



GoHome

심야 귀가 서비스

컴퓨터과학종합설계 최종발표

Contents

1. 팀 소개
2. 프로젝트 소개
3. 세부 설계 사항
4. 결과 및 평가
5. 향후 과제

1. 팀 소개

팀원 소개



강민성

2015540001

관심 분야
Web(FE/BE), AI

경험
얼마닥, yanawa, RundryRunner

<https://github.com/arkss>



김건호

2015920003

관심 분야
Web(FE/BE), UI/UX

경험
UOSTime, UnivUs

<https://github.com/gunhoflash>



이상엽

2015920029

관심 분야
IoT, Embedded Sys.

경험
추천엔진, 웹 크롤링, 해커톤 수상

<https://github.com/yeoptm>



최연웅

2015540041

관심 분야
ML, PS

경험
ACM-ICPC, 자율주행 대회 참여,
Kaggle 등

<https://github.com/yonsweng>



홍찬표

2015920063

관심 분야
Mobile App, PS

경험
추천엔진, UosMap

<https://github.com/kain6245>

1. 팀 소개

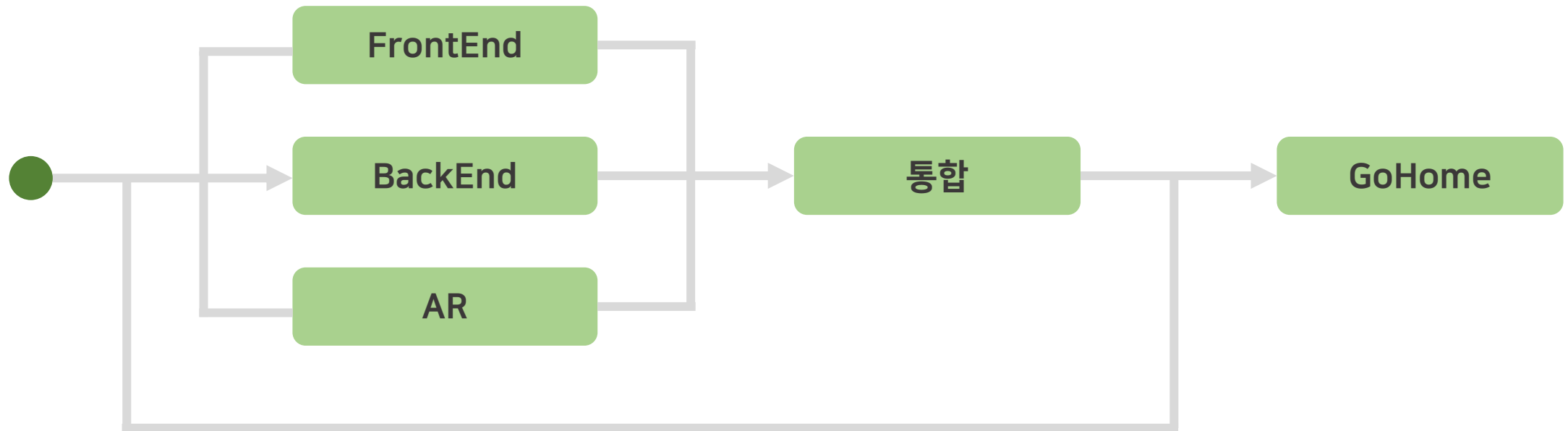
역할 및 추진 체계



이름	책임	세부 내용
김건호	프로젝트 관리자(PM) Logic Server 구현	- 전체 프로젝트 관리/감독 - 최종 검토 및 승인 - Node.js를 통한 로직 서버 구현 - 길 찾기 알고리즘 구현
강민성	User Management Server 구현 API Gateway Server 구현	- Django를 통한 로그인 서버 구현
이상엽	문서 작성 AR 구현	- ARCore 및 Sceneform을 이용한 AR 경로 안내 구현 - 회의록 및 문서 작성
최연웅	Front End 구현	- Android Application 작성 - 클라이언트-서버 간 통신 구현
홍찬표	Front End 구현	- Android Application 작성 - 클라이언트-서버 간 통신 구현

1. 팀 소개

역할 및 추진 체계

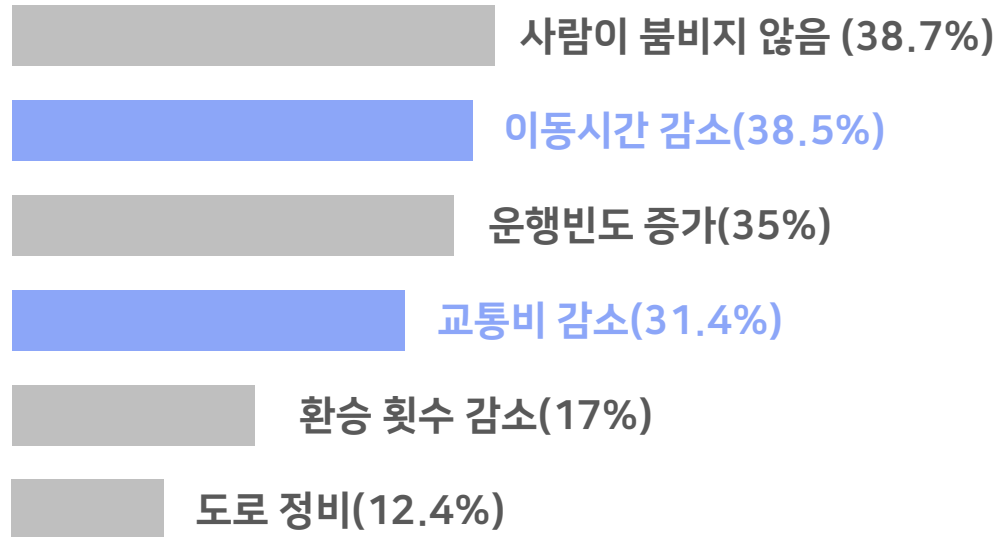


PM 및 조원의 피드백을 통한 각 분야 수정

2. 프로젝트 개요

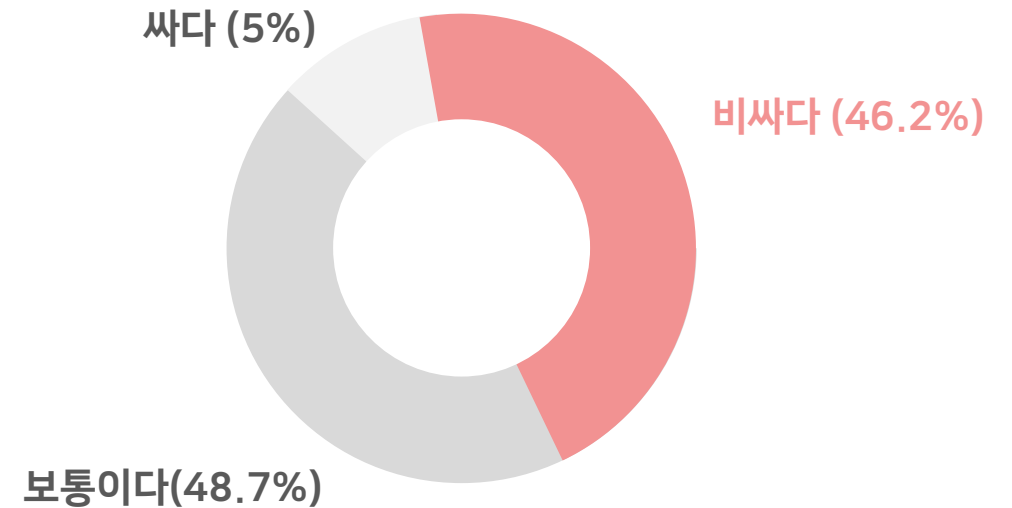
프로젝트 배경

교통수단 개선 필요성(복수응답)



출처 : <https://doooit.co.kr/>

서울시 택시 요금, 가격 어떠한가?

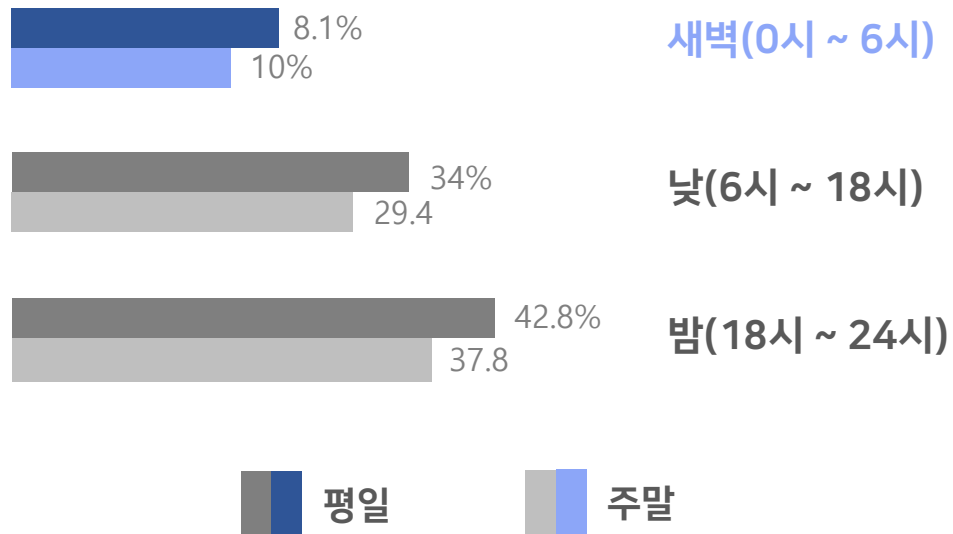


출처 : <https://opengov.seoul.go.kr/council/13969230>

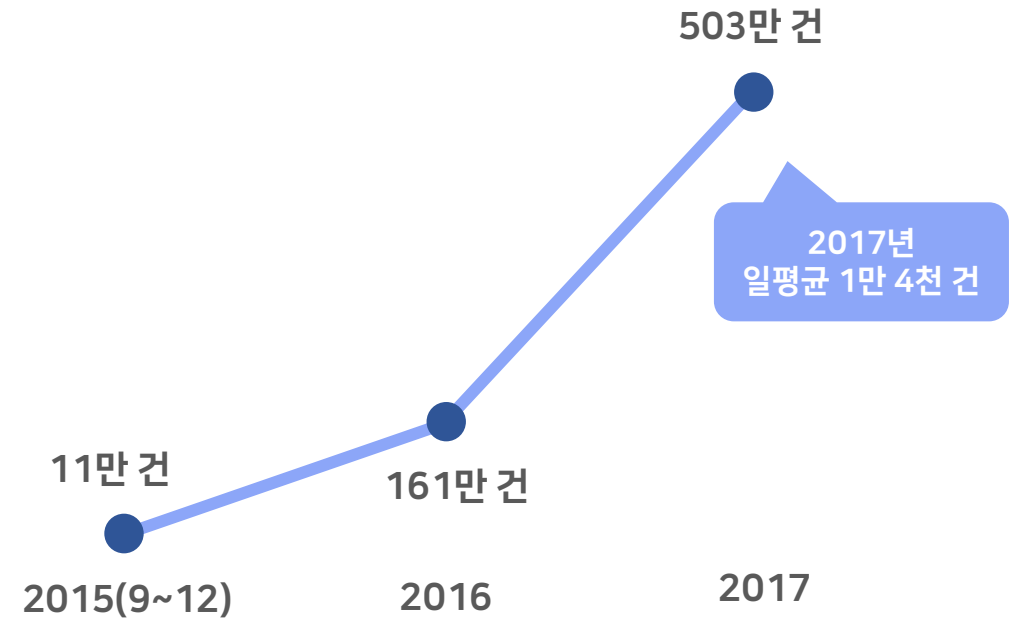
2. 프로젝트 개요

프로젝트 배경

따릉이 시간별 이용 비율



따릉이 이용 건수



2. 프로젝트 개요

프로젝트 목표



2. 프로젝트 개요

SWOT 분석

S

따릉이, 대중교통 경로 안내
AR을 통한 시각적 경로 안내

W

따릉이 이용 시 추가비용 발생
사용 지역이 서울로 국한됨

O

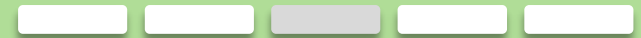
따릉이 연계 길 안내 애플리케이션 부재
N버스의 긴 배차/정류장 간격
새벽 택시 사용자가 많음

T

지도 애플리케이션 시장의 포화
초반 신규 사용자 유치의 어려움

3. 세부 설계 사항

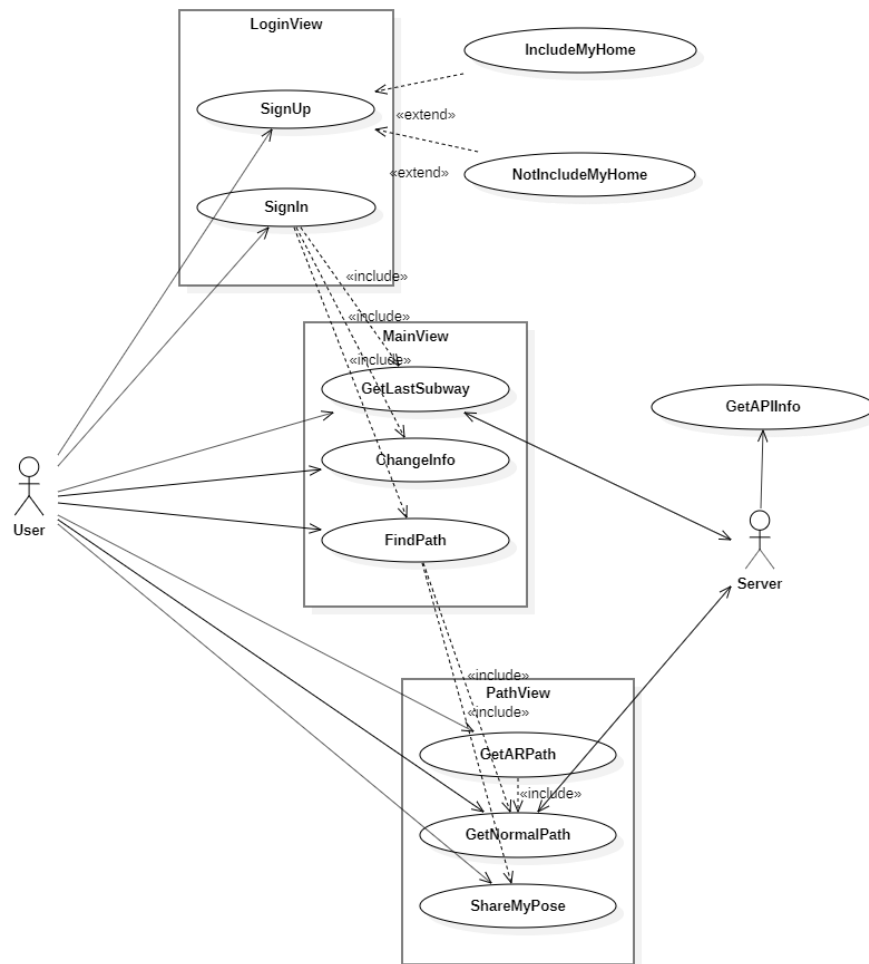
사용자 요구사항



번호	요 구 사 항	D or W
1	현 위치에서 목적지까지 최적 경로를 얻는다.	D
2	AR(Augmented Reality)을 통한 경로 안내를 사용한다.	D
3	사용자 정보를 변경한다.	D
4	지인과 위치공유 기능을 사용한다.	D
5	버스, 지하철 등 대중교통 막차를 안내한다.	W
6	현재 위치를 지도에 표시한다.	W
7	주변 숙박업소를 검색한다.	W
8	맞춤형 안내 서비스를 사용한다.	W
9	Client Application의 로딩시간은 2초를 넘기지 않는다.	D
10	Client Application의 UI/UX가 통일감을 준다.	W
11	Client Application의 색상이 통일감을 준다.	W

3. 세부 설계 사항

Use Case Diagram



Actor 목록

User

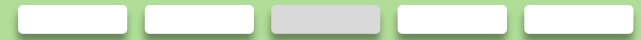
회원가입이 되어 있는 사용자

Server

회원 정보, 경로 정보 등을 관리하는 서버

3. 세부 설계 사항

Use Case Diagram

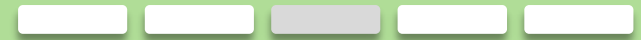


Use Case 목록

SignUp	User가 신규 회원가입을 수행함
SignIn	User가 로그인을 수행함
IncludeMyHome	User가 본인의 집 정보를 포함한 신규 회원가입 수행
NotIncludeMyHome	User가 본인의 집 정보를 포함하지 않은 신규 회원가입 수행
GetLastSubway	User가 현재 대중교통의 막차 시간을 전달받음
ChangeInfo	User가 사용자 정보를 갱신

3. 세부 설계 사항

Use Case Diagram



Use Case 목록

FindPath

User가 경로 탐색을 요청

GetNormalPath

User의 현재 위치에서 도착지까지의 경로를 반환

ShareMyPose

User의 위치를 지인에게 공유

GetARPath

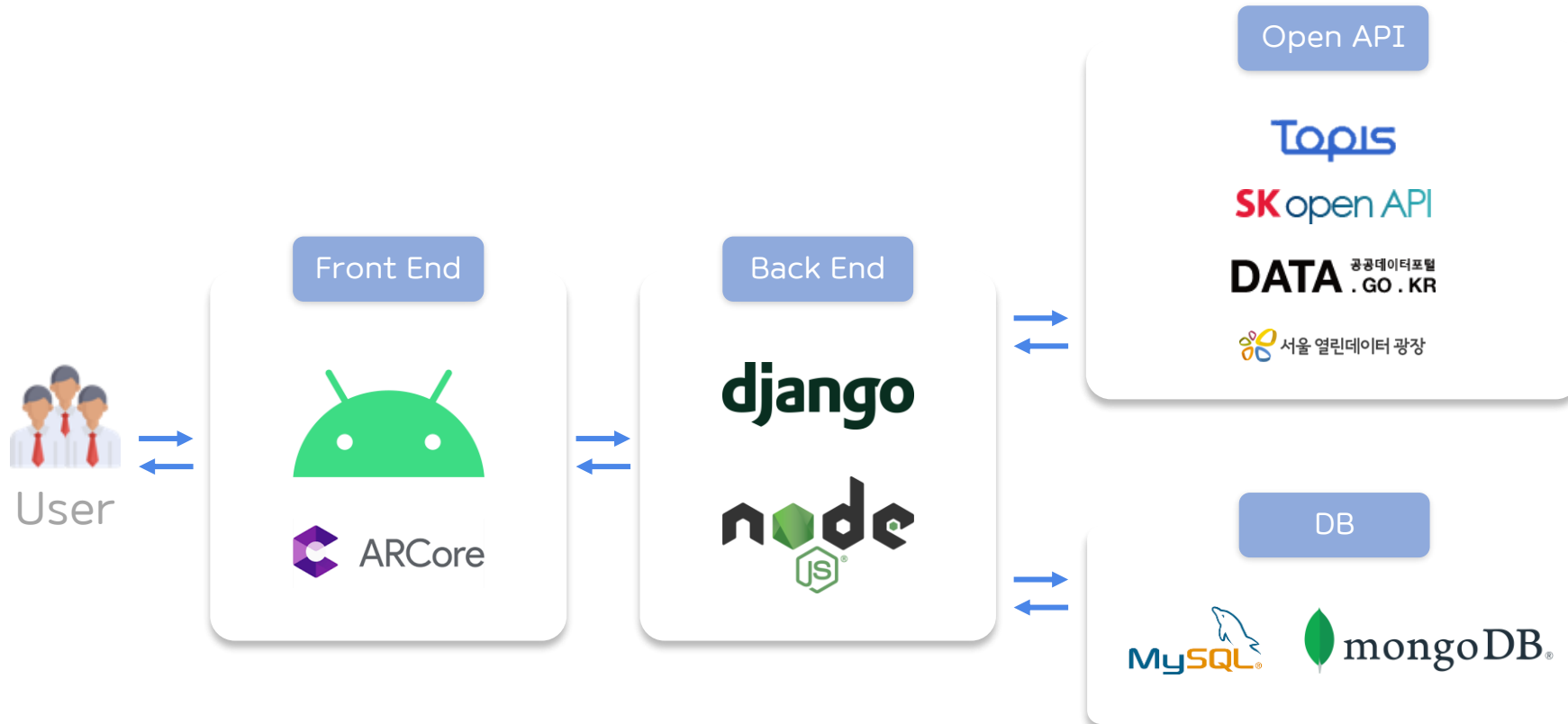
User가 AR 이미지를 통한 경로 탐색을 수행

GetAPIInfo

Server가 버스, 지하철, 따릉이, T map API 정보를 받아온다.

3. 세부 설계 사항

시스템 설계-소프트웨어 아키텍처



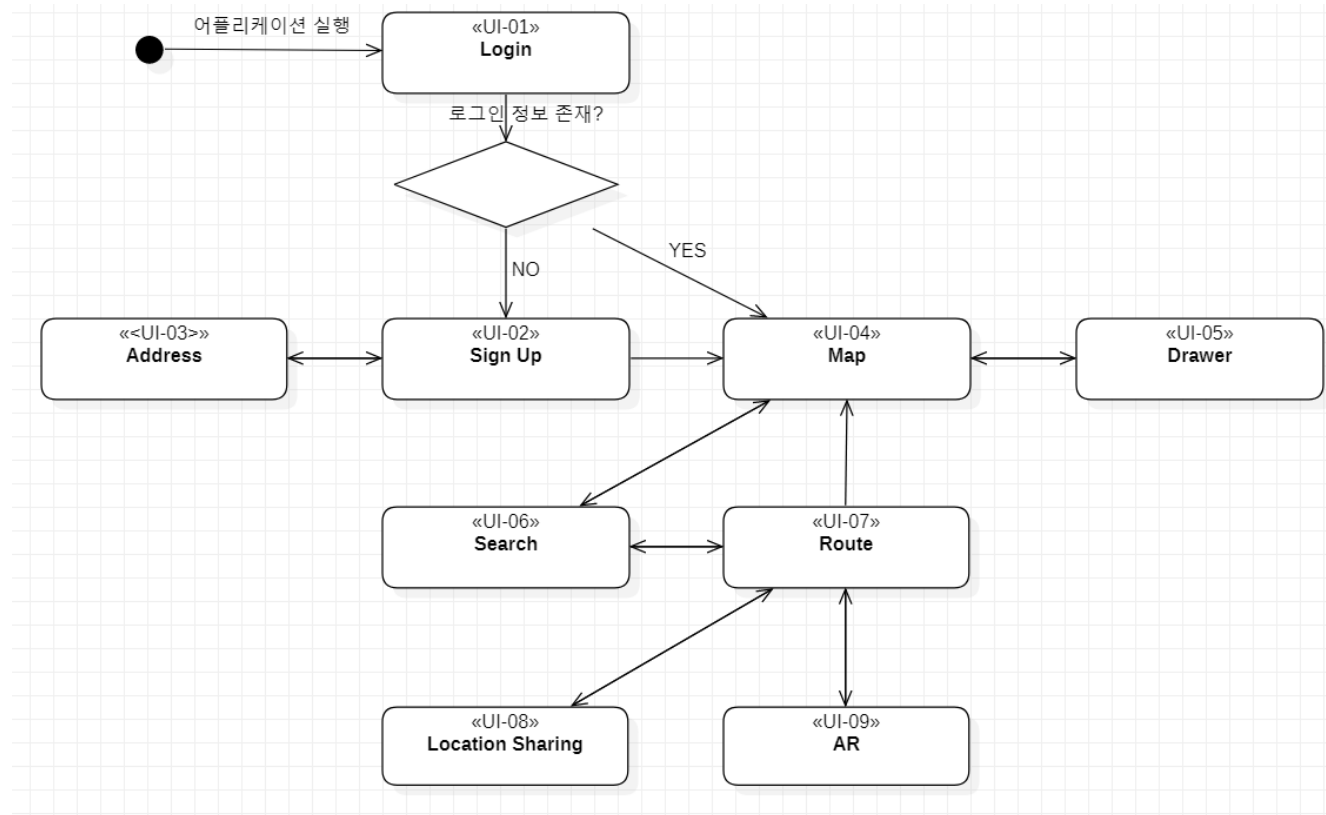
3. 세부 설계 사항

시스템 설계-UI Flow



3. 세부 설계 사항

시스템 설계-UI Flow

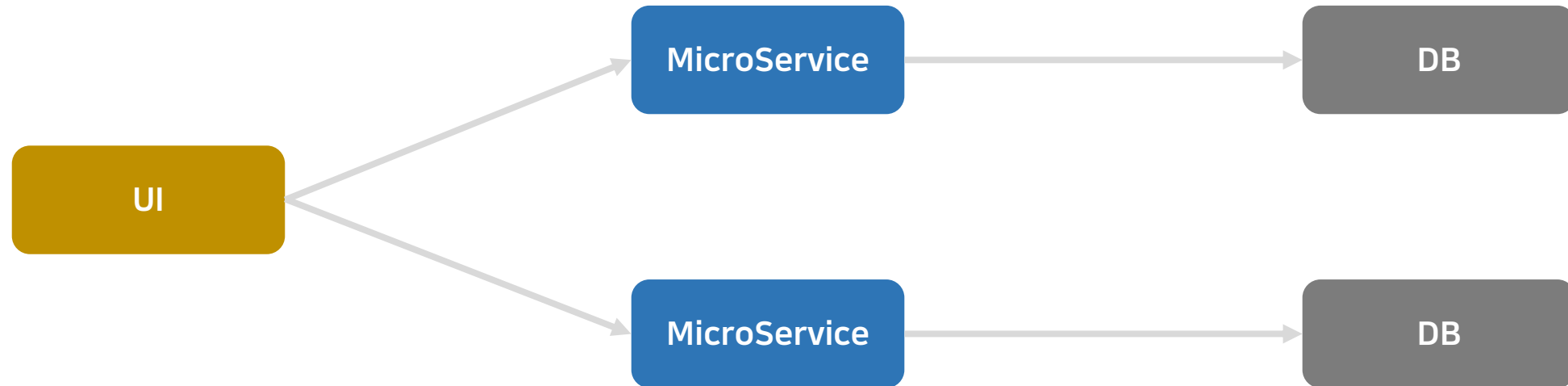


3. 세부 설계 사항

세부 기술 선택 사항 - MSA

MSA(Micro Service Architecture)

- 큰 어플리케이션을 여러 개의 작은 어플리케이션으로 나눠 만드는 아키텍처
- 빌드 및 테스트 시간 단축, 유연성이 높음, 트래픽이 높은 서버에 적합
- 본 프로젝트에서는 Django, Node.js 서버, MongoDB, MySQL을 활용함



3. 세부 설계 사항

세부 기술 선택 사항 - RESTful API

REST(Representational State Transfer) API

- 자원을 **URI로 표시**하고 해당 자원의 상태를 주고 받는 것

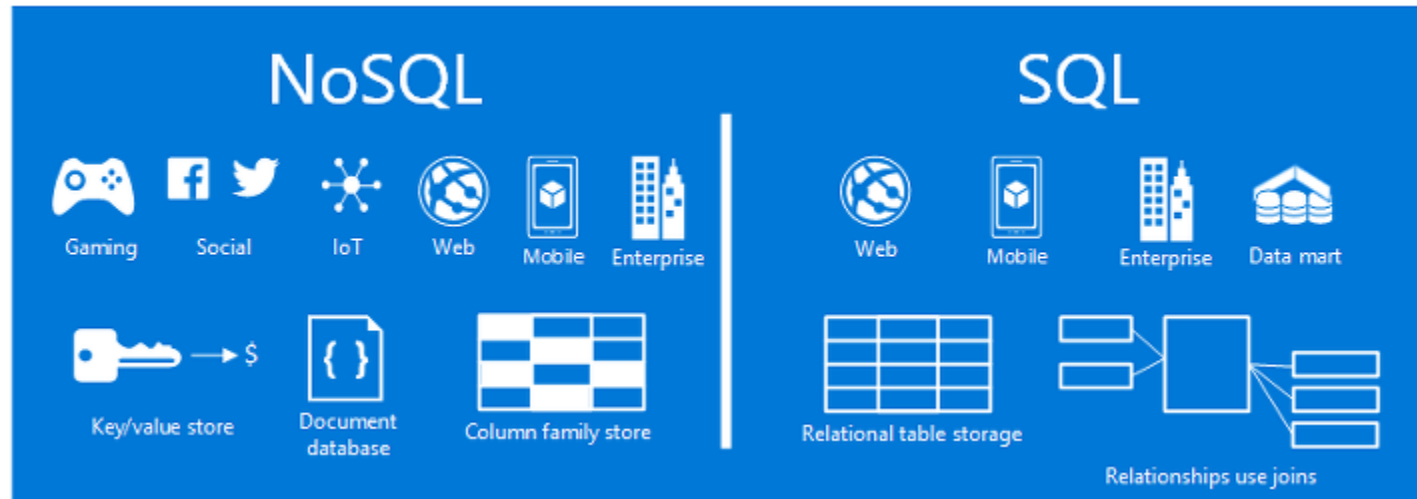


3. 세부 설계 사항

세부 기술 선택 사항 - MongoDB

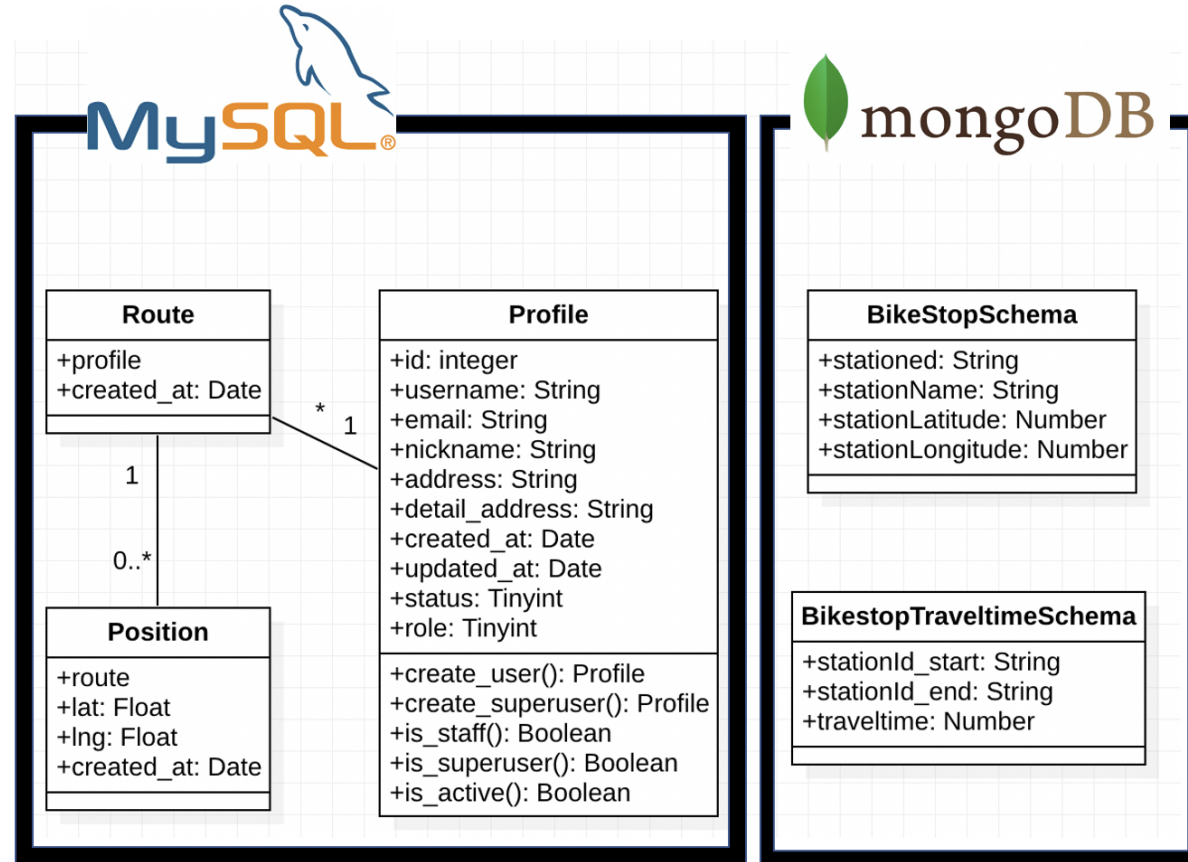
MongoDB

- NoSQL(Not Only SQL)의 일종
- Key-Value 형태의 BSON으로 데이터 저장
- SQL을 사용하지 않아 단순, 대량의 데이터 다루기 용이



3. 세부 설계 사항

시스템 설계-DB Table



3. 세부 설계 사항

세부 기술 선택 사항 - 인증 Token

JWT(Json Web Token)

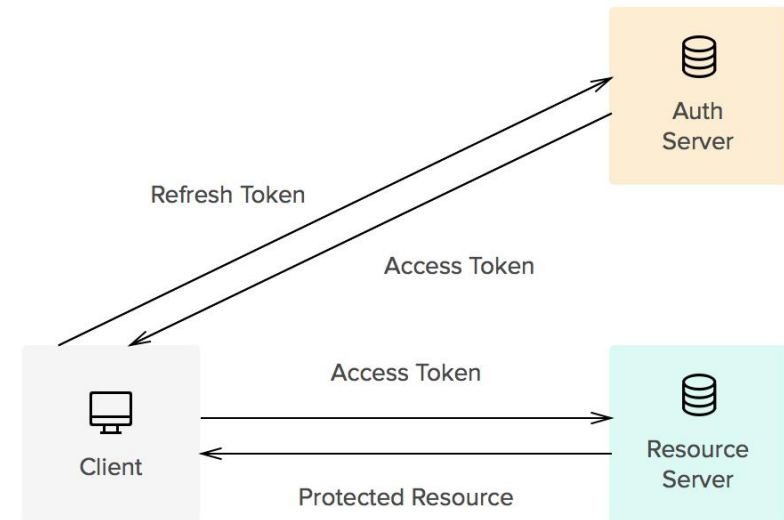
- 클라이언트의 인증을 위한 방식
- Payload에 자주 참조되지만 보안 문제가 없는 데이터 저장(서버, DB 트래픽 낮춤)

Access Token

- JWT의 토큰 탈취 위협 방지

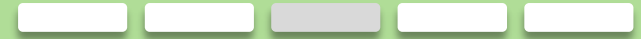
Refresh Token

- JWT의 토큰 탈취 위협 방지



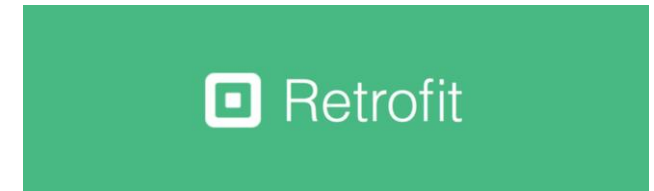
3. 세부 설계 사항

세부 기술 선택 사항 - Retrofit2, Fragment Layout



Retrofit2

- Android 클라이언트에서 서버와 HTTP 통신을 위해 사용하는 라이브러리
- Type-Safe HTTP 클라이언트
- 많은 기능을 빠르고 안전하게 구현할 수 있음



Fragment Layout

- Android Studio에서 Intent를 사용하여 Activity를 전환하면 끊김 현상 발생
- 부드러운 화면 전환을 위하여 Fragment Layout 사용

3. 세부 설계 사항

세부 기술 선택 사항 - AR 관련

ARCore, Sceneform

- ARCore은 증강 현실 어플리케이션 구현을 위해 Google에서 개발한 소프트웨어 키트
- Sceneform은 ARCore 내에서 3D UI 개발을 위해 지원하는 라이브러리
- Android Studio, Unity, Unreal Engine 등에서 지원
- 본 프로젝트에서는 Android Studio를 통해 구현



.obj, .mtl, .sfa 파일

- 3D 모델링 이미지 정보를 담고 있는 파일
- .obj : 3D 모델 이미지 정보 포함
- .mtl : 3D 모델 이미지의 재질 정보 포함
- .sfa : .obj 파일을 Sceneform 에서 활용할 수 있도록 변환



3. 세부 설계 사항

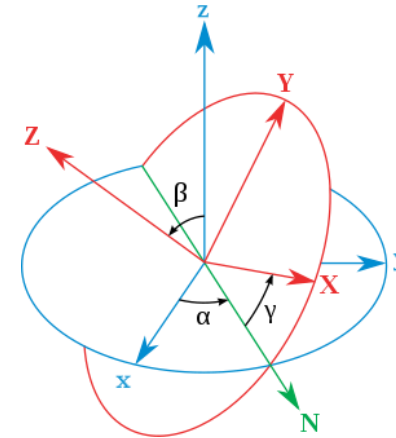
세부 기술 선택 사항 - Euler Angle과 Quaternion

Euler Angles

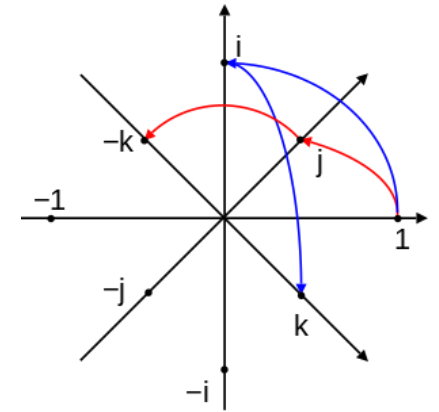
- X, Y, Z의 절대적인 축을 기준으로 회전을 나타내는 개념
- 일반적인 회전 개념을 나타내는 데에 활용

Quaternion

- (x,y,z)의 회전의 기준이 되는 원점 벡터가 주어짐
- (x,y,z)의 회전 원점 벡터에 대한 회전 스칼라 값인 w가 주어짐
- Euler Angles 에서 나타나는 축이 겹치는 짐벌락 현상이 나타나지 않음
- AR 이미지의 회전에는 Quaternion 회전을 사용



Euler Angles

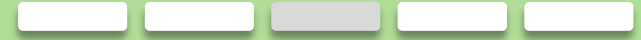


Quaternion

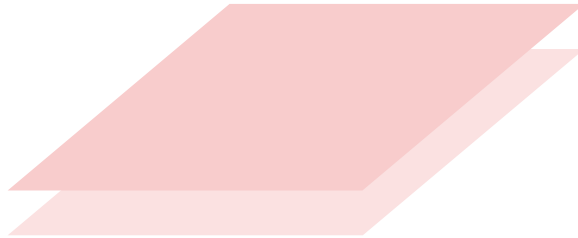
$$\begin{aligned} ij &= k \\ ji &= -k \\ ij &= -ji \end{aligned}$$

3. 세부 설계 사항

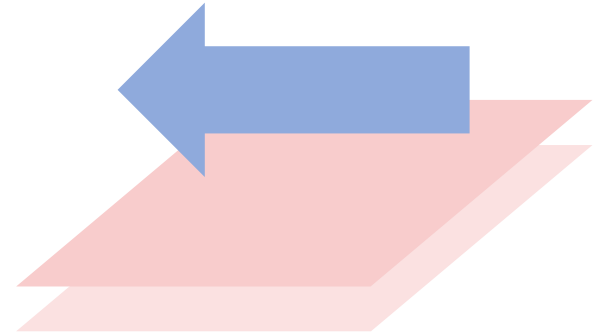
시스템 설계-AR 이미지 생성



평면 인식



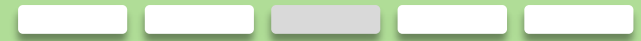
평면 위 노드 생성



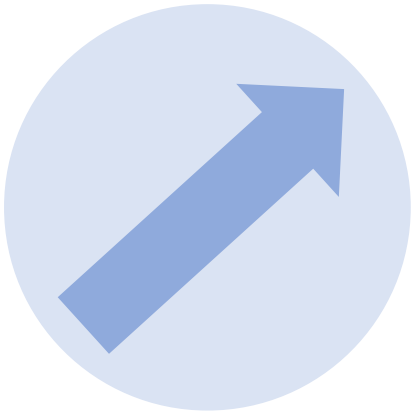
AR 이미지 생성

3. 세부 설계 사항

시스템 설계-AR 이미지 회전

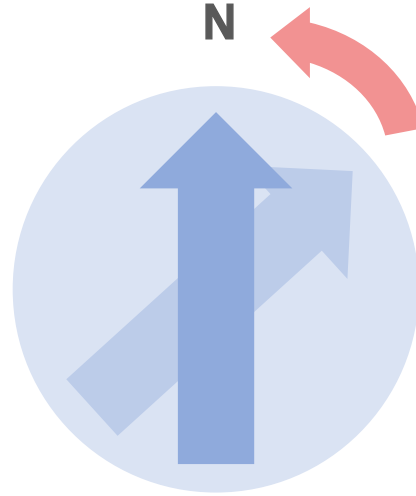


N



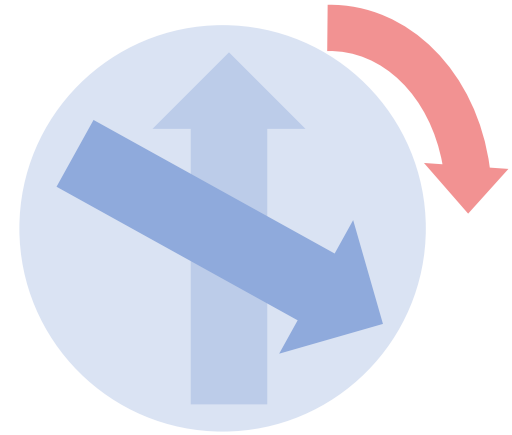
사용자 방향에 생성

N



- 센서 각도만큼 회전
(북쪽을 향하게)

N



- 목표 각도만큼 회전

3. 세부 설계 사항

시스템 설계-길 찾기 알고리즘



3. 세부 설계 사항

시스템 설계-길 찾기 알고리즘



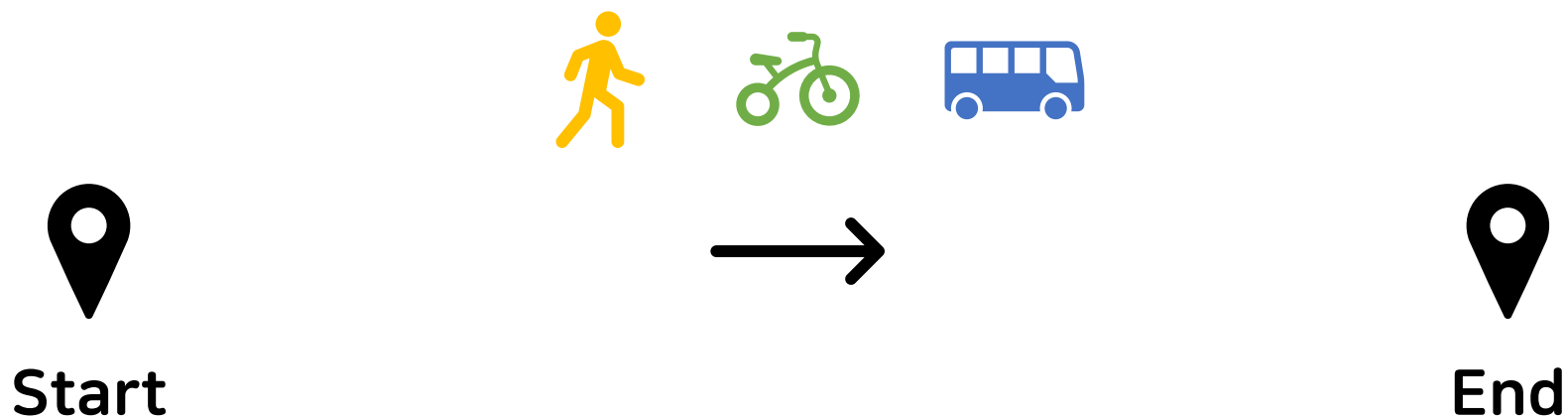
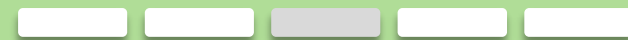
Start



End

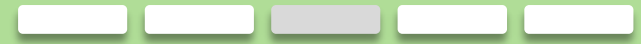
3. 세부 설계 사항

시스템 설계-길 찾기 알고리즘



3. 세부 설계 사항

시스템 설계-길 찾기 알고리즘



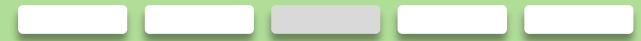
Start



End

3. 세부 설계 사항

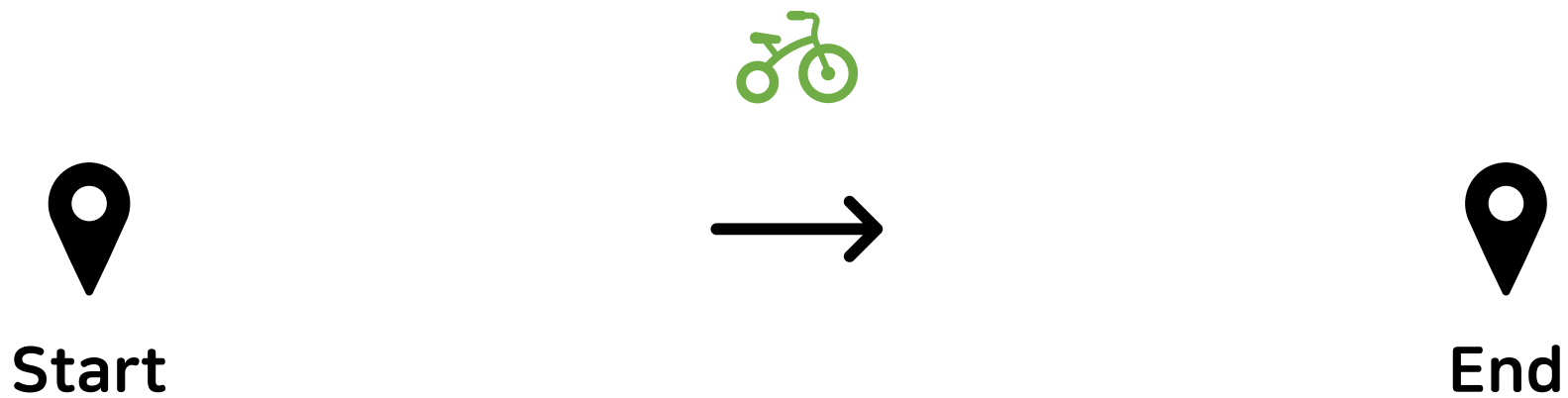
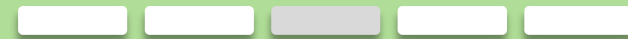
시스템 설계-길 찾기 알고리즘



Use T map API

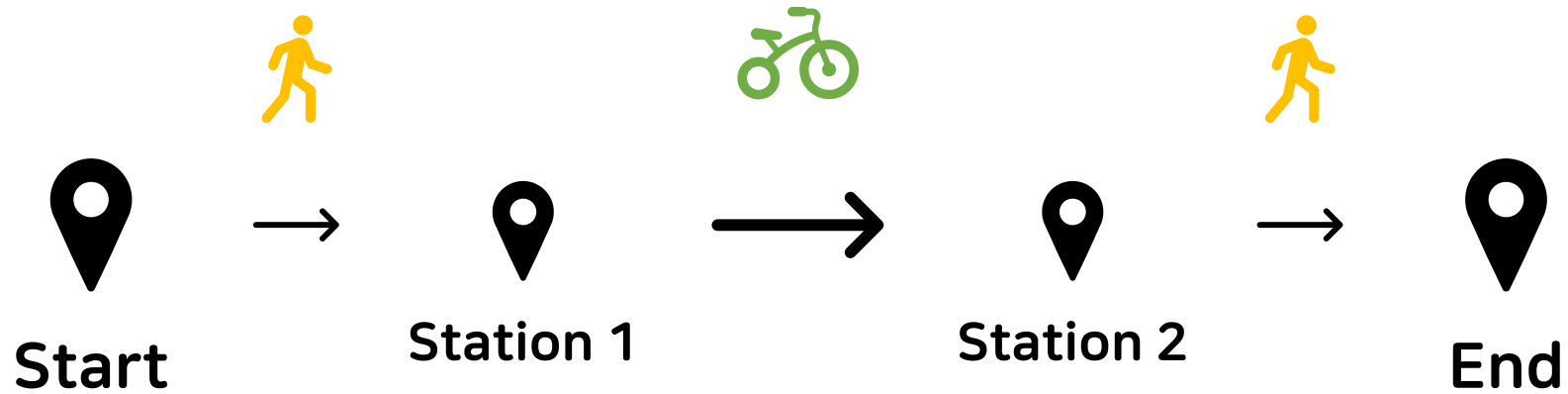
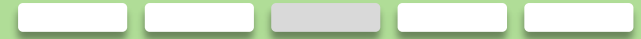
3. 세부 설계 사항

시스템 설계-길 찾기 알고리즘



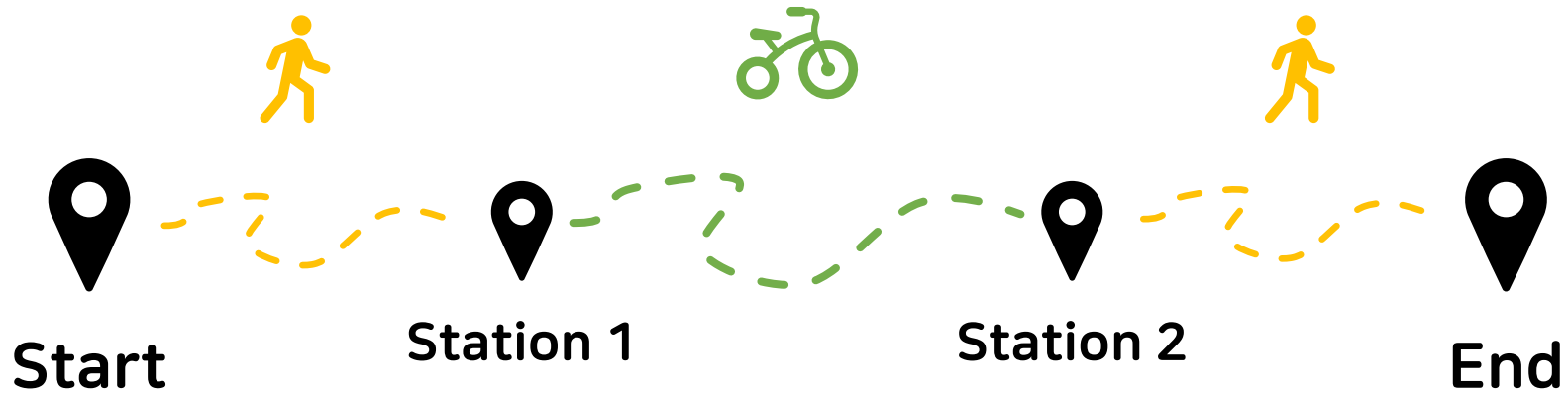
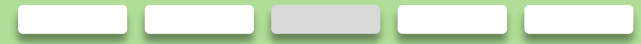
3. 세부 설계 사항

시스템 설계-길 찾기 알고리즘



3. 세부 설계 사항

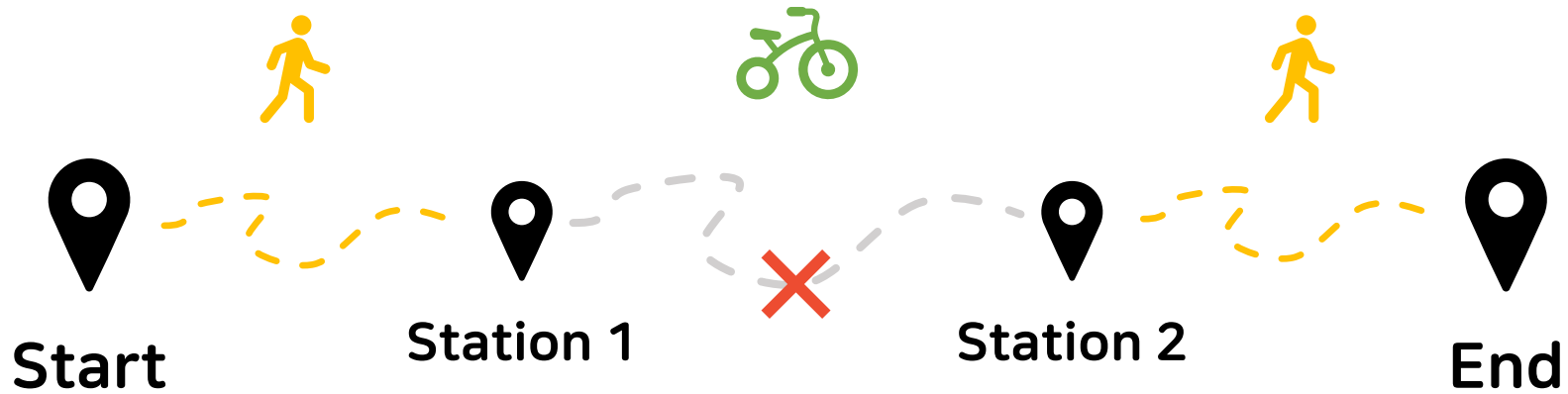
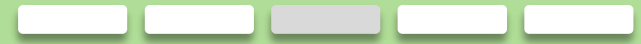
시스템 설계-길 찾기 알고리즘



Use T map API

3. 세부 설계 사항

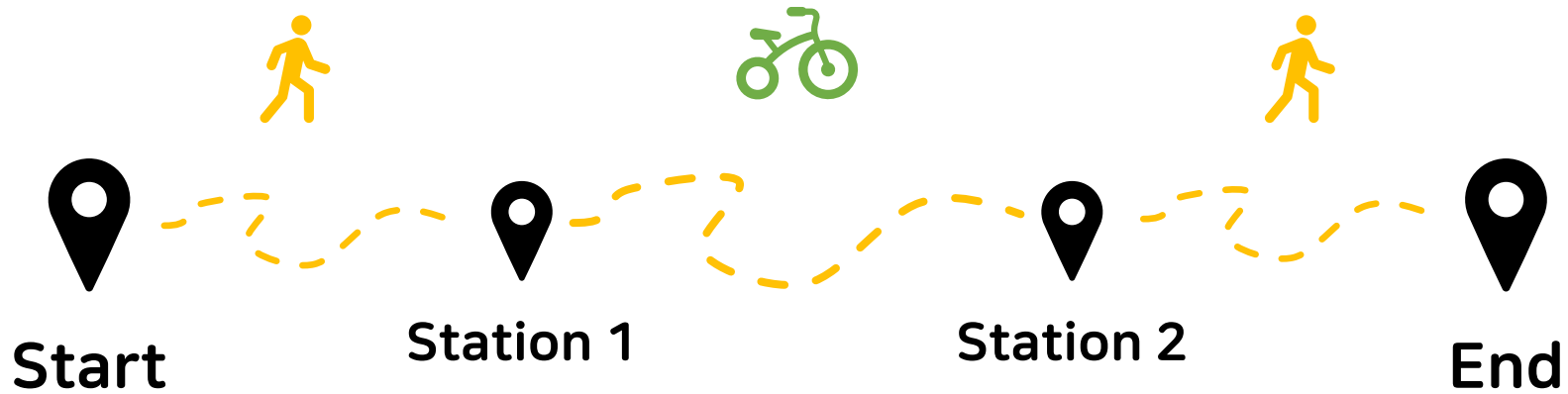
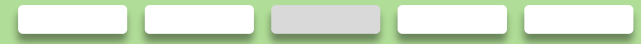
시스템 설계-길 찾기 알고리즘



Use T map API

3. 세부 설계 사항

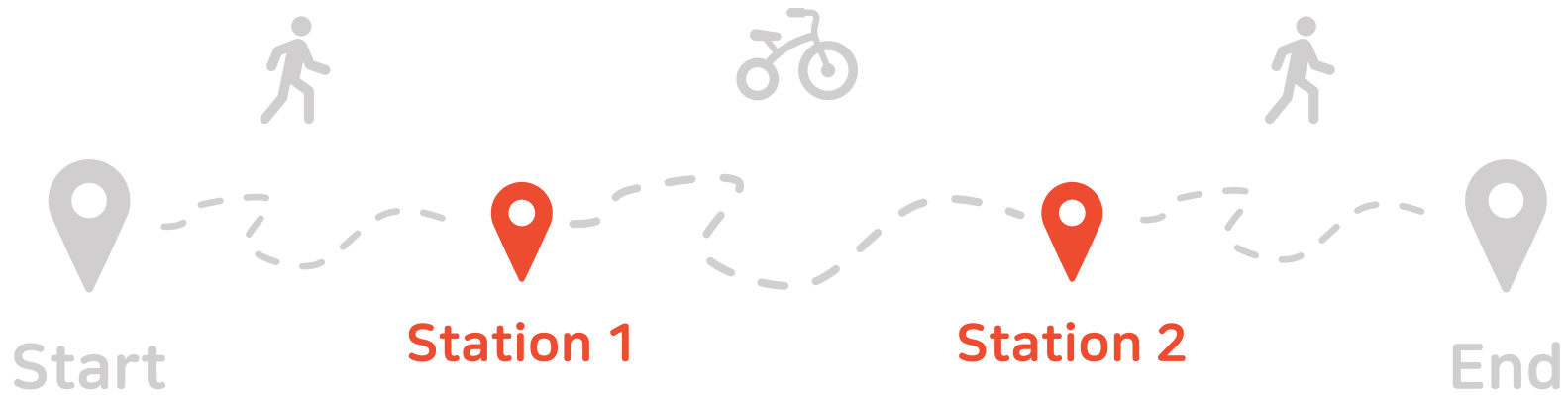
시스템 설계-길 찾기 알고리즘



Use T map API

3. 세부 설계 사항

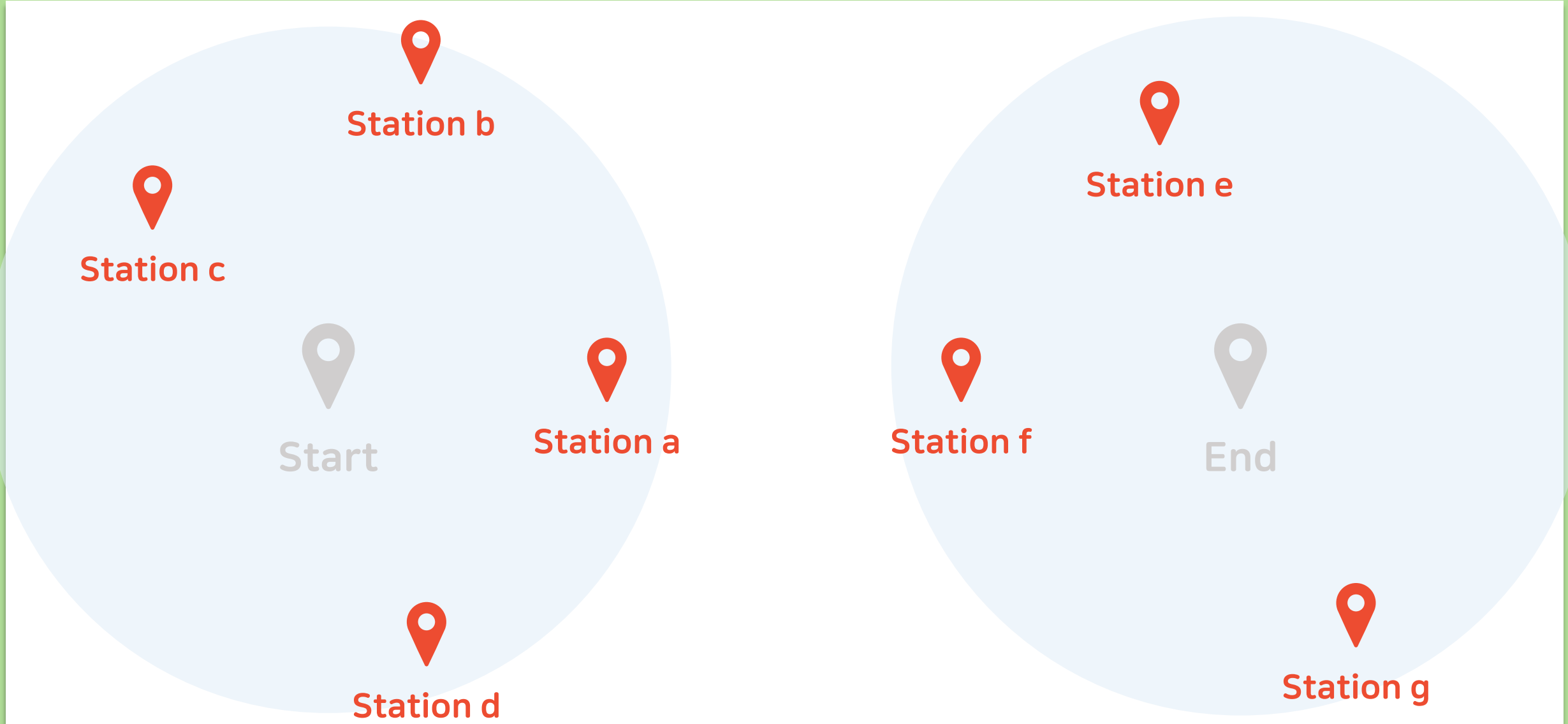
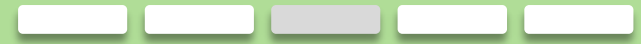
시스템 설계-길 찾기 알고리즘



How to select 2 stations?

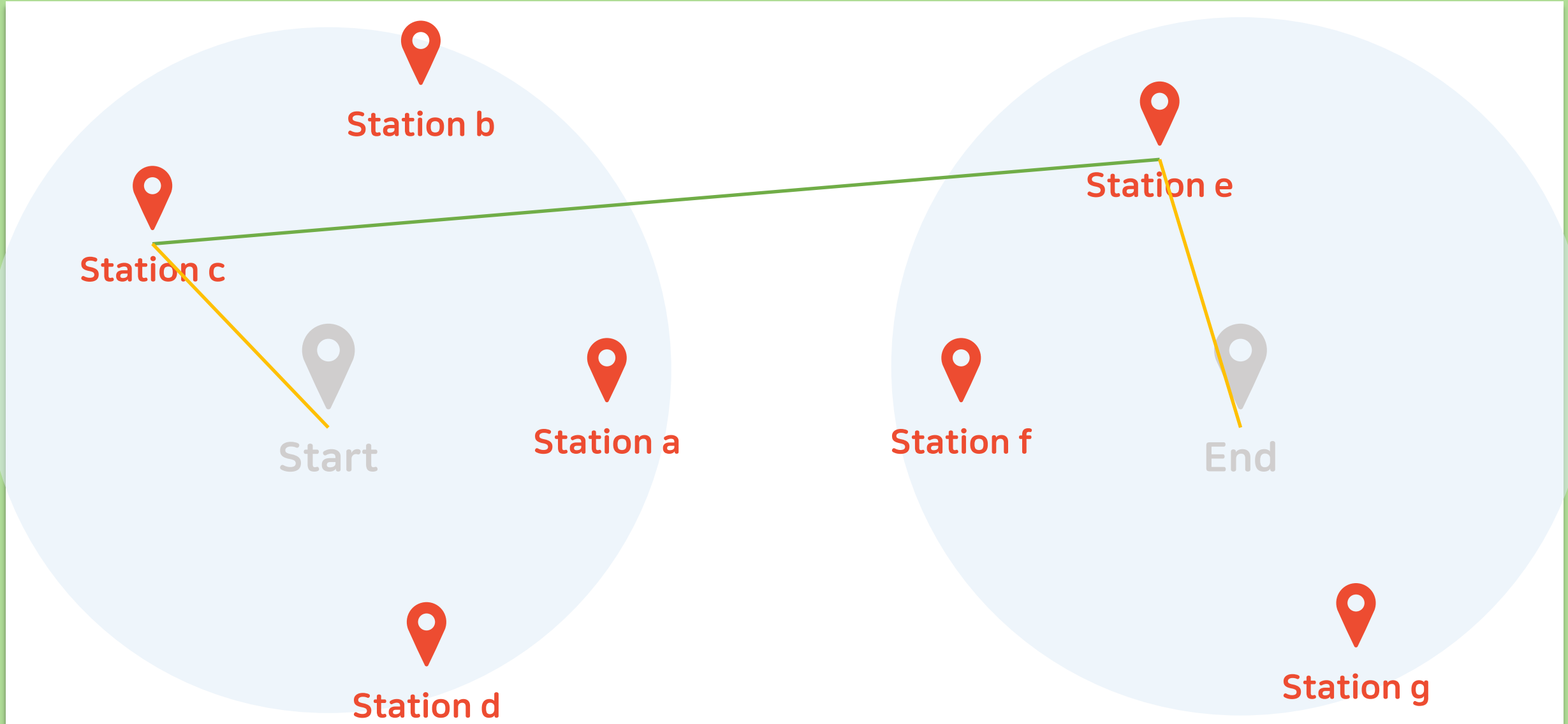
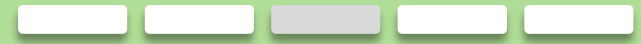
3. 세부 설계 사항

시스템 설계-길 찾기 알고리즘



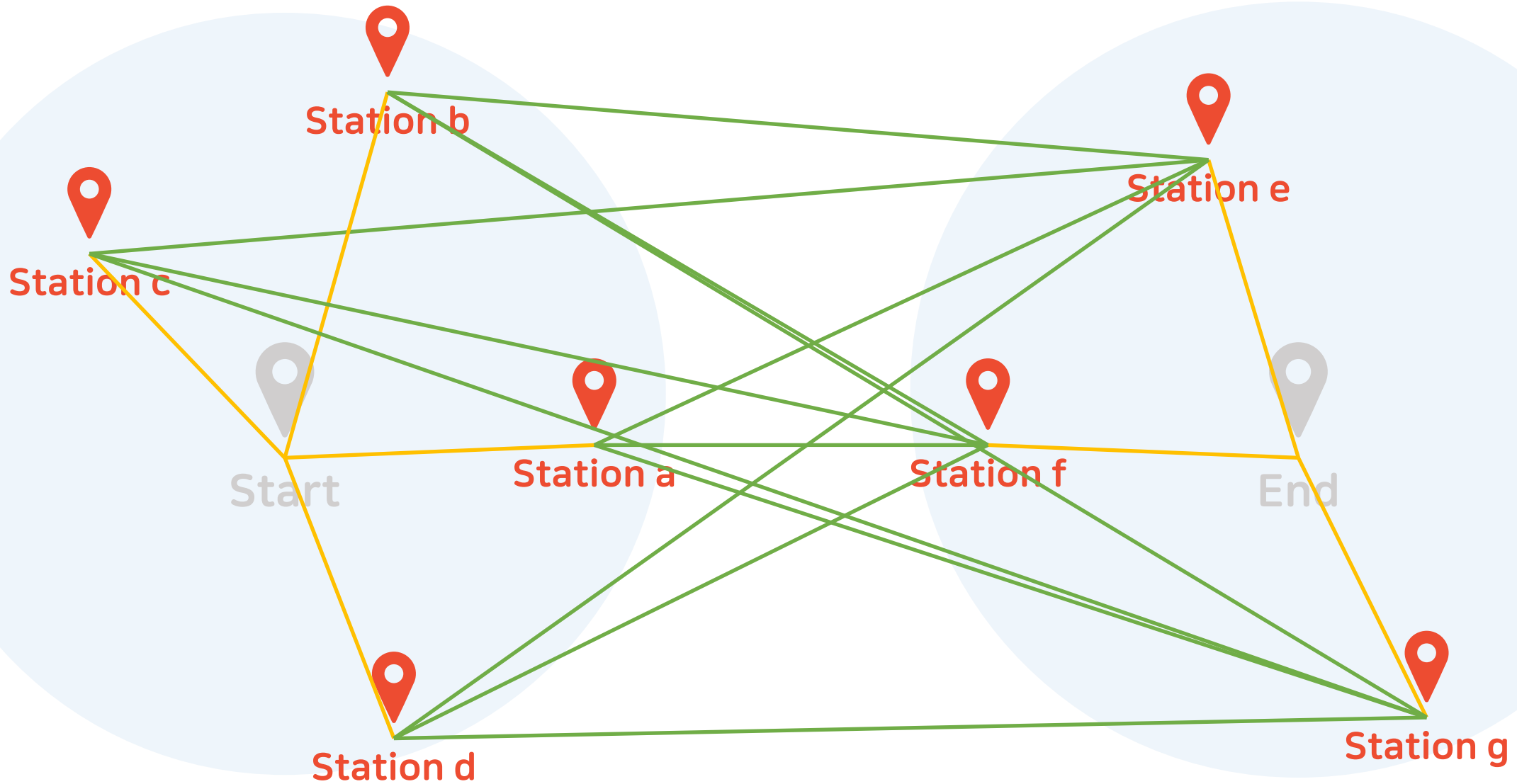
3. 세부 설계 사항

시스템 설계-길 찾기 알고리즘



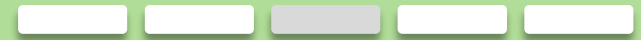
3. 세부 설계 사항

시스템 설계-길 찾기 알고리즘



3. 세부 설계 사항

시스템 설계-길 찾기 알고리즘



Estimated Time

Real Time



1500



1600



1800



1900



2200



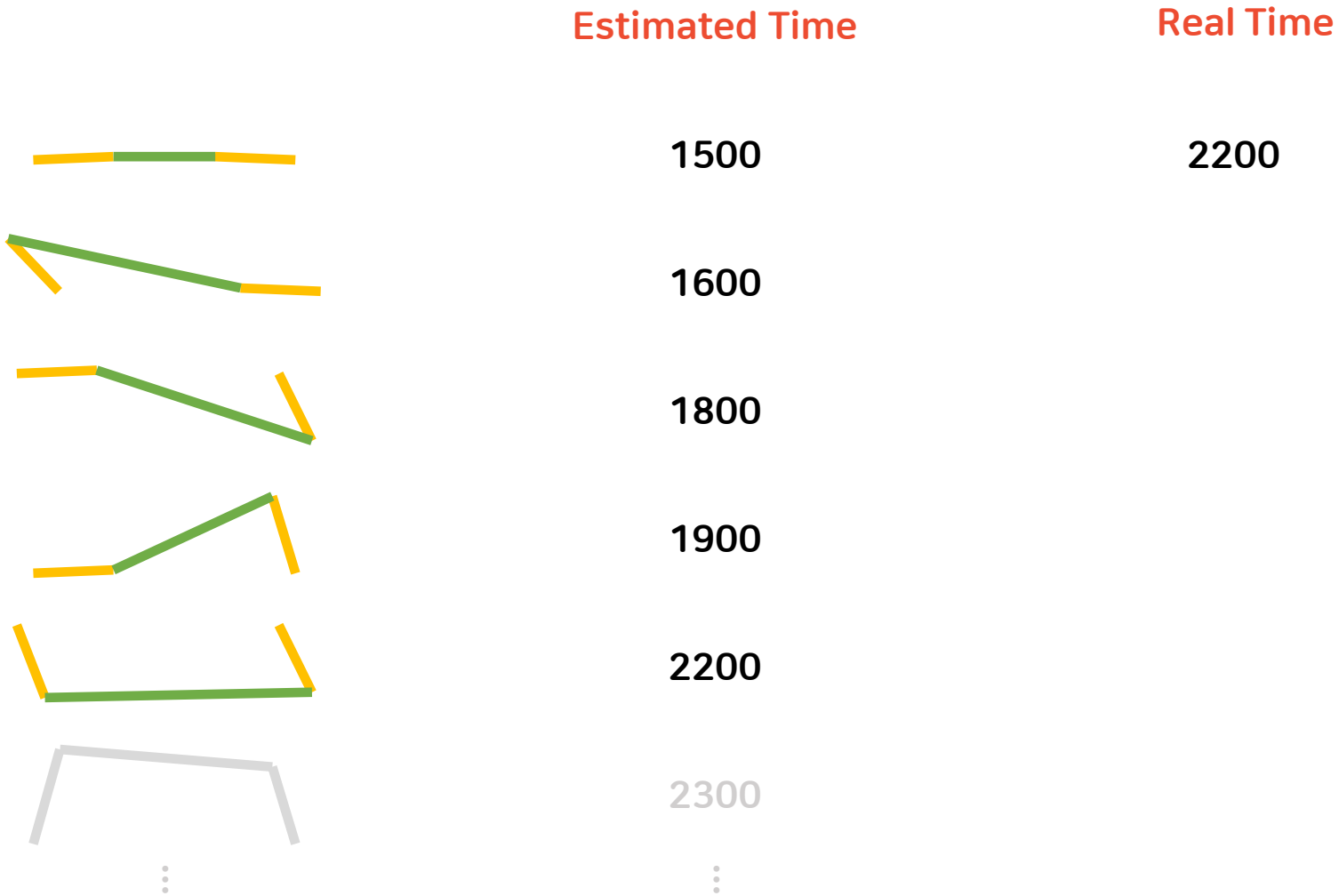
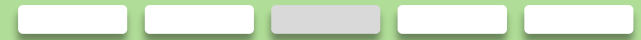
2300

⋮

⋮

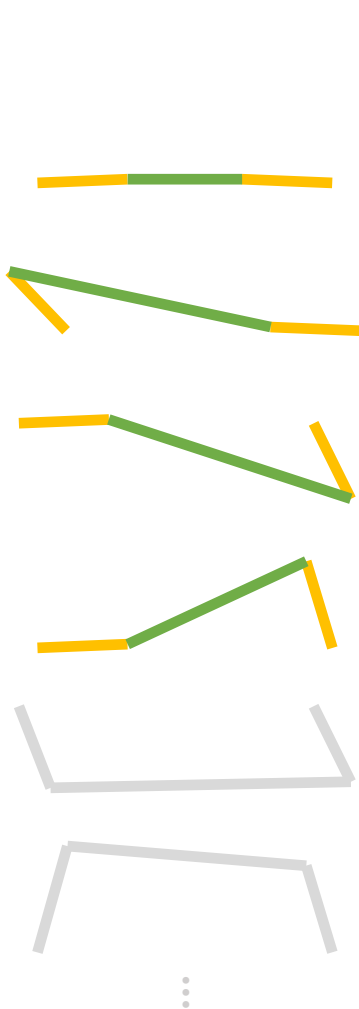
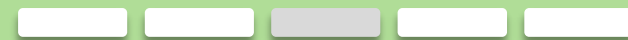
3. 세부 설계 사항

시스템 설계-길 찾기 알고리즘



3. 세부 설계 사항

시스템 설계-길 찾기 알고리즘



Estimated Time

Real Time

1500

2200

1600

2100

1800

1900

2200

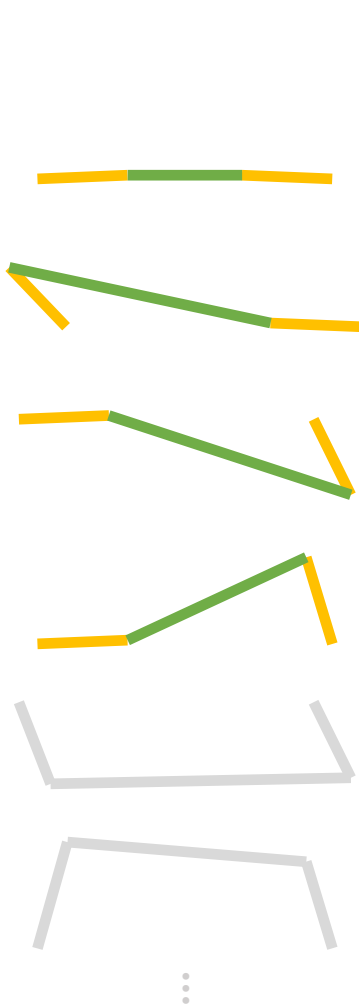
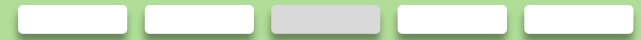
2300

⋮

⋮

3. 세부 설계 사항

시스템 설계-길 찾기 알고리즘



Estimated Time

Real Time

1500

2200

1600

2100

1800

2700

1900

2200

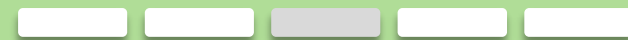
2300

⋮

⋮

3. 세부 설계 사항

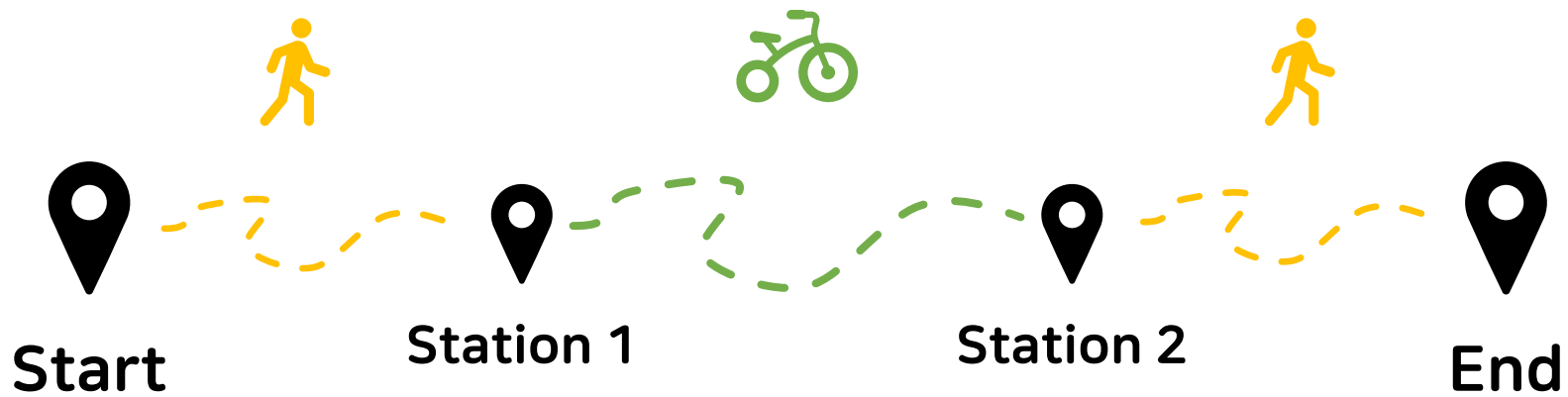
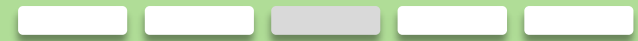
시스템 설계-길 찾기 알고리즘



	Estimated Time	Real Time
	1500	2200
	1600	2100
	1800	2700
	1900	3300
	2200	
	2300	
⋮	⋮	

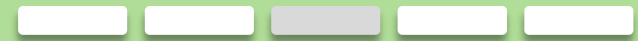
3. 세부 설계 사항

시스템 설계-길 찾기 알고리즘



3. 세부 설계 사항

시스템 설계-길 찾기 알고리즘



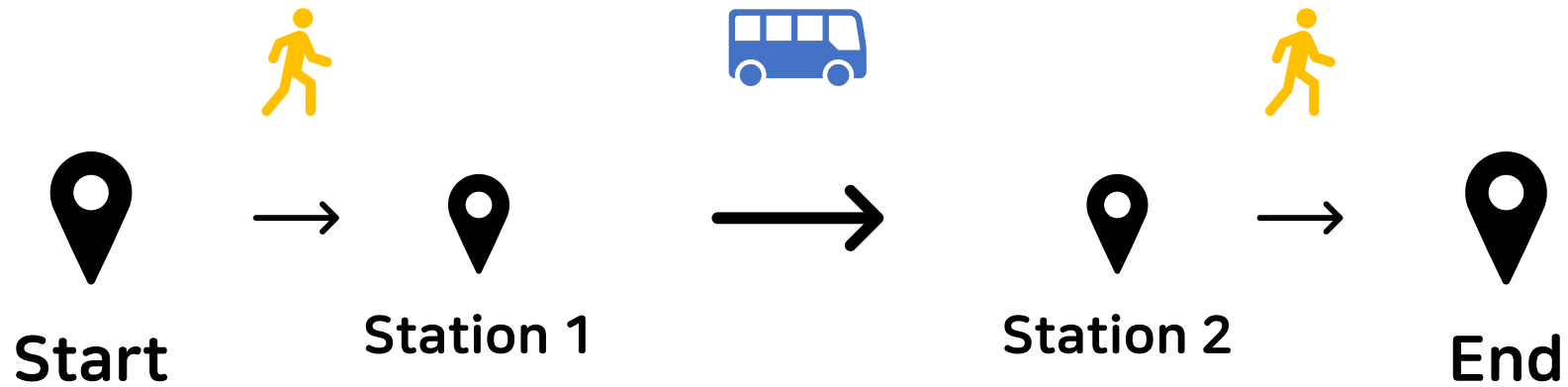
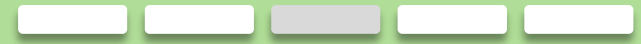

Start




End

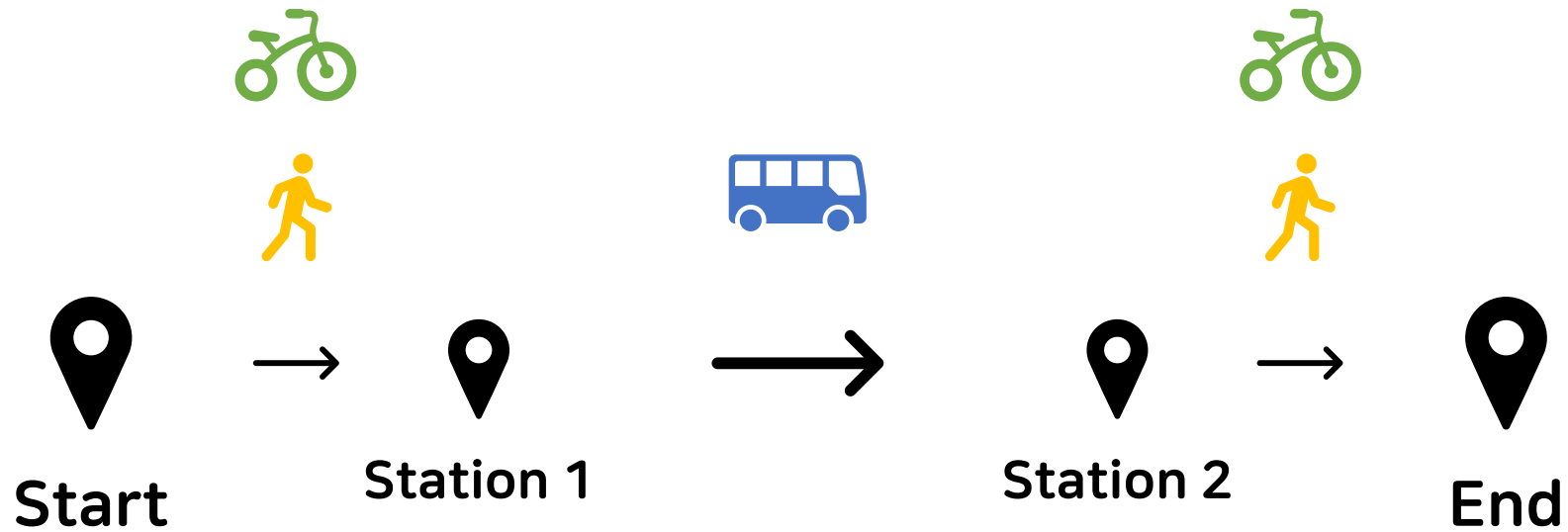
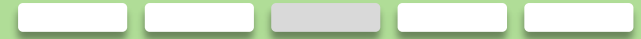
3. 세부 설계 사항

시스템 설계-길 찾기 알고리즘



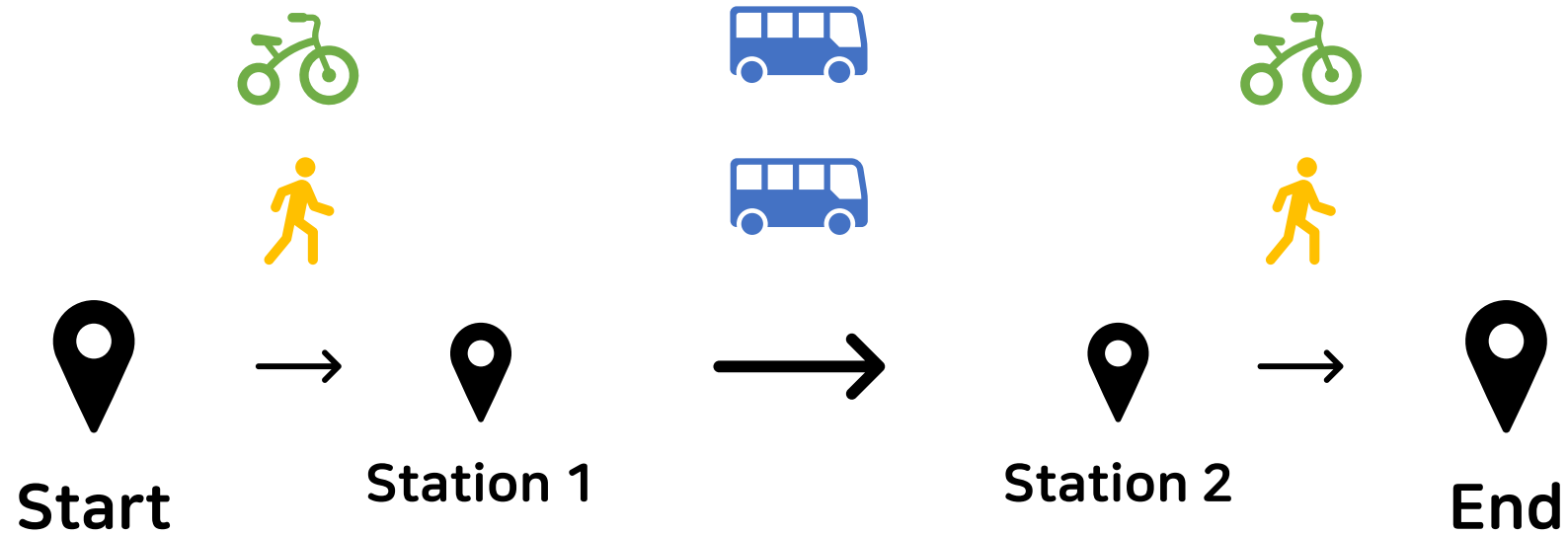
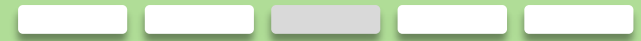
3. 세부 설계 사항

시스템 설계-길 찾기 알고리즘



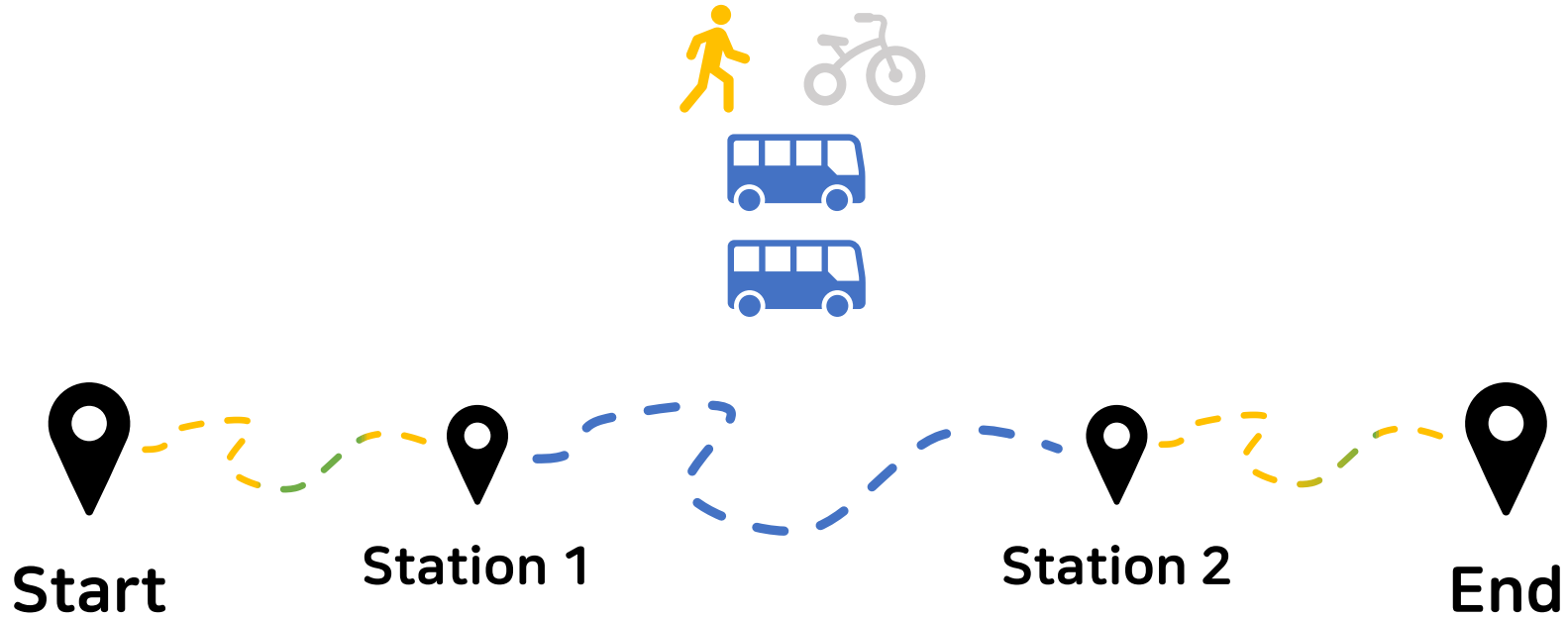
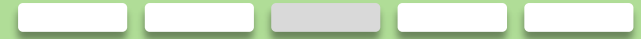
3. 세부 설계 사항

시스템 설계-길 찾기 알고리즘



3. 세부 설계 사항

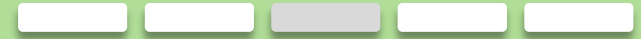
시스템 설계-길 찾기 알고리즘



Use TOPIS Engine

3. 세부 설계 사항

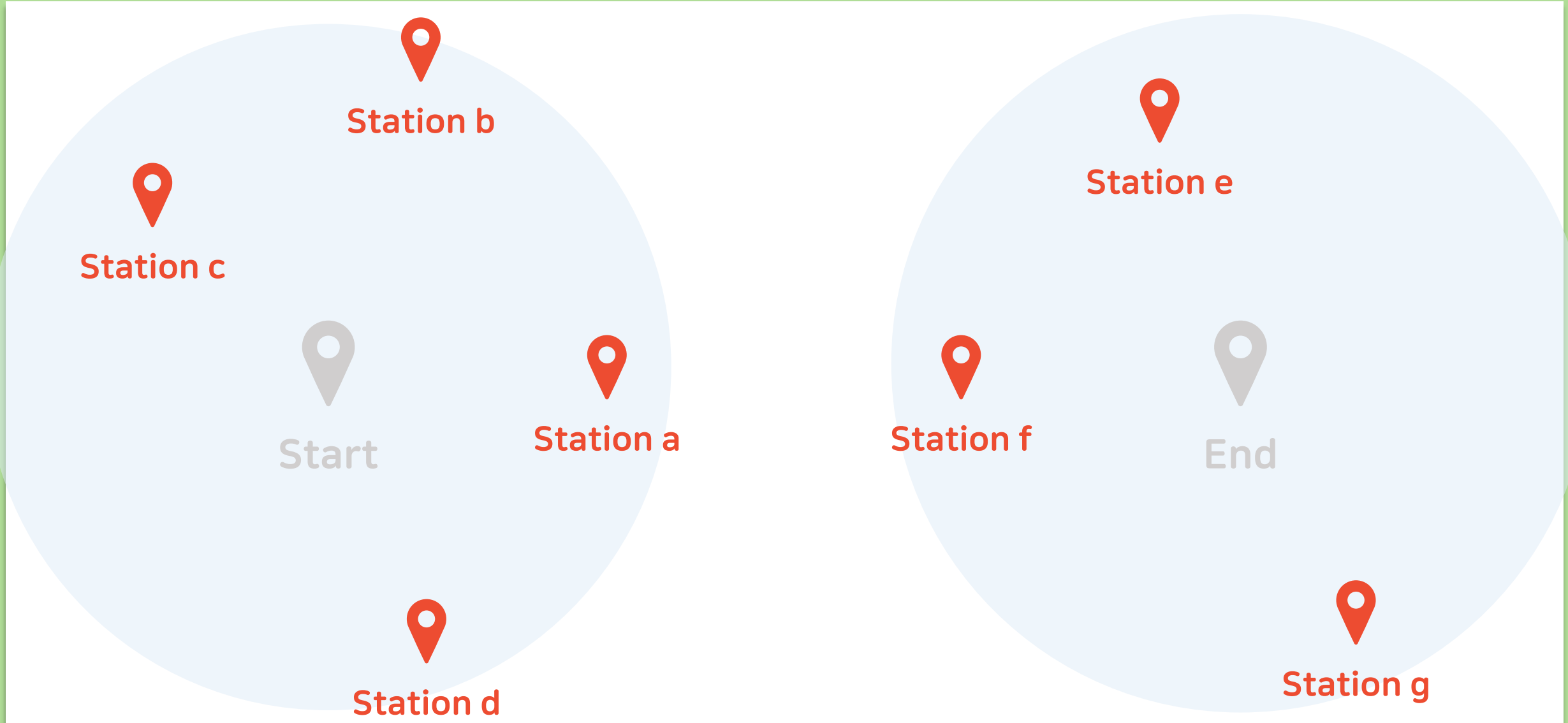
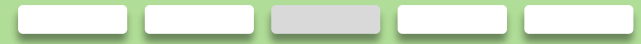
시스템 설계-길 찾기 알고리즘



How to select 2 stations?

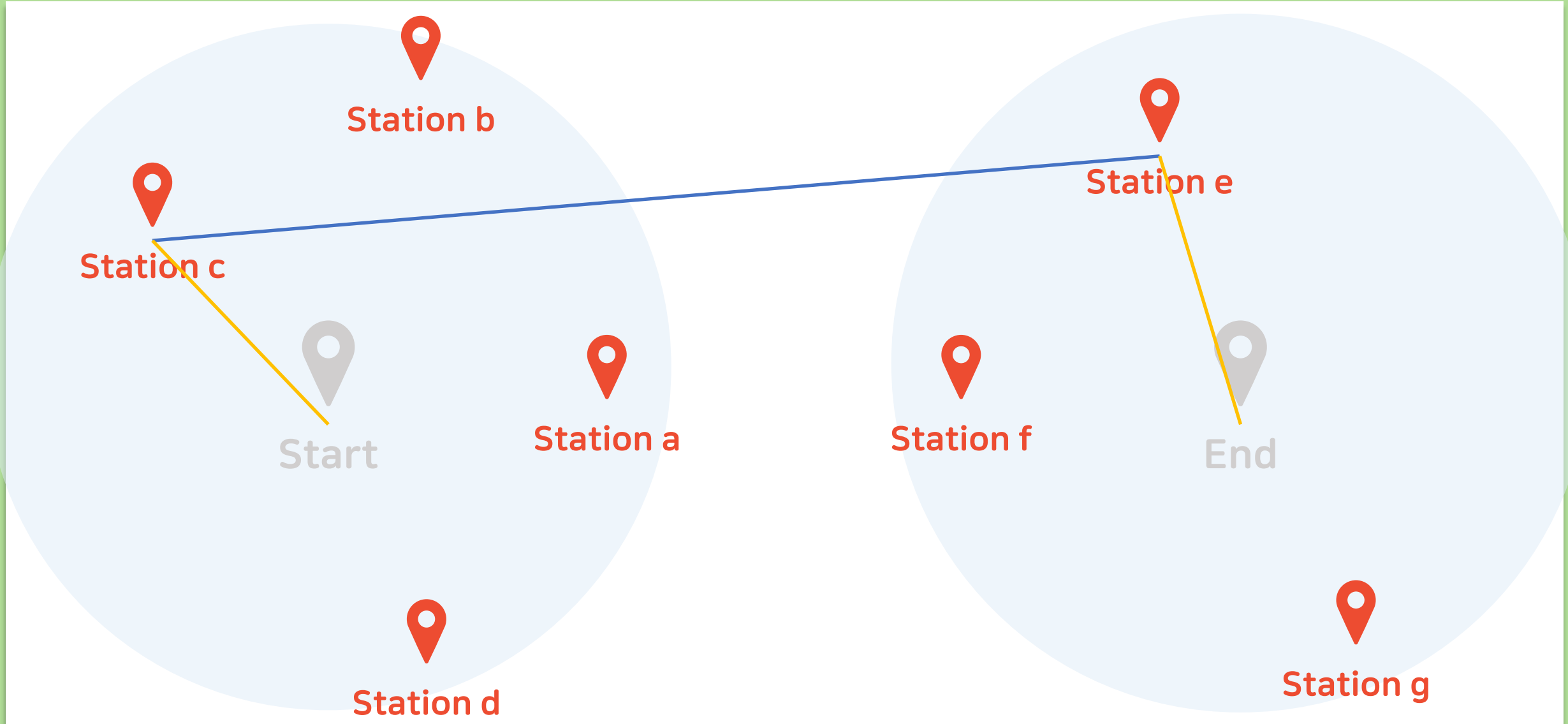
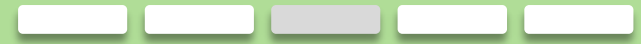
3. 세부 설계 사항

시스템 설계-길 찾기 알고리즘



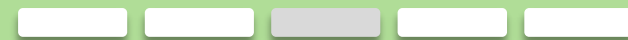
3. 세부 설계 사항

시스템 설계-길 찾기 알고리즘



3. 세부 설계 사항

시스템 설계-길 찾기 알고리즘



4. 결과 및 평가

로고 및 포스터



기본형



세로형

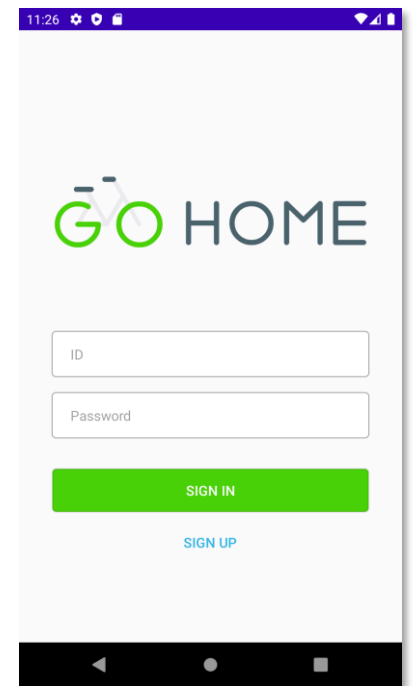
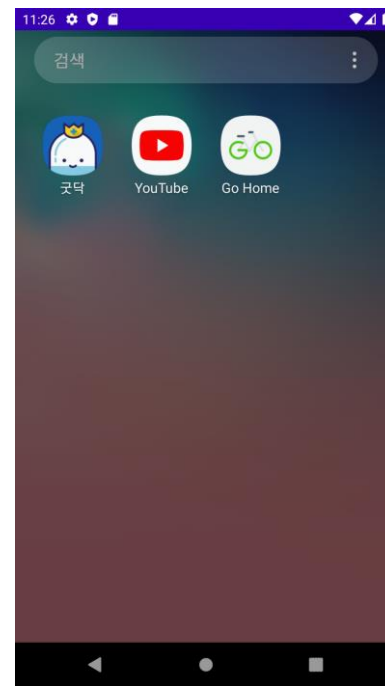
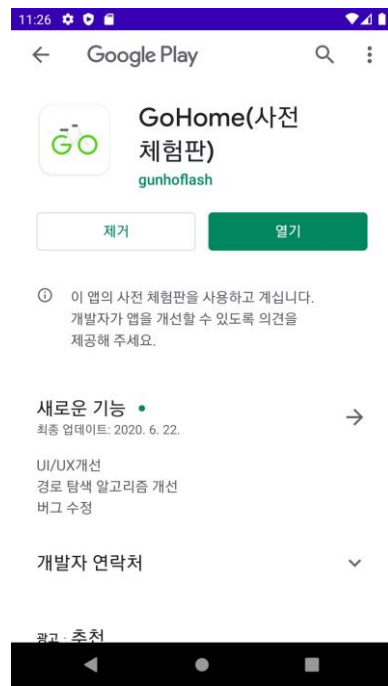


가로형



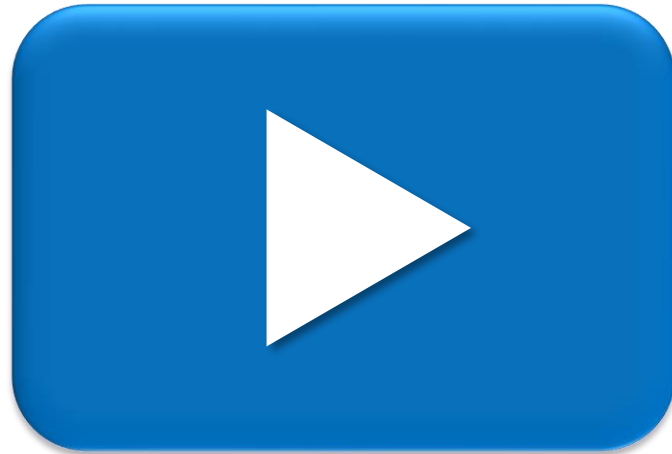
4. 결과 및 평가

설치 과정



4. 결과 및 평가

실행 영상



4. 결과 및 평가

결과물 정량평가



평 가 항 목	평 가 방 법	적 용 기 준	개 발 목표치	비중 (%)	평가결과
경로 탐색 정확도	경쟁 서비스와 최적 경로 소요시간 비교	첫 번째 최적 경로 소요시간, 총 10회 이상 평가	경쟁 서비스 대비 평균 110% 미만의 소요시간	30%	평균 100% 미만
경로 탐색 속도	소요 시간 측정	10회 이상 측정	평균 5초 미만	20%	평균 2.6초
Client-Gateway 간 통신 속도	소요 시간 측정	10회 이상 측정	평균 1초 미만	15%	평균 0.5초
Gateway-Server 간 통신 속도	소요 시간 측정	10회 이상 측정	평균 1초 미만	15%	평균 0.5초
AR 표현 정확성	나침반 앱과 비교하여 각도 차이 측정	3회 이상 측정	평균 $\pm 10^{\circ}$ 미만	10%	평균 $\pm 3^{\circ}$
앱 로딩 속도	화면 전환 시간 측정	10회 이상 측정	평균 1초 미만	10%	평균 1초 미만

4. 결과 및 평가

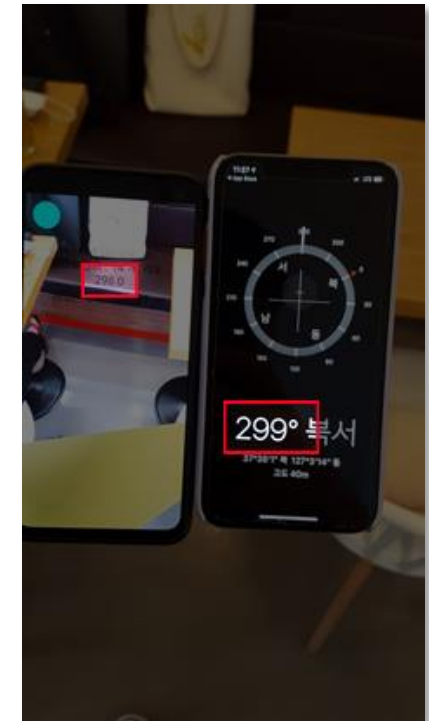
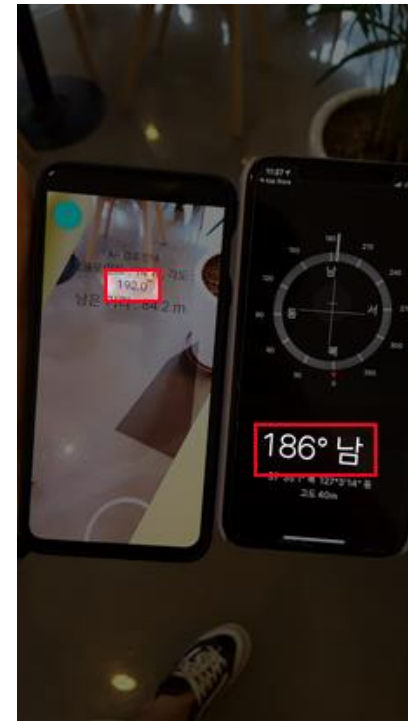
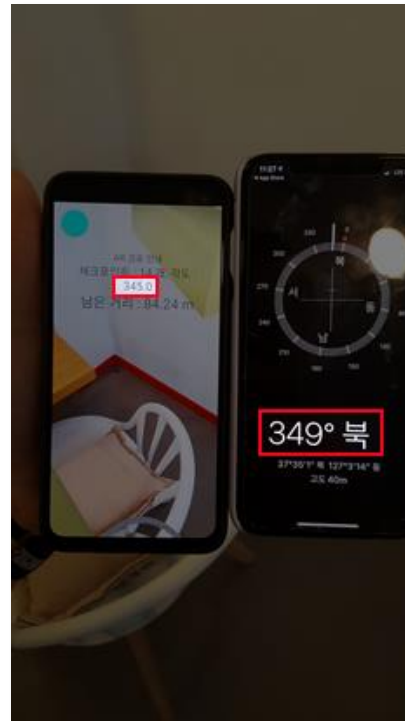
결과물 정량평가-경로 탐색

```
ubuntu@ip-172-31-44-58: ~
2020-06-23 20:27:13.546 +0900 [info]: (59ms) https://topis.seoul.go.kr/map/getRoute.do
2020-06-23 20:27:13.550 +0900 [info]: (60ms) https://topis.seoul.go.kr/map/getRoute.do
2020-06-23 20:27:13.557 +0900 [info]: (62ms) https://topis.seoul.go.kr/map/getRoute.do
2020-06-23 20:27:13.709 +0900 [info]: (206ms) https://topis.seoul.go.kr/map/getRoute.do
2020-06-23 20:27:13.876 +0900 [info]: (372ms) https://apis.openapi.sk.com/tmap/routes/pedestrian?version=1
2020-06-23 20:27:13.877 +0900 [info]: (375ms) https://apis.openapi.sk.com/tmap/routes/pedestrian?version=1
2020-06-23 20:27:13.881 +0900 [info]: (364ms) https://apis.openapi.sk.com/tmap/routes/pedestrian?version=1
2020-06-23 20:27:14.014 +0900 [info]: (151ms) https://apis.openapi.sk.com/tmap/routes/pedestrian?version=1
2020-06-23 20:27:14.164 +0900 [info]: (148ms) https://apis.openapi.sk.com/tmap/routes/pedestrian?version=1
2020-06-23 20:27:14.176 +0900 [info]: (168ms) https://apis.openapi.sk.com/tmap/routes/pedestrian?version=1
2020-06-23 20:27:14.179 +0900 [info]: (167ms) https://apis.openapi.sk.com/tmap/routes/pedestrian?version=1
2020-06-23 20:27:14.692 +0900 [info]: (172ms) https://apis.openapi.sk.com/tmap/routes/pedestrian?version=1
2020-06-23 20:27:14.696 +0900 [info]: (169ms) https://apis.openapi.sk.com/tmap/routes/pedestrian?version=1
2020-06-23 20:27:14.709 +0900 [info]: (187ms) https://apis.openapi.sk.com/tmap/routes/pedestrian?version=1
2020-06-23 20:27:15.037 +0900 [info]: (161ms) https://apis.openapi.sk.com/tmap/routes/pedestrian?version=1
2020-06-23 20:27:15.228 +0900 [info]: (185ms) https://apis.openapi.sk.com/tmap/routes/pedestrian?version=1
2020-06-23 20:27:15.563 +0900 [info]: (174ms) https://apis.openapi.sk.com/tmap/routes/pedestrian?version=1
2020-06-23 20:27:16.061 +0900 [info]: (164ms) https://apis.openapi.sk.com/tmap/routes/pedestrian?version=1
2020-06-23 20:27:16.250 +0900 [info]: (168ms) https://apis.openapi.sk.com/tmap/routes/pedestrian?version=1
2020-06-23 20:27:16.259 +0900 [info]: (178ms) https://apis.openapi.sk.com/tmap/routes/pedestrian?version=1
2020-06-23 20:27:16.263 +0900 [info]: (2856ms) routes
2020-06-23 20:27:16.533 +0900 [info]: (44ms) https://topis.seoul.go.kr/map/getRoute.do
2020-06-23 20:27:16.537 +0900 [info]: (43ms) https://topis.seoul.go.kr/map/getRoute.do
2020-06-23 20:27:16.540 +0900 [info]: (34ms) https://topis.seoul.go.kr/map/getRoute.do
2020-06-23 20:27:16.671 +0900 [info]: (169ms) https://topis.seoul.go.kr/map/getRoute.do
2020-06-23 20:27:16.672 +0900 [info]: (159ms) https://apis.openapi.sk.com/tmap/routes/pedestrian?version=1
2020-06-23 20:27:16.834 +0900 [info]: (305ms) https://apis.openapi.sk.com/tmap/routes/pedestrian?version=1
2020-06-23 20:27:16.983 +0900 [info]: (318ms) https://apis.openapi.sk.com/tmap/routes/pedestrian?version=1
2020-06-23 20:27:16.999 +0900 [info]: (332ms) https://apis.openapi.sk.com/tmap/routes/pedestrian?version=1
2020-06-23 20:27:17.160 +0900 [info]: (141ms) https://apis.openapi.sk.com/tmap/routes/pedestrian?version=1
2020-06-23 20:27:17.320 +0900 [info]: (148ms) https://apis.openapi.sk.com/tmap/routes/pedestrian?version=1
2020-06-23 20:27:17.325 +0900 [info]: (149ms) https://apis.openapi.sk.com/tmap/routes/pedestrian?version=1
2020-06-23 20:27:17.689 +0900 [info]: (158ms) https://apis.openapi.sk.com/tmap/routes/pedestrian?version=1
2020-06-23 20:27:17.704 +0900 [info]: (175ms) https://apis.openapi.sk.com/tmap/routes/pedestrian?version=1
```

경로 탐색 및 통신 속도 측정

4. 결과 및 평가

결과물 정량평가-AR



실제 나침반

157°

349°

186°

299°

계산 각도

156°

345°

192°

296°

평균 3° 오차

4. 결과 및 평가

어려웠던 점-FrontEnd



안드로이드 개발

UI 제작 어려움

JAVA 언어 숙련도 부족

서버와의 통신

Retrofit의 통신 오류

디버깅의 어려움

Manifest 변경으로 해결

4. 결과 및 평가

어려웠던 점-BackEnd



T map API 사용

T map API 요청 수의 제한
Web V2 API 가이드 부실
일부 기능의 부재

자전거 경로탐색

자전거 경로 API 부재
최적의 대여소 결정 어려움
소요시간 어림 사용

버스 경로탐색

N버스 경로 API 부재
그래프 표현의 성능 저하
ODSay API 사용

API Gateway Authentication 서버

Access token 난이도 높음
Refresh token 난이도 높음

4. 결과 및 평가

어려웠던 점-AR



Quaternion 이해

Quaternion 개념 이해
Euler Angles와의 변경
Quaternion의 적용

방위각 계산

현재-목표 위치 방위각 계산
가속도 센서의 오차
가속도 센서 값 보정으로 해결

손전등 기능 적용

Camera2 객체 구현 불가
Torch 클래스 사용 불가
ARCore에서 지원하지 않음

5. 향후 과제

차후 구현할 내용



버스와 지하철 막차, 첫차 알림

AR 도보 안내 화면에서 지도 표시

외부 독립적인 버스 경로 탐색 알고리즘 작성

HTTP/2 통신

더 자세한 위치 공유 기능

감사합니다