

멋쟁이사자처럼 at 숙명여대

# Meet Wizard

해커톤 중간점검

Manna조

김소은 문다연 박혜준 장한빛

# 목차

## FRIEND/GROUP

친구, 그룹 관리

## SCHEDULE/LOCATION

스케줄, 위치 데이터 입력 및 저장

## 회원가입 form

회원가입

## STEP1

그룹 선택

## STEP2

나의 시간표/ 출발위치 확인

<아이디어톤 이후 개발한 것들>

<앞으로 할 일>

## 중간지점 도출 과정

그라함 스캔 알고리즘  
다각형의 무게중심 알고리즘  
다익스트라 알고리즘  
중간지점 변경 알고리즘

## 약속시간 도출 알고리즘

# FRIEND/GROUP

## 친구관리 및 친구조회

아이디 검색을 통해 친구 요청을 보낼 수 있고,  
나에게 온 친구 요청을 수락하거나  
거절할 수 있다.

친구 관리

내가 보낸 친구 요청

테스트2 (test2)

나에게 온 친구 요청

문다연 (user)

테스트1 (test1)

수락

거절

수락

거절

친구 조회

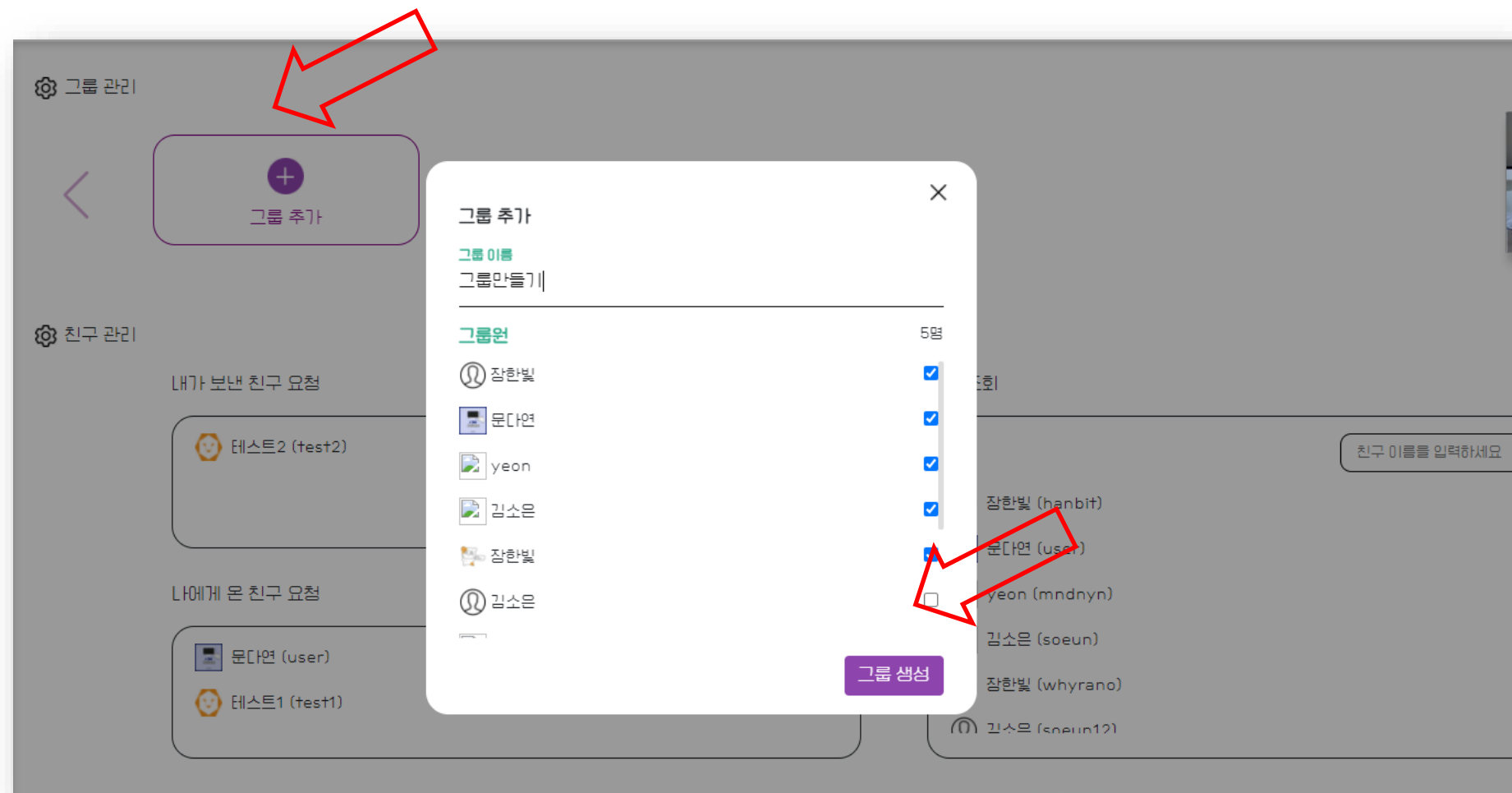
테



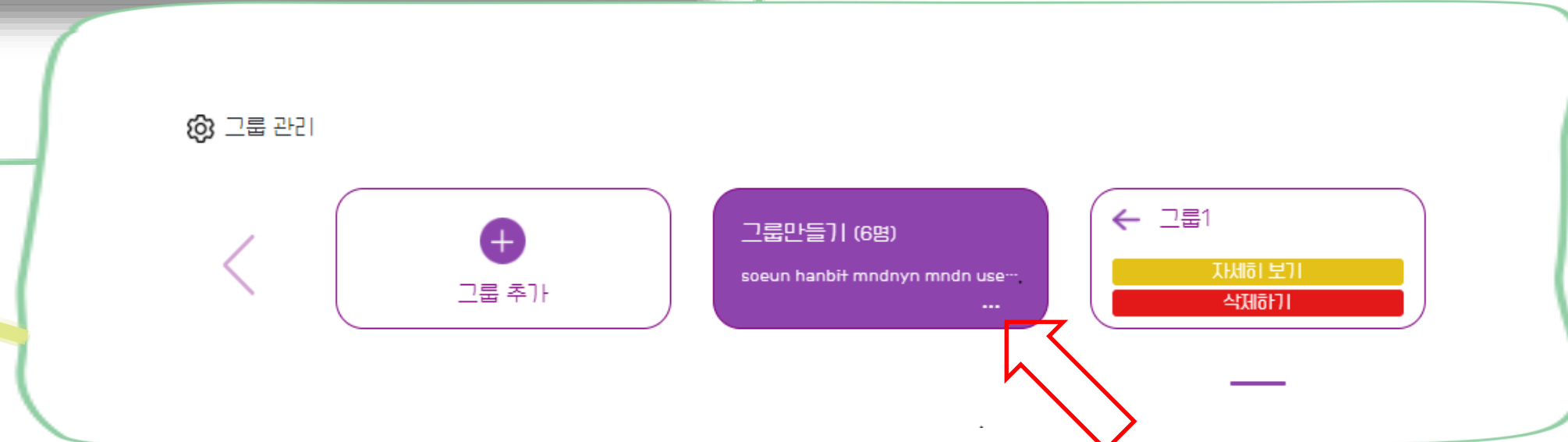
테스트1 (test1)

위에서 수락한 친구는 친구조회창에 보이며,  
아이디 일부만 검색하여 쉽게 찾을 수도 있다.

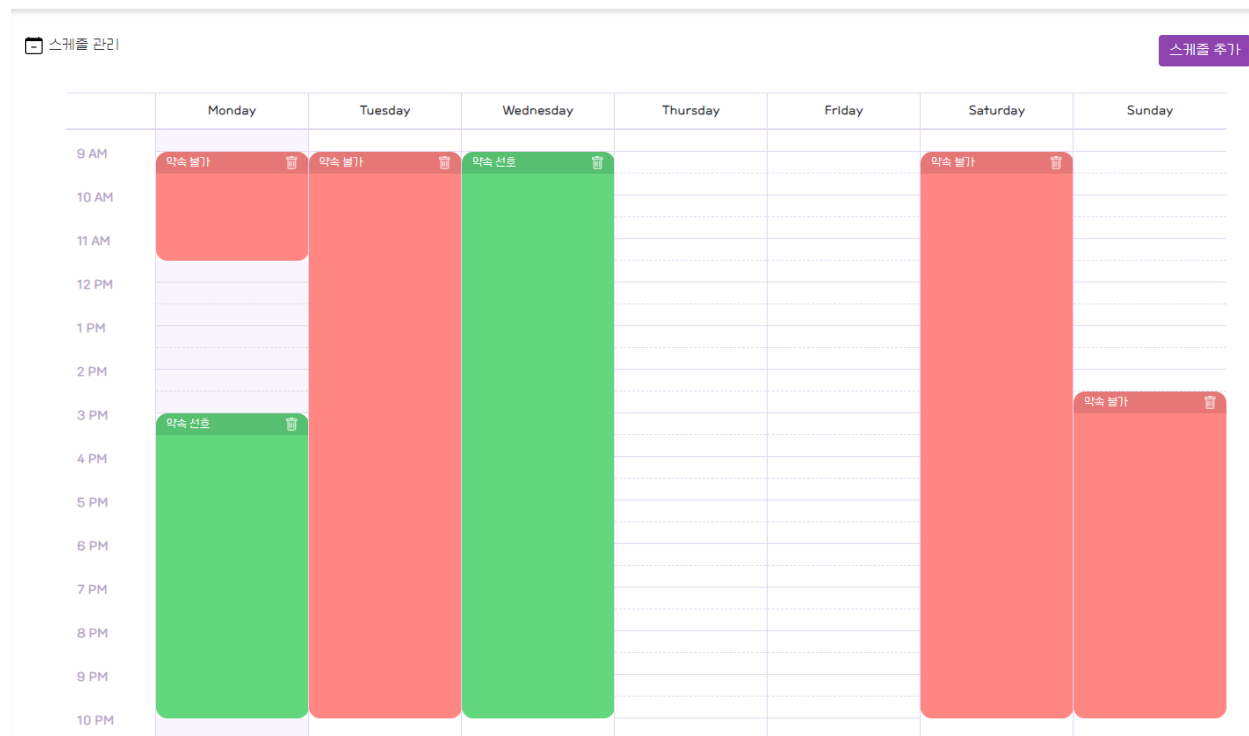
# FRIEND/GROUP



- 그룹추가 버튼을 눌러, 그룹 이름과 그룹원을 선택하여 그룹을 생성할 수 있다.
- 생성된 그룹은 그룹관리 탭에 추가된다.
- ... 버튼을 눌러 그룹원을 자세히 보거나, 해당 그룹을 삭제할 수 있다.

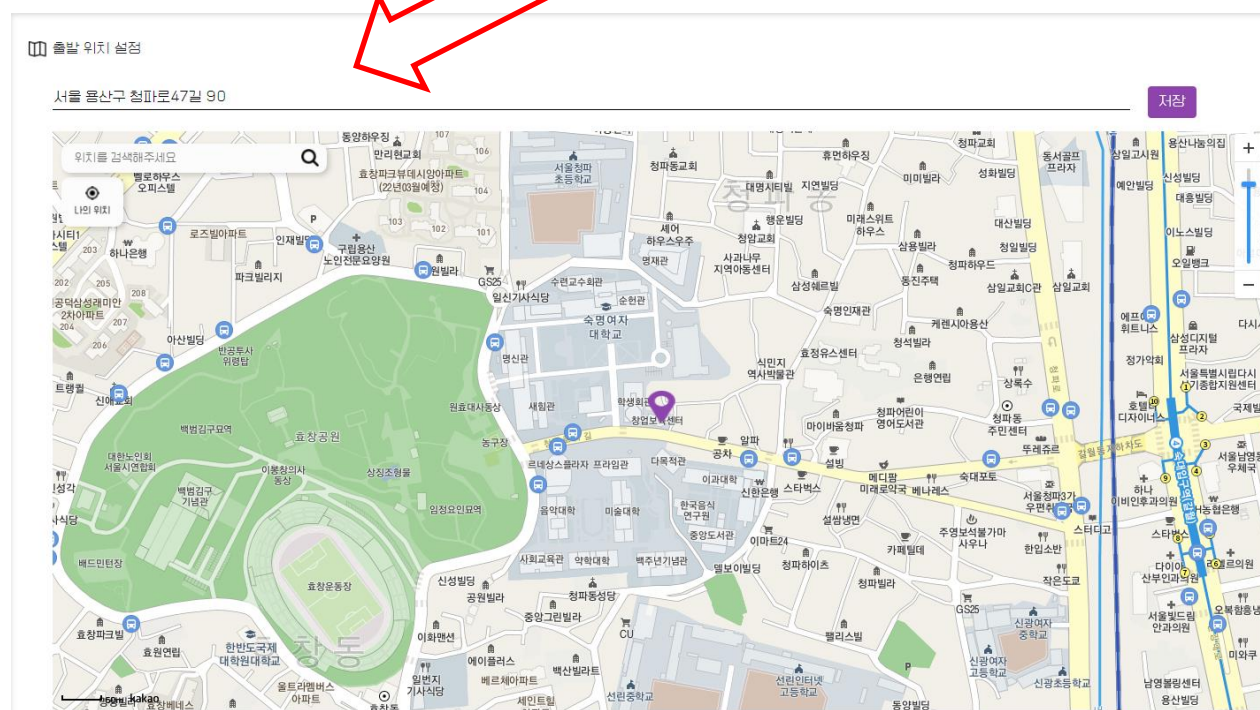


# SCHEDULE/LOCATION



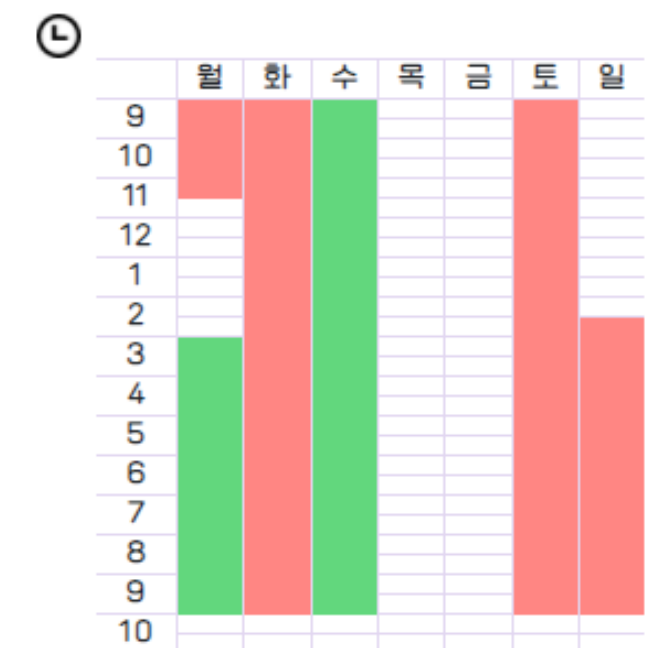
> schedule과 location 탭에서  
스케줄과 위치를  
저장할 수 있다.

> 저장된 정보는  
좌측 상태바에  
그대로 저장된다.



mndn

서울 용산구 청파로47길 90



# 회원가입 form

로그인하세요.

로그인 후, 이용 가능합니다.

월

화

수

목

금

토

일

9

10

11

12

13

14

15

16

17

18

19

20

21

22

→

⚙

회원가입

Username:

사용자 ID

Email:

이메일 형식으로 입력하세요.

Name:

사용자 이름

Date of birth:

'yyyy-mm-dd' 형식으로 입력하세요

Tel:

'-'없이 입력하세요.

Image:

파일 선택

선택된 파일 없음

비밀번호:

비밀번호 확인:

가입

- 로그인되지 않은 상태에서 홈페이지 좌측 메뉴바의 톱니 아이콘을 누르면 회원가입 화면으로 이동한다.
- 회원가입 입력 요소
  - Username: 사용자 ID의 기능을 한다.
  - Name: 사용자의 이름이다.
  - 요소별 특성에 따라 Placeholder를 통해 입력 형식을 요구한다.
- Image 입력란에 '파일 선택'을 누른 후 이미지를 업로드하면, 프로필 사진의 미리보기가 보여진다.



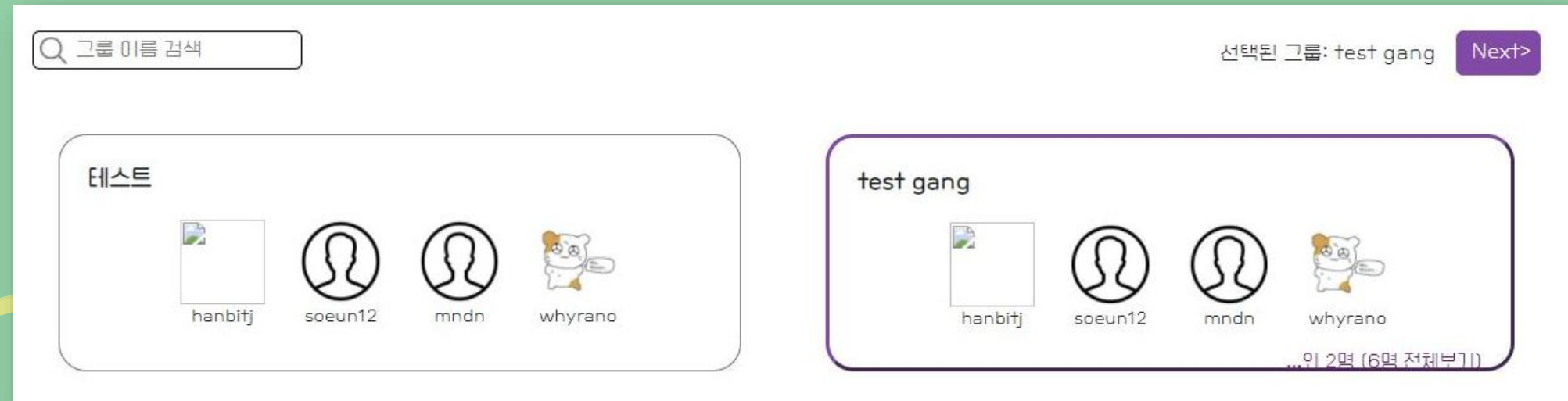


# STEP1(그룹선택)

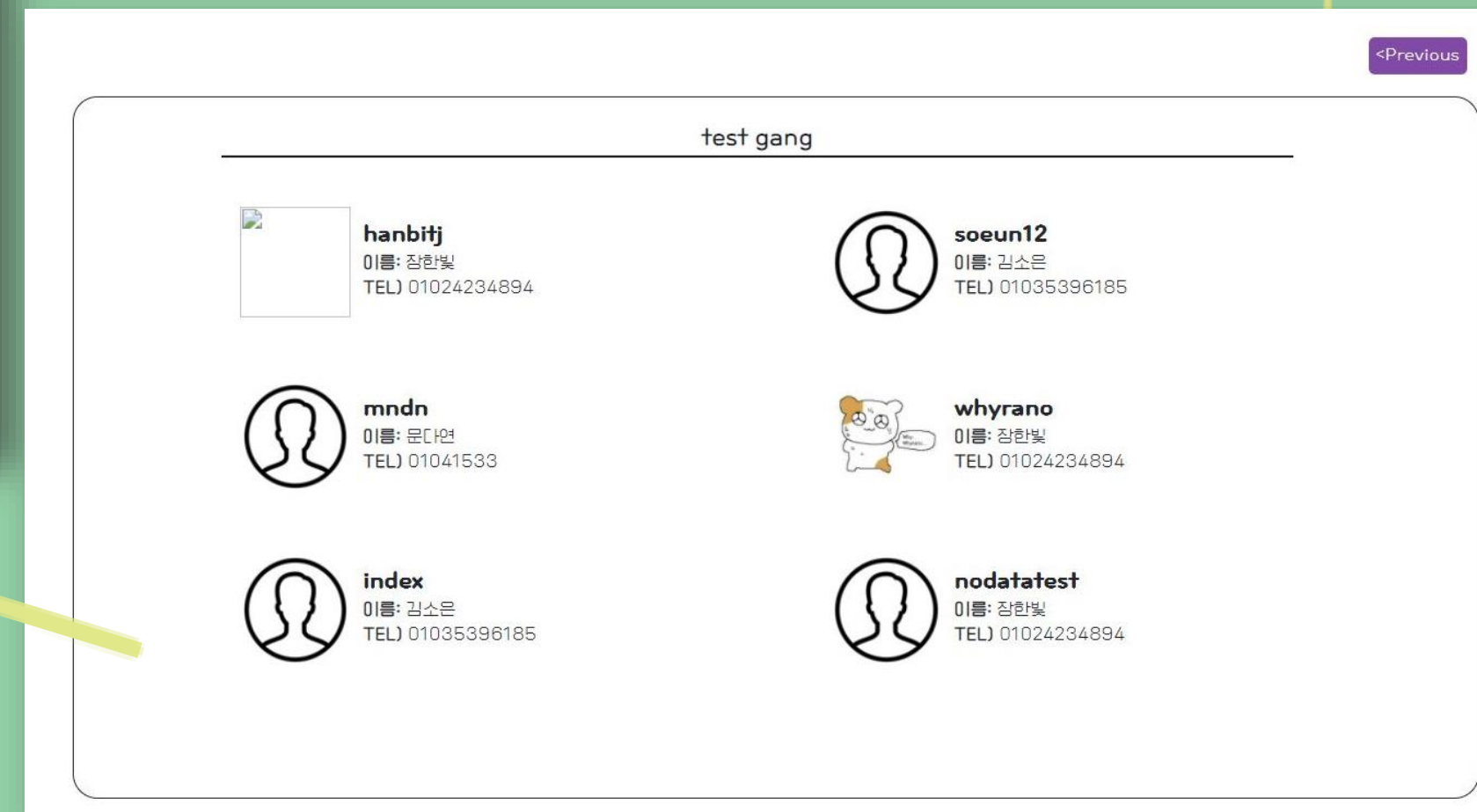
STEP1~STEP4의 서비스는 로그인한 회원에게만 제공됩니다.

그룹이 선택되었음이 보라색 테두리로 표현된다.

사전에 만들어져 있는 그룹 중 하나를  
선택한다.



4명 이상의 그룹에 생긴 '...인 n명(~명 전체보기)'  
링크를 클릭할 시, 멤버 전원의  
Username(아이디), 이름, 전화번호가 표현된다.



# STEP 2 (나의 시간표, 출발위치 확인)

skip 버튼에 인접한 물음표 아이콘에  
마우스 커서를 올리면  
skip 기능에 대한 설명창이 나타난다.

whyrano님의 시간표

?

skip

수정

약속장소만 맞춰보고 싶을 때,  
나의 시간표 설정을 스킵하세요!

	월	화	수	목	금	토	일
9							
10							

whyrano님의 출발위치

?

skip

수정

약속시간만 맞춰보고 싶을 때,  
나의 출발위치 설정을 스킵하세요!

whyrano님의 시간표

?

skip

수정

	월	화	수	목	금	토	일
9							
10							
11							
12							
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							

whyrano님의 출발위치

?

skip

수정

서울 용산구 한강대로 지하 306

유저의 시간표 설정을 skip했을 때,  
취소 시 시간표가 다시 나타난다.

☆ 탐색 결과 찾기 - STEP 2

<Previous

Next>

whyrano님의 출발위치

?

skip

수정

서울 용산구 한강대로 지하 306

> whyrano님의 시간표 설정을 건너뛰었습니다.

취소

유저의 출발위치 설정을 skip했을 때,  
취소 시 지도 api가 다시 나타난다.

☆ 탐색 결과 찾기 - STEP 2

<Previous

Next>

whyrano님의 시간표

?

skip

수정

	월	화	수	목	금	토	일
9							
10							
11							
12							
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							

> whyrano님의 출발위치 설정을 건너뛰었습니다.

취소



# 중간지점 도출 과정

## 그라함 스캔 알고리즘

그라함 스캔 알고리즘을 통해 사용자 위치들의 외곽 점들로 이루어진 다각형을 구한다.

## 다각형의 무게중심

다각형의 무게중심을 구해서 처음 중간지점으로 삼는다.

## 다익스트라 알고리즘

각 사용자 위치와 중간지점과의 최단 경로를 다익스트라 알고리즘을 통해 구한다. 지하철 데이터를 이용하며, 경로 값은 지하철 노드 개수로 한다.

## 중간지점 도출 완료!

도출한 결과를 step4에서 보여준다. 그 지점과 관련된 스팟들을 카테고리별로 보여준다.

## 새로운 중간지점으로 이동

중간지점에서 사용자 위치까지의 단위벡터와 소요시간을 고려하여 새로운 중간지점 도출한다. 만약 그라함 스캔 알고리즘으로 구한 다각형을 벗어난다면 벡터 값을 줄여서 다시 구한다.

부적합

## 중간지점으로 적합한지 검사

모든 사용자의 위치에서 중간지점까지 소요시간을 비교했을 때, 최대, 최소 통행 비용의 차이가 설정한 오차 내에 속하면 중간지점으로 선정한다.

적합

# 중간지점 찾기 전제

## 전제1.

한 지하철역에서 다음 지하철역까지의 이동 시간은 동일하다.

## 전제2.

각 사용자의 위치에서 가장 가까운 지하철역을 기반으로 중간지점을 산출하기 때문에 지하철이 발달한 수도권에서 약속장소를 찾는 것을 전제로 하며, 지하철이 발달된 지역이 아닐 시 위도/경도를 이용한 중간지점 좌표 값만 산출된다.

## 전제3.

최종 산출되는 중간지점은 지하철역이며, 여타의 교통수단은 고려하지 않는다.

# 1. 그라함 스캔 알고리즘

그라함 스캔 알고리즘은 평면상에서 유한한 점들의 볼록 껍질을 찾는 방법이다. 유한개의 점들 중 다른 점을 가둘 수 있는 외곽점을 찾는 알고리즘 중 하나로, 이 외곽점을 이으면 볼록 껍질(Convex hull)이 된다. 이번 프로젝트에서는 이 점들을 각 사용자의 위치와 가장 가까운 지하철역들의 좌표로 설정한다.

## 알고리즘

1. y 값이 가장 작은 점을 찾는다. (만약 여러 개 존재 시 x 값이 가장 작은 점을 선택한다) 이 때, 이 점을  $p_0$ 라 부른다.
2.  $p_0$ 를 기준으로 다른 모든 점의 각도를 구하여 각도가 작은 순서대로 정렬한다.
3.  $p_0$ 와 정렬된 점을 2개의 Convex hull에 추가한다.
4. 그 다음 점부터 다음 조건을 반복하여 수행한다.
  - 4-1. Convex hull의 마지막 직선에서 현재 점이 왼쪽에 있으면 Convex hull의 마지막 점을 Convex hull에서 제외한다. (현재 점이 4-2번 조건을 만족할 때까지 진행한다)
  - 4-2. Convex hull의 마지막 직선에서 현재 점이 오른쪽에 있으면 현재 점을 Convex hull에 추가하고 다음 점을 가져온다.
5. Convex hull을 이루고 있는 점을 이어준다.

## 2. 다각형의 무게중심 알고리즘

n각형에서 한 꼭지점과 이 꼭지점을 제외한 나머지 (n-1)개의 꼭지점으로 이루어진 다각형의 중선들의 중심을 연결한 선분을 n각형의 중선이라 부른다.

n각형의 모든 n개의 중선은 한 점에서 교차하며 교점은 각 중선을 (n-1):1로 분할한다.

이 점을 n각형의 중선들의 중심 또는 다각형의 무게중심이라 부른다.

$$Area = A = \frac{1}{2} \sum_{i=0}^{N-1} (x_i y_{i+1} - x_{i+1} y_i)$$

$$Centroid_x = \frac{1}{6A} \sum_{i=0}^{N-1} (x_i + x_{i+1}) (x_i y_{i+1} - x_{i+1} y_i)$$

$$Centroid_y = \frac{1}{6A} \sum_{i=0}^{N-1} (y_i + y_{i+1}) (x_i y_{i+1} - x_{i+1} y_i)$$

### 3. 다익스트라 알고리즘



다익스트라 알고리즘은 도로 교통망 같은 곳에서 나타날 수 있는 그래프에서 꼭짓점 간의 최단 경로를 찾는 알고리즘이다.

원래는 두 꼭짓점 간의 가장 짧은 경로를 찾는 알고리즘이지만, 일반적인 변형은 한 꼭짓점을 “소스” 꼭짓점으로 고정하고, 그래프의 다른 모든 꼭짓점까지의 최단경로를 찾는 알고리즘으로 최단 경로 트리를 만드는 것이다.

이번 프로젝트에서는 위의 한 꼭짓점을 최초 중간지점 위치로 고정하고, 그래프의 다른 모든 꼭짓점들을 각 사용자 위치와 가장 가까운 지하철역 위치로 하여 각 사용자 위치에서 중간지점까지의 최단경로를 찾는 용도로 다익스트라 알고리즘을 이용한다.



## 4. 중간지점 변경 알고리즘

중간지점 선정을 위한 탐색 과정은 노드들의 좌표 평균인 곳에서 시작한다.

먼저, 현재 위치한 중간지점이 합리적인지 확인하여(유효성 검사), 합리적이지 않은 경우 다음에 중간지점이 위치할 곳을 구한다.

### 알고리즘

1. 중간지점을 시점으로 보고 한 노드를 종점으로 보았을 때 생기는 벡터를 단위 벡터로 만들고, 시간 가중치(노드의 합)와 곱한 값을 모두 더한다.
2. 그 후, 노드의 개수와 특정 상수로 나누어 다음 중간지점이 위치할 곳에 대한 정보를 가진 벡터를 생성하고, 현재 위치한 중간지점에 더하여 중간지점을 이동한다.
3. 상수는 벡터의 크기를 조절하는 역할로 중간지점 이동 범위를 조절한다.

식:

$$\overrightarrow{center_{i+1}} = \overrightarrow{center_i} + \frac{\sum_{j=1}^n (time W_j \times \overrightarrow{user_j})}{n \times \alpha}$$

여기서 center는 중간지점, timeW는 시간 가중치(노드 개수 합)를 시간 가중치들의 평균으로 나눈 값, user는 노드와 중간지점의 단위벡터, n은 노드들의 총 개수, α는 벡터의 크기를 결정할 상수이다.

# 약속 시간 도출 알고리즘



각 사용자의 약속 불가 시간(분홍색 스케줄)을 합하여 공통의 시간표에 약속 불가 시간으로 출력한다.



각 사용자의 약속 선호 시간(초록색 스케줄)의 교집합인 시간대를 파악하여 공통의 시간표에 약속 선호 시간으로 출력한다.



약속 시간 도출

우리 지금 만나

