

선견지명'S 프로젝트

ABLE

팀원 : 백지명, 고예성, 고예진,
송태인, 안영준

선견지명's 프로젝트

+

프로젝트 순서

첫번째

제작 배경

두번째

개발 기간

세번째

IA(Information Architect)

네번째

서비스 활용방안 및 기대효과(1) (2)

다섯번째

개발 툴 및 참고문헌

여섯번째

핵심 코드

일곱번째

화면 및 기능 설명

ABLE

선견지명'S 프로젝트



전 페이지

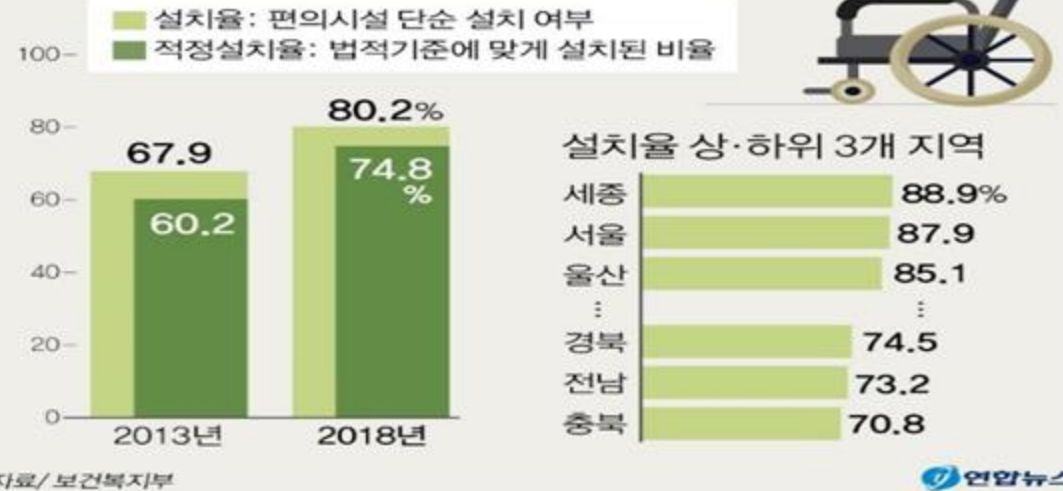
→ 시작 페이지

다음 페이지

→ 제작 배경 페이지

제작 배경

장애인 편의시설 설치 현황



장예진 기자 / 20190226 트위터 @yonhap_graphics 페이스북 tune.kr/LeYN1

지하철역 부딪침 사고 시각장애인의 하소연

점자블록 따라 걷다 어르신과 충돌 '안5천원 배상'
너무나 졸랐던 점자블록과 우대권 발급기 간격 때문

지하철 기자 입력 2022.03.04 11:15

본문 9줄 0 줄

한 나그네가 캄캄한 밤길을 걸어가고 있었다. 낯선 길인데 길이 험하여 걸어가기가 매우 힘들었다. 나그네가 조심조심 더듬거리고 있는데 뜻밖에도 앞쪽에서 등불이 반짝이는 게 보였다. 반가운 마음에 등불 가까이 다가간 나그네는 깜짝 놀랐다. 등불을 든 사람이 시각장애인이기 때문이었다.

나그네는 "알을 보지 못하는 분이 왜 등불을 들고나오셨습니까?"하고 물었다. 그러자 그 시각장애인이 말하기를 "나는 등불이 필요 없지만 다른 사람에게는 도움이 될 것이기에 들고나왔지요."라고 대답했다. 등불을 든 시각장애인 이야기는 할무드에 나오는 이야기라고 한다.

캄캄한 밤길에 시각장애인이 등불을 든 것은 자신을 위해서가 아니라 타인을 위한 것이라고 했다. 그런데 할무드는 2천년 전의 이야기다. 2천년이 지난 현대에서도 시각장애인의 등불 같은 역할을 하는 것이 있으니 바로 흰지팡이다.

시각장애인이 길을 갈 때는 흰지팡이를 짚고 간다. 흰지팡이는 할무드에 나오는 시각장애인의 등불처럼 어두운 곳에서도 잘 보이는 야광 띠가 붙어 있어서 밤에도 잘 보인다.

장애인도 안전하게...서울지하철, 교통약자 이용 지원 확대

보안팀 기자 입력 2022.04.11 15:08

발빠짐 방지등신 안내시설 설치...수어통역 서비스도



서울교통공사 서울교통공사 본사 (2022.04.11)

서울교통공사가 장애인 등 교통약자가 지하철을 안전하게 이용할 수 있도록 발빠짐 방지 시설과 바닥 동선 안내 표시를 일부 역에 시범 설치했다. 청각장애인을 위한 수어 통역 서비스와 시각장애인을 위한 이동 안내도우미 서비스도 제공 중이다. 이들 시설과 서비스의 효과가 확인되면 적용 역을 확대할 계획이다.



서비스 제작 배경

- 편의시설이란 장애인,노인,임산부 등 모든 사람들의 이동과 편의를 향상시키기 위해 공공장소, 상업시설, 교통수단 등 어디에나 존재하고, 사회적으로도 편의시설을 증축하는 추세이다.
- 그러나 편의시설의 설치 현황에 비해 사용자들이 해당 시설을 이용하기 위한 정보 안내가 부족하여 이용에 있어 어려움이 종종 발생한다.
- 이러한 상황을 해결하기 위해 'Able'앱을 기획하였다. Able 앱을 사용하면 편의시설에 대한 정보를 쉽게 얻을 수 있으며, 이를 통해 편의시설을 더욱 편리하게 이용할 수 있다.

서비스 필요성

- 약자의 이동권 향상
- 편의시설의 안내가 부족하여 이용률이 낮음
- 다양한 이용자들의 시설이용 접근성을 높임

선견지명'S 프로젝트



전 페이지

→ 제작 배경 페이지

다음 페이지

→ 활용 공공데이터

활용 공공데이터



[서울교통공사 지하철역 주소 및
전화번호 정보](#)
[서울시 장애인 지하철 편의시설 정보](#)



[서울교통공사 교통약자
이용시설\(승강기\) 가동현황](#)



[서울시 전동보장구 급속충전소](#)



[서울시 장애인 도서관 현황 정보](#)



[서울시 응급실 위치 정보](#)



[서울시
사회복지시설\(장애인의료재활시설\) 목록](#)



[서울시
사회복지시설\(장애인기타\) 목록](#)



[서울시
사회복지시설\(장애인거주시설\) 목록](#)

선견지명'S 프로젝트



전 페이지

→ 활용 공공데이터 페이지

다음 페이지

→ 개발 기간 페이지

개발 기간

4월

일	월	화	수	목	금	토
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
		기획				
23	24	25	26	27	28	29
		프론트엔드 개발				
30	31	1	2	3	4	5



기획



프론트엔드 개발

5월

일	월	화	수	목	금	토
	1	2	3	4	5	6
7	8	9	10	11	12	13
		백엔드 개발				
14	15	16	17	18	19	20
21	22	23	24	25	26	27
		발표 준비				
28	29	30	31	1	2	3
		유지 보수				
4	5	6	7	8	9	10



백엔드 개발



발표 준비



유지 보수

선견지명'S 프로젝트



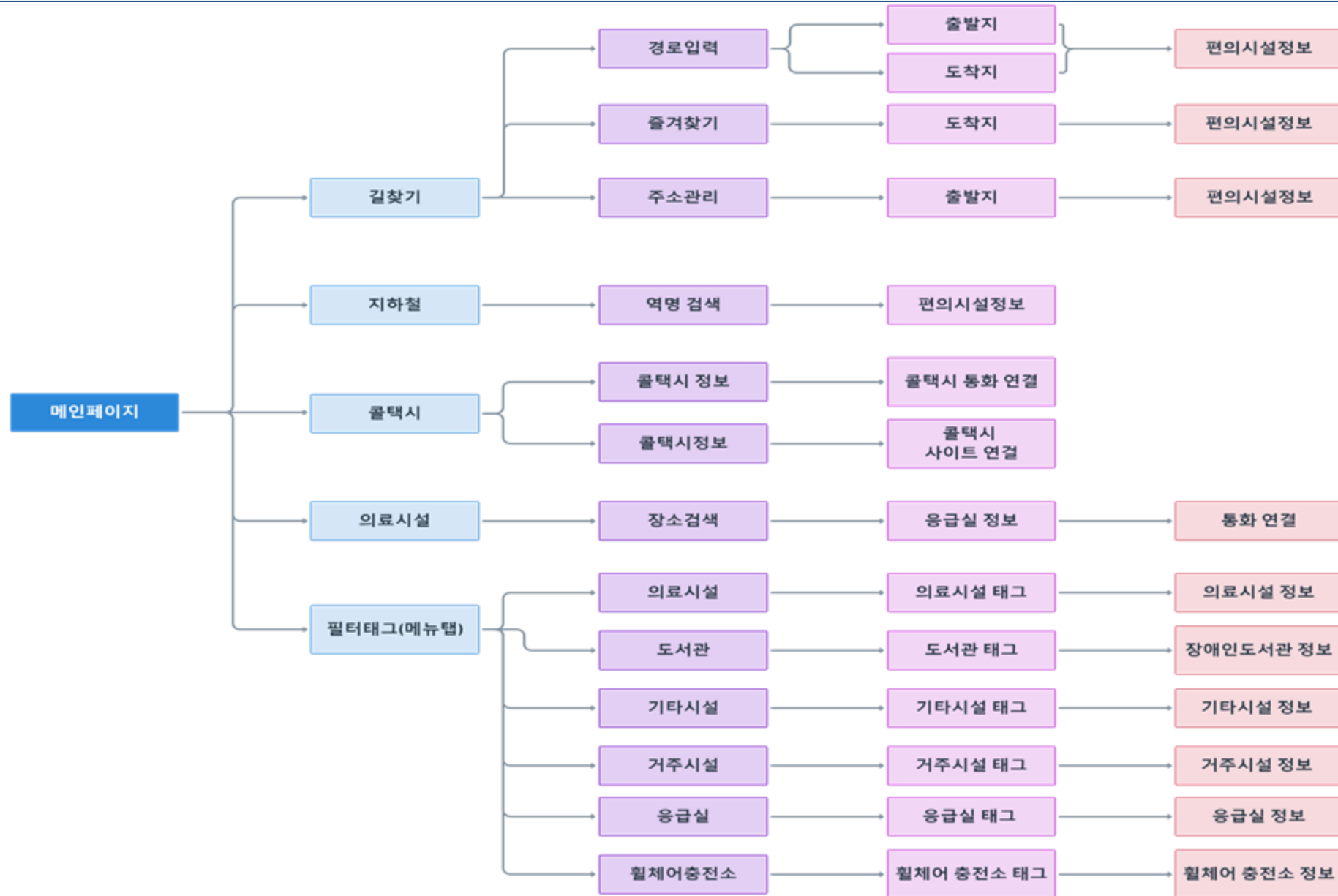
전 페이지

→ 개발 기간 페이지

다음 페이지

→IA 페이지

IA(Information Architect)



선견지명'S 프로젝트



전 페이지

→ Information Architect

다음 페이지

→ 서비스 활용방안 및 기대효과 페이지

1. 이동경로에 대한 편의시설 사전 파악

교통 약자의 이동 경로 근처에 위치한 편의 시설에 대해 미리 파악할 수 있어서 유사시에 대비하거나 사전에 경로를 파악하는데 용이하다.

2. 즐겨찾기 기능을 통한 편의시설 검색 시간 단축

자주 이용하는 편의 시설의 정보를 즐겨 찾기에 저장할 수 있기 때문에 검색할 시간이 단축된다.

3. 다양한 종류의 편의시설을 구분하여 검색

편의 시설을 종류별로 구별하여 볼 수 있기 때문에 특정 편의 시설들의 정보를 보다 정확하고 빠르게 알 수 있다

4. 보호자 및 일반 사용자의 접근성 확대

보호자들 또는 다른 일반인들도 쉽게 접근이 가능하기 때문에 편의 시설의 종류와 위치, 그리고 정보 등을 알 수 있다.

5. 실시간 서비스로 빠른 편의시설 정보 제공 및 이용

약자가 필요한 서비스와 편의 시설의 위치가 실시간으로 제공되기 때문에 보다 빠르게 편의 시설의 정보를 얻어 이용할 수 있다.

6. 장애인을 위한 접근성 향상을 통한 앱 이용자의 다양성 확보

현재로서는 없지만 추후에 시각 장애인을 위한 대체 텍스트나 청각 장애인들을 위한 폐쇄 자막 등을 넣어준다면 앱이용자들의 범용성이 확대될 수 있다.

선견지명'S 프로젝트



전 페이지

→ 서비스 활용방안 및 기대효과 페이지

다음 페이지

→ 개발 툴 및 참고문헌

Data
Base

Web



Server



Tool

선견지명'S 프로젝트



전 페이지

→ 개발 툴 및 참고문헌

다음 페이지

→ 핵심 코드 페이지

핵심코드

```
let show_etc_marker = []
for(let i = 0; i < markers_etc.length; i++){
  for (let j = 0; j < total_route_coordinate.length; j++){
    c1 = total_route_coordinate[j]
    c2 = markers_etc[i].getPosition()
    var poly_check_line = new kakao.maps.Polyline({
      // map: map, 을 하지 않아도 거리는 구할 수 있다.
      path: [c1, c2]
    });
    var dist = poly_check_line.getLength();

    if(dist <= 500){
      show_etc_marker.push(markers_etc[i].Gb)
      console.log(markers_etc[i].Gb)
    }
  }
}
const set_show_etc = new Set(show_etc_marker)

const uniqueEtc = [...set_show_etc];

for (let k = 0; k < markers_etc.length; k++){
  if(!uniqueEtc.includes(markers_etc[k].Gb)){
    markers_etc[k].setMap(null)
  }
}
```

길찾기(기타시설)

- **total_route_coordinate**는 polyline(경로)가 만들어지는 좌표들이 들어 있는 Array이다.
- **total_route_coordinate**의 각 좌표들을 c1으로 한다.
- 기타시설 마커들의 좌표를 c2라 했을때
- c1, c2로 polyline을 만들어서 지도에 표시는 하지 않고 **getLength()**를 이용해 polyline의 직선 거리를 구할 수 있다.
- 이때 500미터 이하인 마커들의 이름은 **show_etc_marker**에 넣는다.
 - 1) 다만 **show_etc_marker**에 들어가는 이름들은 중복이 있을 테니 set을 **set_show_etc**에서 새로 정의를 해준다.
 - 2) Set을 이용하면 object가 되니 **uniqueEtc**라는 새로운 배열 형태로 만들어준다.
- 마지막 for문의 경우 **uniqueEtc**안에 기타시설 마커의 이름이 들어가 있지 않으면(반경 범위 밖에 있다면) 지도에서 지워준다.

핵심코드

```
if (markers_medical != undefined) {  
  for (let j = 0; j < markers_medical.length; j++) {  
    let La = markers_medical[j].La  
    let Ma = markers_medical[j].Ma  
    c1 = coordiante[i];  
    c2 = new kakao.maps.LatLng(Ma, La)  
    var poly_check_line = new kakao.maps.Polyline({  
      // map: map, 을 하지 않아도 거리는 구할 수 있다.  
      path: [c1, c2]  
    });  
    // console.log(poly_check_line)  
    var dist = poly_check_line.getLength();  
    distance_medical.push(dist)  
  }  
}
```

```
console.log(distance_medical);  
const ambulance_array = distance_ambulance.map((value, index) => ({ value, index }));  
const medical_array = distance_medical.map((value, index) => ({ value, index }));  
  
ambulance_array.sort((a, b) => a.value - b.value);  
medical_array.sort((a, b) => a.value - b.value);  
  
const ambulance_min_1 = ambulance_array[0].value  
const ambulance_min_2 = ambulance_array[1].value  
const medical_min_1 = medical_array[0].value  
const medical_min_2 = medical_array[1].value  
  
const new_array = [ambulance_min_1, ambulance_min_2, medical_min_1, medical_min_2]  
new_array.sort((a, b) => a - b);  
  
// 데이터를 수신하여 처리하는 함수  
const data = document.querySelectorAll('.hospital_info_container')  
const indices = [];  
for (let i = 0; i < data.length; i++) {  
  const value = new_array[i];  
  let index;
```

의료시설 (가장 가까운 2개 고르기, 가장 짧은 2개 고르기)

사진1 → 의료시설

- c1에는 검색하고자 하는 장소의 좌표가 들어간다
- c2에는 길찾기와 마찬가지로 각 의료시설 마커들의 좌표를 넣는다.
- 이 두개를 이용해 polyline을 형성하여 **getLength()**를 이용해 나온 dist를 **distance_medical**에 넣는다.
- 응급실(ambulance)도 이와 같은 원리이다.

사진2 → 가장 짧은 2개 고르기

- **ambulance_array**와 **medical_array**는 map을 이용하여 각 해당 요소와 그 요소의 index를 객체로 가지는 새로운 object를 생성한다.
- 그 다음에 value(거리)값을 가지고 정렬을 진행한다.
- 가장 짧은 2개를 골라야 하므로 각각에서 가장 짧은 2개를 골라 **ambulance_min_1, ambulance_min_2, medical_min_1, medical_min_2**라고 한다.
- **ambulance_min_1, ambulance_min_2, medical_min_1, medical_min_2**를 요소로 가지는 new_array를 만들어서 정렬을 하면
=> 거기에서 맨 앞에 2개가 가장 가까운 거리에 있는 응급실 혹은 의료시설이 된다.

핵심 코드



```
function ClickSubmit() {  
  
    const stationName = document.getElementById('input_station_name').value;  
    // 입력값이 없는 경우 처리할 내용  
    if (stationName == '' || !stationName) {  
        alert('역 이름을 입력하세요!')  
        return false  
    } else {  
        const encodedStationName = encodeURIComponent(stationName);  
        const url = "/Client/web/subway_detail_page.html?station_name=" + encodedStationName;  
        window.location.href = url;  
    }  
}
```

지하철 노선도

- input태그 안에 역명을 입력하고 검색 버튼에 **onclick=ClickSubmit()** 함수를 실행하도록 옵션을 넣어준다.
- 입력된 역명을 url의 value값으로 가져와 url을 저장하되 암호화 하여 저장(url에서 직접 변경하지 못하도록)
- 현재 창의 URL을 변경하는 방식으로 페이지를 이동한다.

핵심 코드



```
const queryString = window.location.search; // URL의 쿼리 매개변수 추출
const urlParams = new URLSearchParams(queryString); // URLSearchParams 객체 생성
let station_name = urlParams.get("station_name"); // 변수 가져오기
```

지하철 노선도

- `const queryString = window.location.search`: 현재 URL에서 쿼리 문자열을 추출 `window.location.search`는 현재 URL의 쿼리 문자열 부분을 나타냄
- `const urlParams = new URLSearchParams(queryString)`: 쿼리 문자열로 `URLSearchParams` 객체를 생성
- `let station_name = urlParams.get("station_name")`: `station_name`이라는 쿼리 매개변수 값을 저장

핵심 코드

2 역삼역 2호선

역무실 전화번호

02-6110-2211

전통차 전용공간 승·하차 위치

정보없음

연단간격이 좁은 승·하차 위치
(역자와 승강장 사이가 좁은 승·하차 위치)

내선

5-4번

외선

5-4번

엘리베이터 정보

내

외선 7-3

내

내선 4-1

외

3번 출구측

외

4번 출구측

휠체어 리프트 정보

정보 없음


```
// 사용자가 입력한 역 이름의 길이
let len = station_name.length;


// 마지막 '역'을 가져오기 위해 len - 1
lastchar = station_name.charAt(len - 1)
if (lastchar == '역') {
    console.log(lastchar)
    num = len - 1; // 마지막 '역'을 잘라주기 위해
    station_name = station_name.slice(0, num)
} else {
    num = len;
}
```


지하철 노선도


→ 사용자가 'OO역'이라고 검색할 경우 마지막 '역'을 제외하여 OpenAPI의 station name과 일치하는 지 확인하기 위한 코드


핵심 코드



 역삼역 2호선


 역무실 전화번호
02-6110-2211

 장애인 전용공간 승·하차 위치
정보없음

 연담간격이 좁은 승·하차 위치
(열차와 승강장 사이가 좁은 승·하차 위치)

내선 5-4번

외선 5-4번


 엘리베이터 정보

내 외선 7-3

내 내선 4-1

외 3번 출구측

외 4번 출구측

 휠체어 리프트 정보
정보없음

 : Open API 파싱

 : DataBase fetch

```
var xhr = new XMLHttpRequest();
var url = 'http://openapi.seoul.go.kr:8088/67724c44507268643131395759454f4c/json/StationAdresTelno/1/288/';

xhr.open('GET', url);
xhr.onreadystatechange = function () {
  if (this.readyState == xhr.DONE) { // <== 정상적으로 준비되었을때
    const jsonString = this.responseText;
    const parsedJson = JSON.parse(jsonString);
    // JSON 객체화
    const jsonString2 = parsedJson.StationAdresTelno.row;
    const result = [];
    let line_num;
    let check_bool = false;
    for (let i = 0; i < jsonString2.length; i++) {
      const STATION_NAME = jsonString2[i].STATN_NM.slice(0, num)
      if (station_name == STATION_NAME) {
        check_bool = true;
        result.push({
          line_num: jsonString2[i].LINE[0],
          line_name: jsonString2[i].LINE,
          station_name: jsonString2[i].STATN_NM,
          station_telNum: jsonString2[i].TELNO
        })
      }
    }

    //check_bool != false -> true
    if (!check_bool) {
      window.location.href = './no_subway_detail.html'
    }
  }
}
```

필요한 정보만 파싱하기 위해 새로운 객체를 만듦

Open API에서 제공하는 정보가 없을 경우
not found 페이지로 이동

선견지명'S 프로젝트



전 페이지

→ 핵심코드 페이지

다음 페이지

→ 화면 및 기능 설명 페이지

화면 및 기능 설명



메인 페이지

1. 메뉴 탭 버튼을 누를 경우 하단 메뉴바와 마이 페이지, 로그인 창이 나옵니다.
2. 찾고자 하는 장소를 입력하면 됩니다.
3. 찾고자 하는 장소를 입력 후 검색버튼을 누르면 검색 결과가 지도에 나옵니다.
4. 필터 메뉴 탭을 터치하면 클릭한 필터의 편의시설 마커만 표시가 됩니다.
5. 원하는 마커를 터치할 경우 그 위치의 자세한 정보가 나옵니다.
6. 터치하면 길 찾기 기능이 활성화 됩니다.
7. 터치하면 콜택시 페이지로 이동합니다.
8. 터치하면 지하철 페이지로 이동합니다.
9. 터치하면 의료시설 페이지로 이동합니다.
10. 로그인 페이지로 이동합니다.
11. 즐겨찾기 페이지로 이동합니다.
12. 주소관리 페이지로 이동합니다.

화면 및 기능 설명



메인페이지

1. 필터 메뉴 탭을 누르면 클릭한 필터의 편의시설 마커만 표시됩니다.
2. 마커를 터치하면 마커가 찍혀 있는 장소에 대한 간단한 편의시설 정보가 나옵니다.
3. 시설명을 터치할 경우 편의시설의 더 자세한 정보를 알 수 있습니다.
- 3-1. 해당 시설의 더 자세한 정보를 확인할 수 있습니다.
4. 터치하면 즐겨찾기 목록으로 등록할 수 있습니다.
5. 터치하면 해당 장소를 출발지로 등록할 수 있습니다.
6. 터치하면 해당 장소를 도착지로 등록할 수 있습니다.

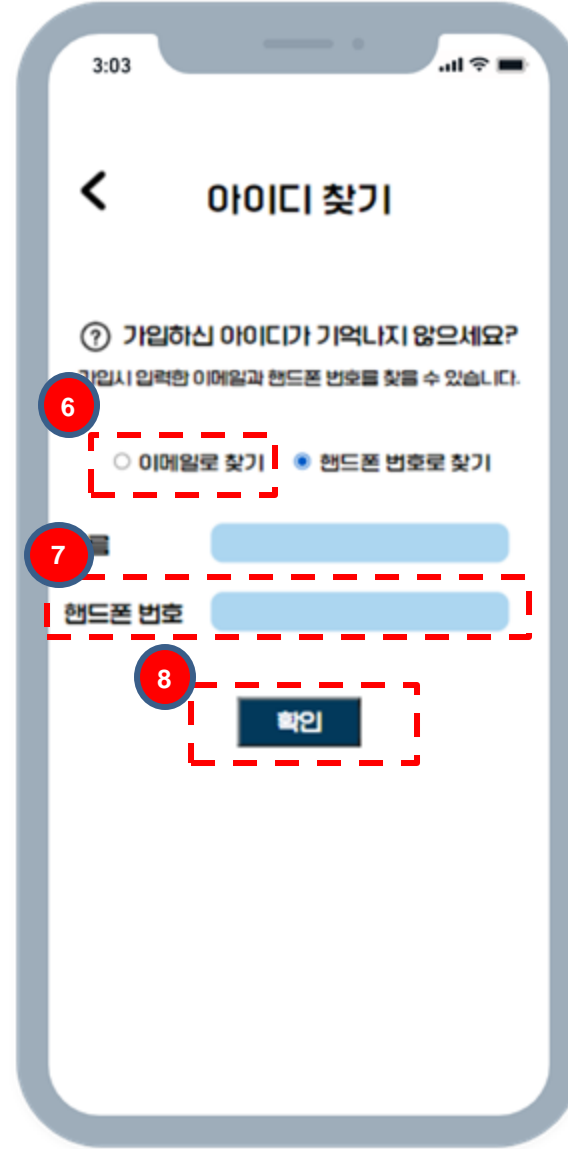
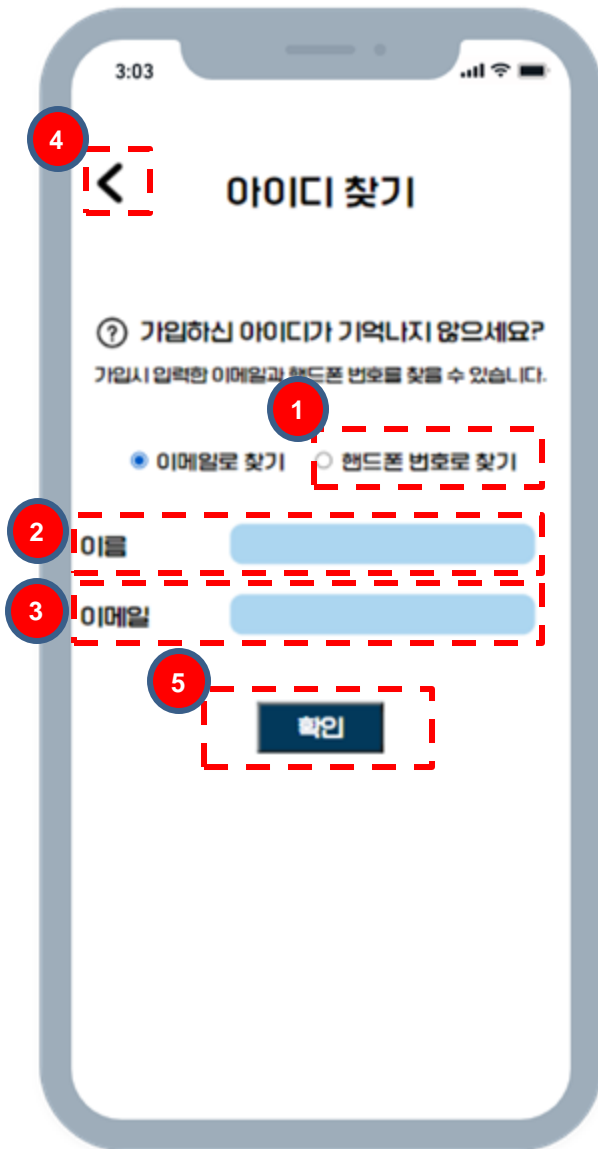
화면 및 기능 설명



회원가입, 개인정보동의 페이지, 로그인 페이지

1. 사용하고자 하는 아이디를 입력합니다(영어, 숫자 포함 입력)
2. 사용할 비밀번호를 입력합니다(숫자)
3. 회원가입 페이지로 이동합니다
4. 아이디 찾기 페이지로 이동합니다.
5. 비밀번호 페이지로 이동합니다.
6. 터치하면 모든 이용 약관 동의 체크박스에 체크가 되면서 다음 버튼이 활성화가 됩니다.
7. 터치하면 이용약관에 체크가 됩니다.
8. 활성화가 된 상태에서 터치하면 회원가입 페이지로 이동합니다.
9. 터치하면 로그인 페이지로 이동합니다.
10. 회원가입 정보를 입력합니다.
11. 터치하면 지금까지 입력한 내용들이 전부 초기화됩니다
12. 조건에 맞게 회원가입 정보들을 입력하고 터치하면 회원가입이 완료됩니다.

화면 및 기능 설명



아이디찾기 페이지

1. 핸드폰 번호로 아이디를 찾고 싶으면 터치하면 됩니다.
2. 회원가입에서 등록한 이름을 입력합니다
3. 회원가입에서 등록한 이메일을 입력합니다
4. 터치하면 로그인 페이지로 이동합니다.
5. 이름과 이메일이 일치하면 아이디를 알려줍니다.
6. 이메일로 아이디를 찾고 싶으면 터치합니다.
7. 회원가입에서 등록한 핸드폰 번호를 입력합니다
8. 이름과 핸드폰 번호가 일치하면 아이디를 알려줍니다.

화면 및 기능 설명

비밀번호 찾기 페이지, 비밀번호 변경 페이지

1. 회원가입에서 등록한 아이디를 입력합니다
2. 회원가입에서 등록한 이름을 입력합니다
3. 회원가입에서 등록한 핸드폰 번호를 입력합니다.
4. 아이디, 이름, 핸드폰 번호가 일치하면 비밀번호 변경 페이지로 이동합니다.
5. 터치하면 로그인 페이지로 이동합니다
6. 터치하면 로그인 페이지로 이동합니다.
7. 새롭게 사용할 비밀번호를 입력합니다.
8. 새롭게 사용할 비밀번호와 동일하게 입력합니다.
9. 터치하면 비밀번호 변경이 완료가 됩니다.

화면 및 기능 설명

3:03

마이페이지

고예성님, 안녕하세요!

1. 내 정보 수정

2. 아이디

yesung

이름

고예성

주소

fsfsd

전화번호

010-2916-4552

이메일

yes@yse.com

다시 입력

비밀번호

수정 완료

3. 4. 5.

3:03

비밀번호 수정

고예성님, 안녕하세요!

6. 기존 비밀번호

새로운 비밀번호

비밀번호 확인

다시 입력

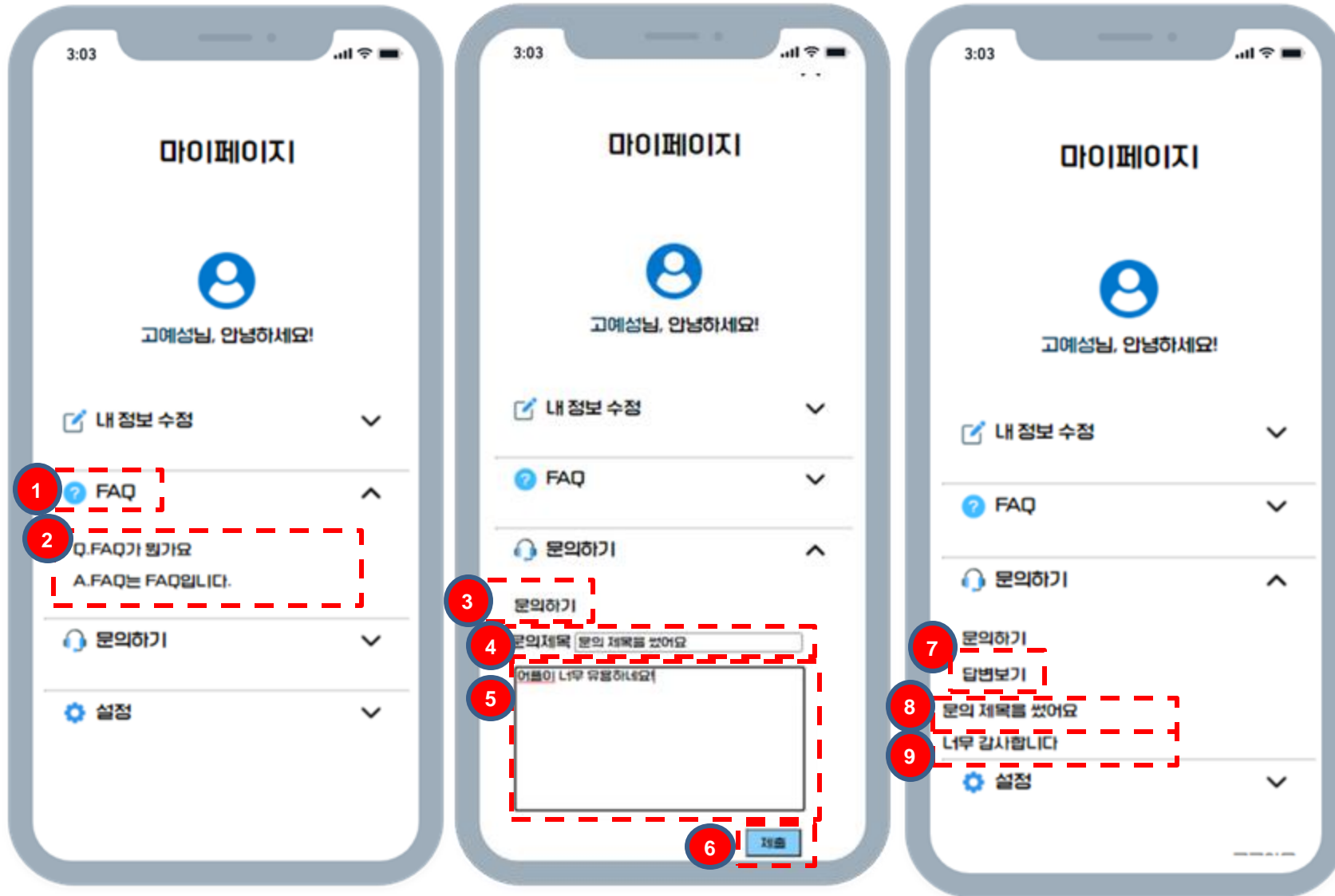
수정 완료

3. 7.

마이 페이지

1. 터치하면 내 정보를 수정할 수 있는 페이지가 나옵니다.
2. 사용 유저의 정보를 수정할 수 있습니다.
3. 터치하면 입력한 내용들이 초기화되면서 다시 입력할 수 있습니다.
4. 터치하면 비밀번호 수정 페이지로 이동합니다.
5. 터치하면 정보가 수정됩니다.
6. 기존 비밀번호와 새로운 비밀번호를 입력합니다.
7. 터치하면 비밀번호 수정이 완료됩니다.

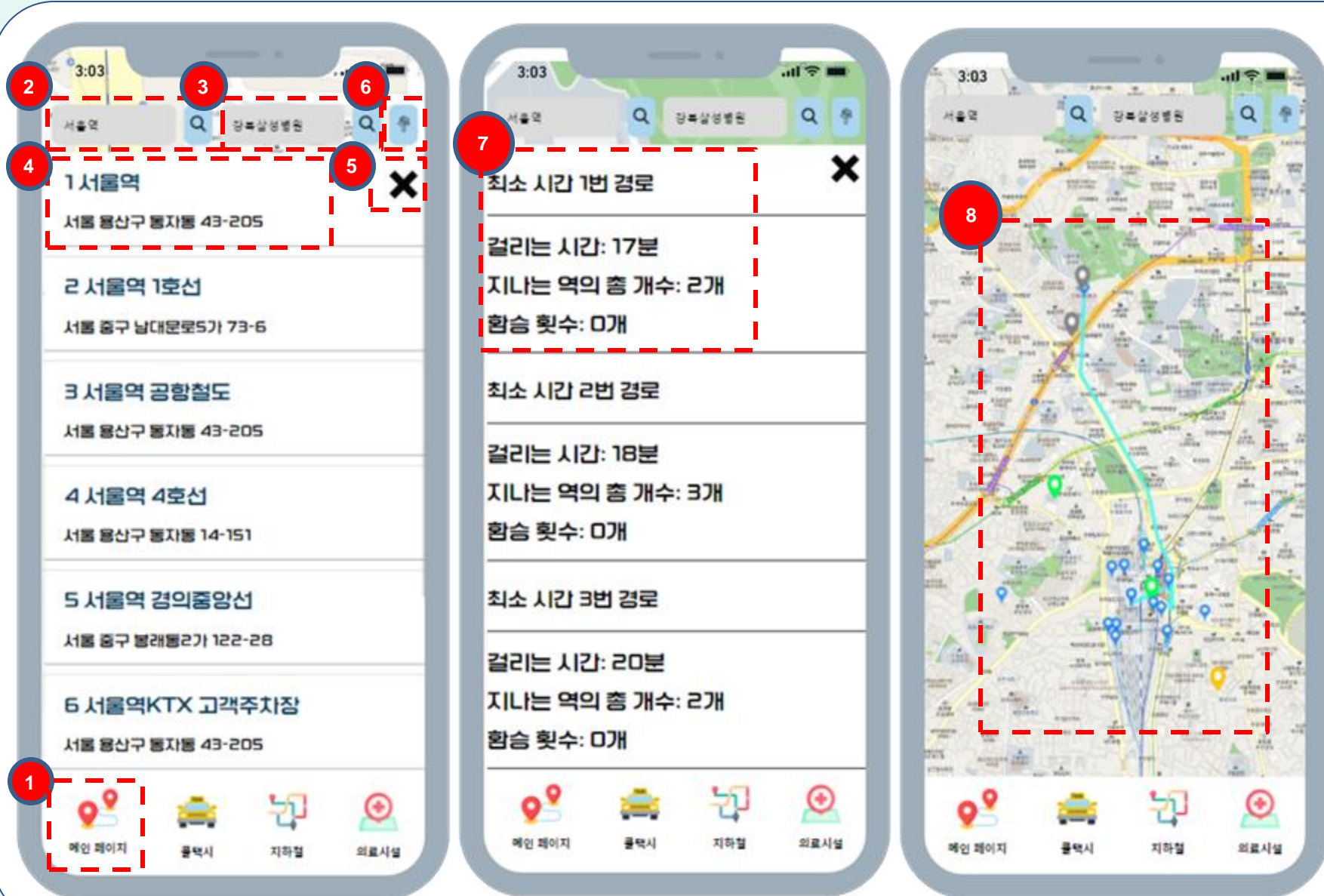
화면 및 기능 설명



마이페이지

1. 터치하면 FAQ를 확인할 수 있습니다.
2. 등록되어 있는 FAQ가 나옵니다.
3. 터치하면 문의 제목과 문의 내용을 등록하는 페이지가 나옵니다.
4. 문의 제목을 입력할 수 있습니다.
5. 문의 내용을 입력할 수 있습니다.
6. 문의 제목과 문의 내용이 기재되어 있으면 제출이 됩니다.
7. 터치하면 문의에 대한 답변을 확인할 수 있습니다.
8. 문의 제목이 나옵니다.
9. 문의 답변 내용이 나옵니다.

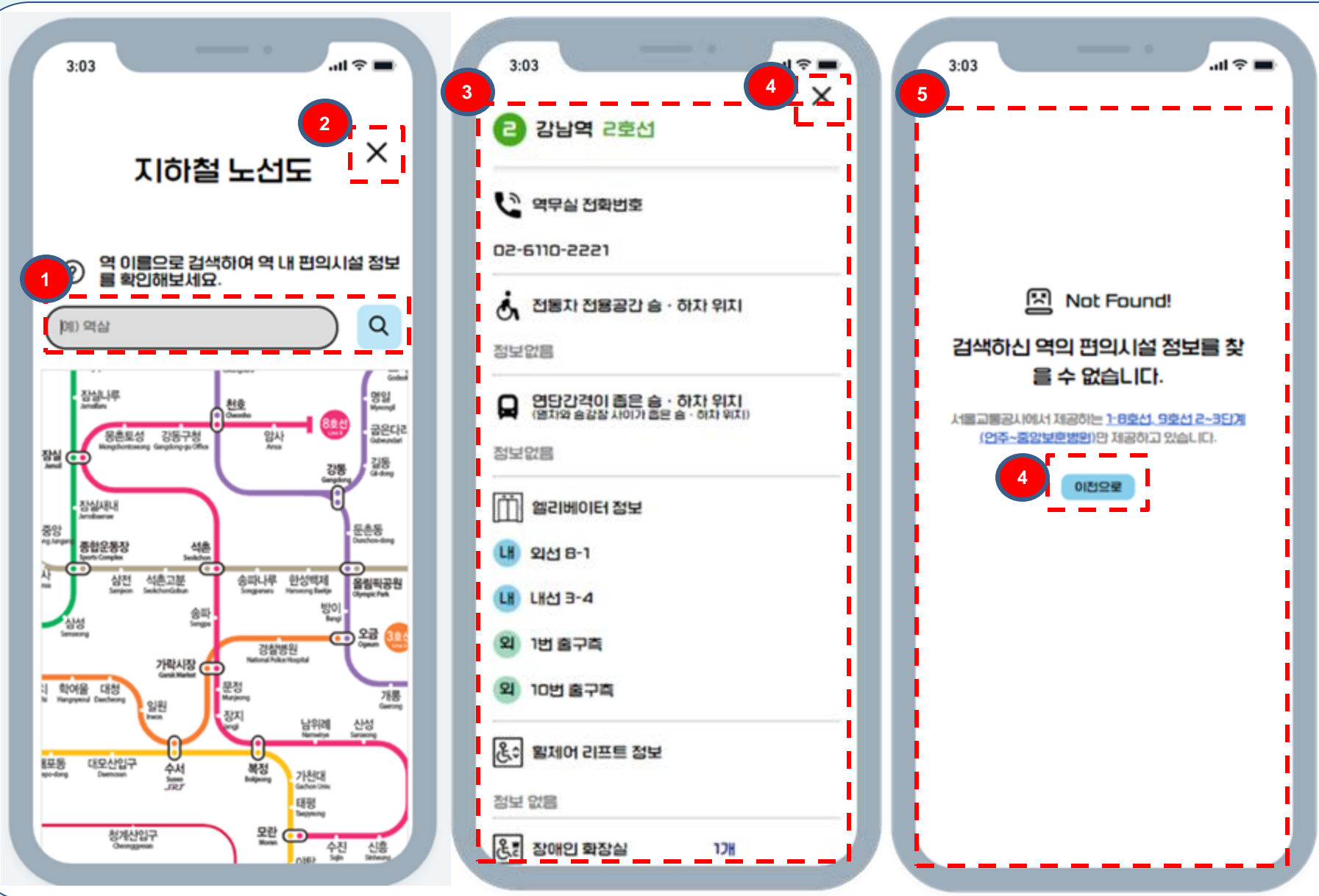
화면 및 기능 설명



길찾기 페이지

1. 메인 페이지로 이동합니다.
2. 출발지를 입력하고 검색 버튼을 터치하면 정보가 나타납니다.
3. 도착지를 입력하고 검색 버튼을 터치하면 정보가 나타납니다.
4. 원하는 정보를 터치 할 수 있습니다.
5. 정보가 잘못되었을 경우 정보 페이지를 닫을 수 있습니다.
6. 출발지와 도착지가 입력되고 경로찾기 버튼을 누르면 경로 정보가 나타납니다.
7. 경로를 선택하면 경로와 반경 500m의 편의시설이 나타납니다
8. 경로와 편의시설의 마커가 나타납니다

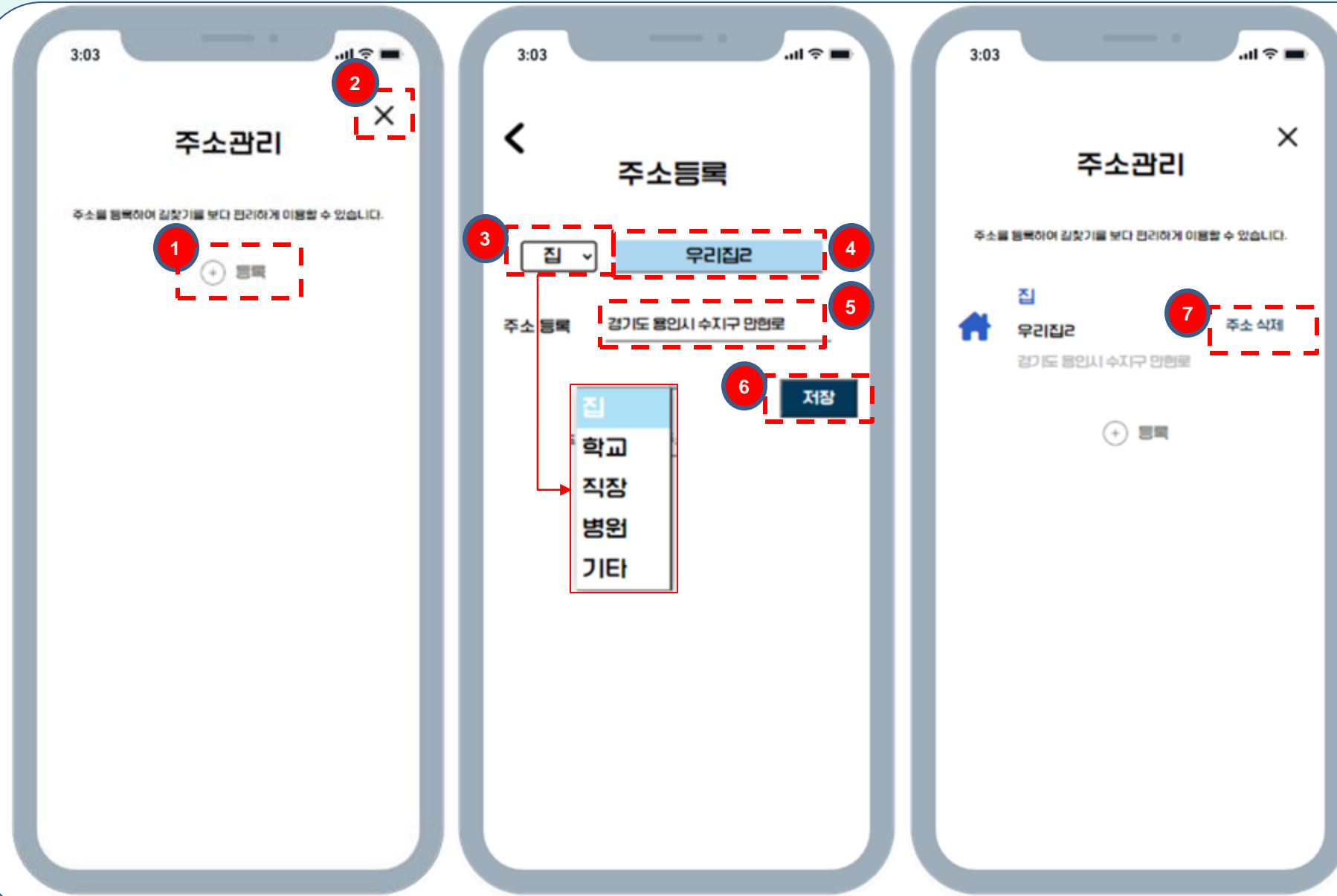
화면 및 기능 설명



지하철 역 검색 페이지

- 1. 지하철 역의 이름을 입력할 수 있습니다.
- 2. 터치하면 메인페이지로 이동합니다.
- 3. 검색한 역에 대한 편의시설 정보를 확인할 수 있습니다.
- 4. 터치하면 지하철 역 검색 페이지로 이동합니다.
- 5. 검색하고자 하는 역명이 존재하지 않거나, 서울교통공사에서 제공하지 않는 역일 경우 해당 페이지로 이동합니다.

화면 및 기능 설명



주소관리 페이지

1. 터치하면 주소 등록 페이지로 이동합니다.
2. 터치하면 메인 페이지로 이동합니다.
3. 터치하면 드롭다운으로 원하는 주소의 종류를 고를 수 있습니다.
4. 주소의 별칭을 입력할 수 있습니다.
5. 주소를 입력할 수 있습니다.
6. 터치하면 주소 정보가 저장됩니다.
7. 터치하면 주소 정보를 삭제할 수 있습니다.

화면 및 기능 설명



즐거찾기 페이지, 콜택시 페이지

1. 마커를 터치하여 정보창의 별표를 누르면 즐거 찾기 목록에 등록됩니다.
2. 터치하면 등록된 즐거 찾기 목록이 전체 선택됩니다.
3. 해당 주소를 터치하면 길 찾기에 도착지로 설정되고 길 찾기 페이지로 이동합니다.
4. 터치하면 선택삭제와 전체삭제 버튼이 등장하고 선택된 즐거찾기 목록을 삭제하거나 전체 목록을 삭제할 수 있습니다.
5. 터치하면 메인페이지로 이동합니다.
6. 터치하면 해당 번호로 전화를 걸 수 있습니다
- 7~11 하이퍼링크 연결
https://www.sisul.or.kr/open_content/calltaxi/introduce/receipt.jsp 7번 페이지로 이동
https://www.sisul.or.kr/open_content/calltaxi/introduce/guide.jsp 8번 페이지로 이동
https://www.sisul.or.kr/open_content/calltaxi/guidance/internet.jsp 9번 페이지로 이동
https://www.sisul.or.kr/open_content/calltaxi/guidance/view.jsp 10번페이지로 이동
https://www.sisul.or.kr/open_content/calltaxi/community/notice.jsp 11번페이지로 이동
12. 터치하면 메인 페이지로 이동합니다.

화면 및 기능 설명



관리자 페이지

1. 관리자 페이지 아이콘과 홈 버튼으로 누르면 메인 페이지로 이동하게 한다.
2. 홈 태그로 누르면 Dashboard가 나오게 되는데 이는 한눈에 다양한 정보를 보여준다. 추후 다양한 Dashboard를 추가함에 따라 다양한 정보를 얻을 수 있다.
3. 유저 태그로 누르면 유저들의 정보에 접근할 수 있다.
4. 고객센터 태그로 누르면 고객센터에 문의를 넣을 수 있다.
5. 관리자로 로그인 할 경우 보이는 페이지로 계정을 지우거나 혹은 계정 정보 등을 갱신할 수 있다.

선견지명'S 프로젝트

마치겠습니다.
감사합니다 😊