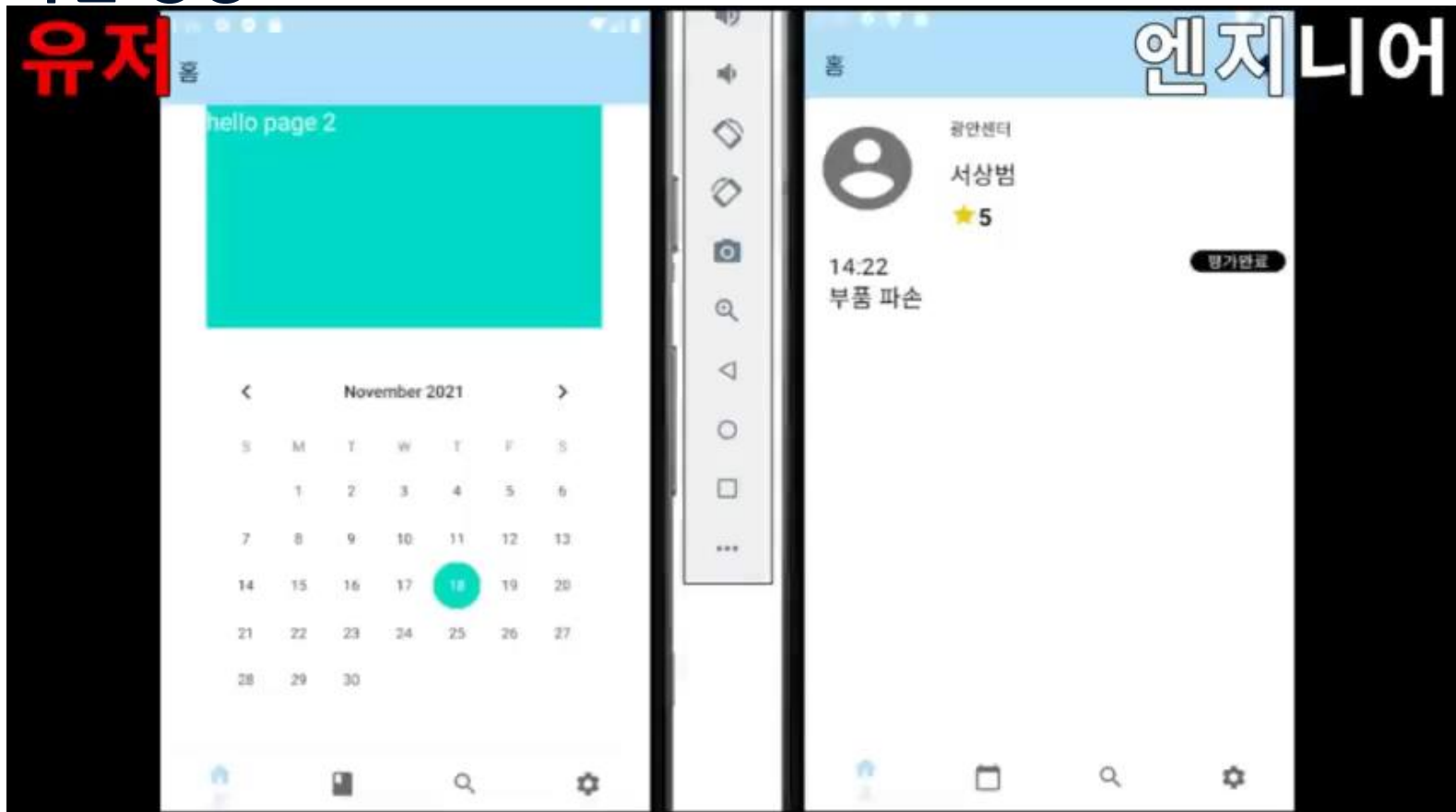




# 04

## 시연 영상

# 시연 영상





# 05

## 후기



# 느낀점

박상환 : 설계부터 구현까지 직접 해보면서 3개월간 클라우드 및 msa에 관해 배운 지식을 활용하는 좋은 기회였다고 생각합니다. 또한 평소 관심을 가지고 있던 android를 통해 작게나마 msa를 체험해볼 수 있어서 좋았습니다. 앞으로 부족했던 부분을 채우고 aws 자격증취득을 위해 노력할 계획입니다.

서상범 : 앞서 배웠던 교육과정에서 부족했던 부분들을 이번 프로젝트를 통해서 채울수 있었고, Kotlin이라는 언어에 대해서 새롭게 배우고 MSA를 경험할수 있었습니다. 4명이 한 팀이 된 첫 프로젝트였지만 팀원들과 함께 잘 마무리할 수 있어서 좋은 시간이었습니다.

신재영 : AWS 설계/구축을 처음으로 직접 해보면서 클라우드의 편리함을 더욱 더 깨달을 수 있었고, 프로젝트를 진행에 설계가 미치는 영향이 매우 크다는 점을 알게 되었습니다. 프리티어만 사용하여 만들려고 하다보니 제한사항이 있었지만 다음번에 AWS 구축을 할 때 이번 기회를 기반으로 더 완성도 높은 프로젝트를 할 수 있을것 같습니다.

장주언: 먼저, FCM을 이용하여 어플리케이션에 알림 기능을 추가 하지 못하여 아쉬웠습니다. MSA 서버별 유지보수, 버전 업데이트를 직접 경험하며 MSA 개발 경험과 Docker 활용 능력을 키울 수 있었습니다.

# 향후 개선 방향

FCM            수신한 엔지니어의 현재 위치 반영  
                 자신의 일정 상태 변화에 따른 알림 기능

Https 통신    웹서버와 암호화 통신으로 고객 및 회사 내부의 정보를 보호

리스케줄링    자정이 될 때 다음 날 일정에 대해 재 배치하여 좀 더 효율적인 동선을 가지게끔 일정 할당

# 06

## QnA





# THANKS!



## 시연 영상 예약 및 평가 처리

