01. 목차

1. 목차
2. 목표 output
3. Study

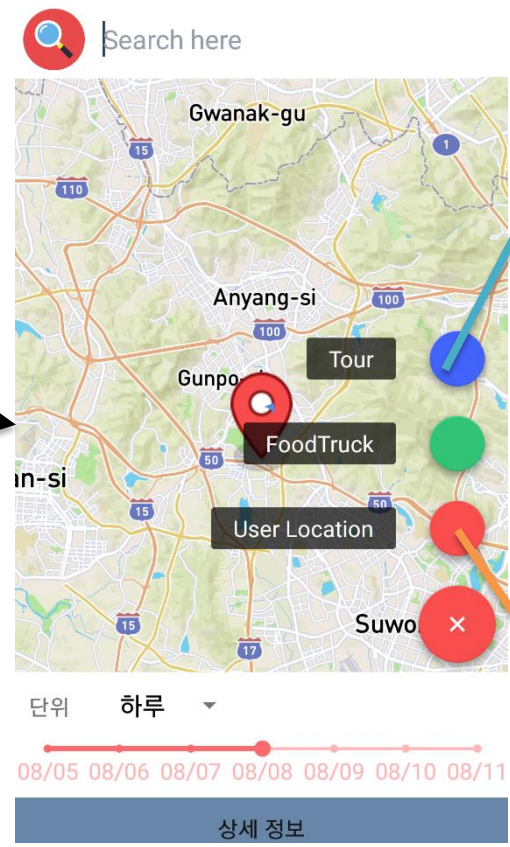
   3-(1) Mapbox click & move event, 상세 검색 , 현재 위치 - 이지은
   3-(2) progress bar, vpn - 이 솔
   3-(3) bottomsheet, chart이승은

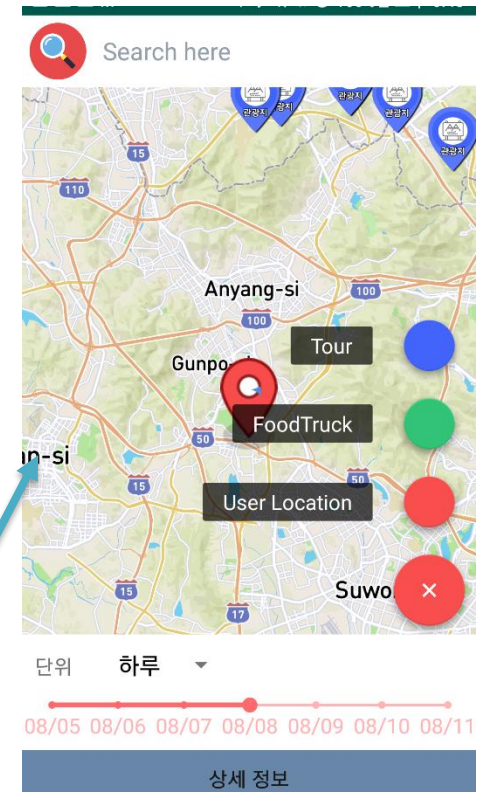
4. question

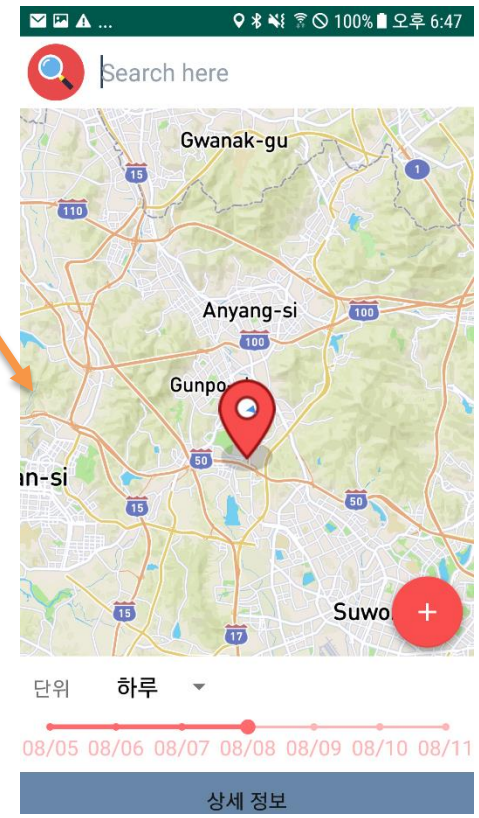
02. 목표 OUTPUT

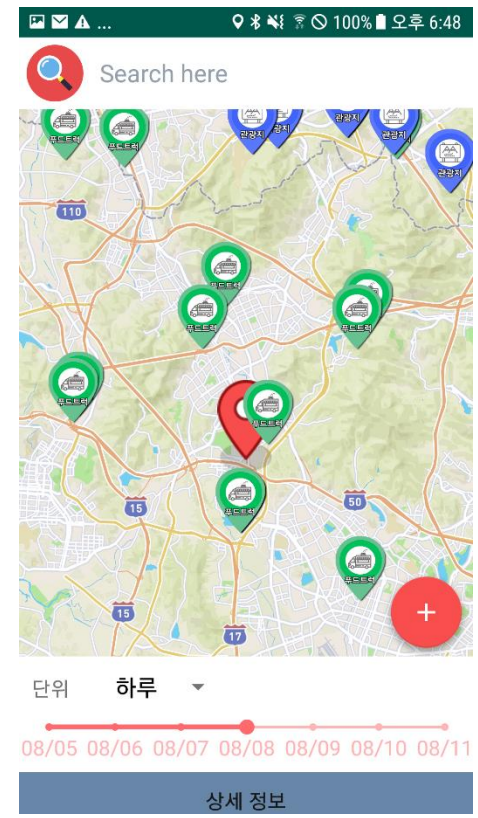
유동인구 분석 앱
Made by 지은 승은 솔

<스플래시 화면 구현>



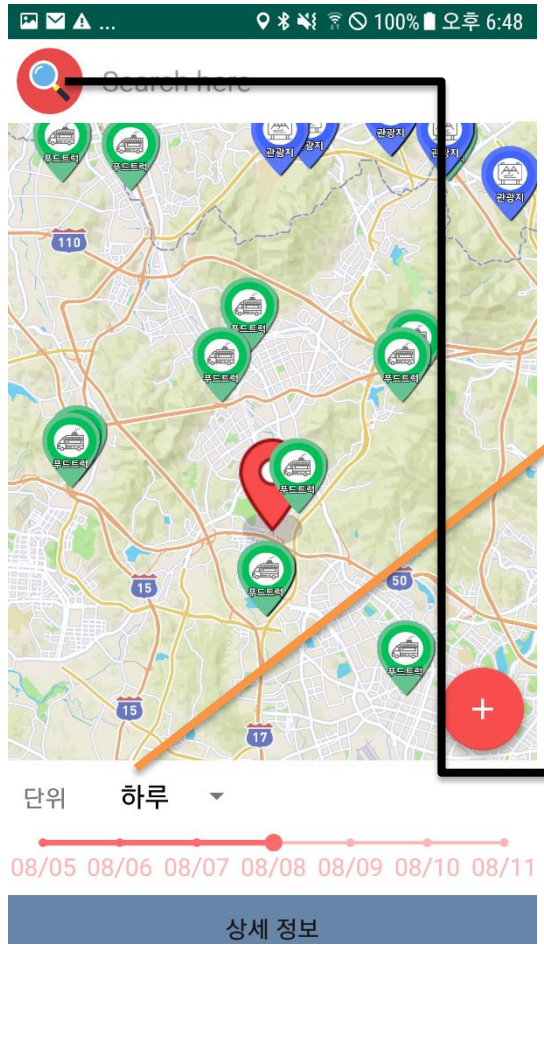


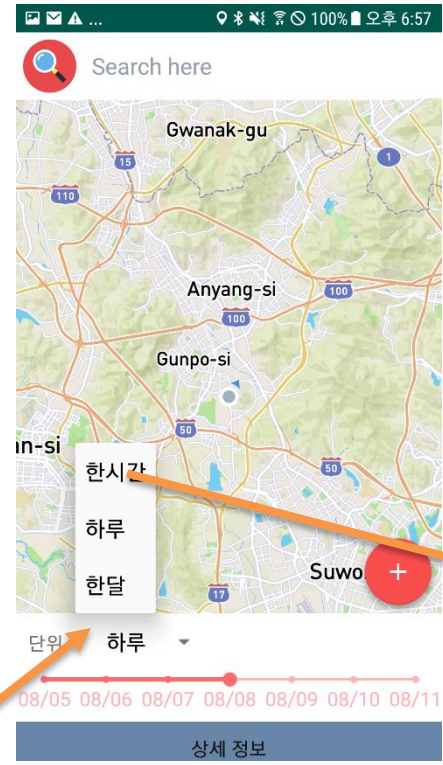


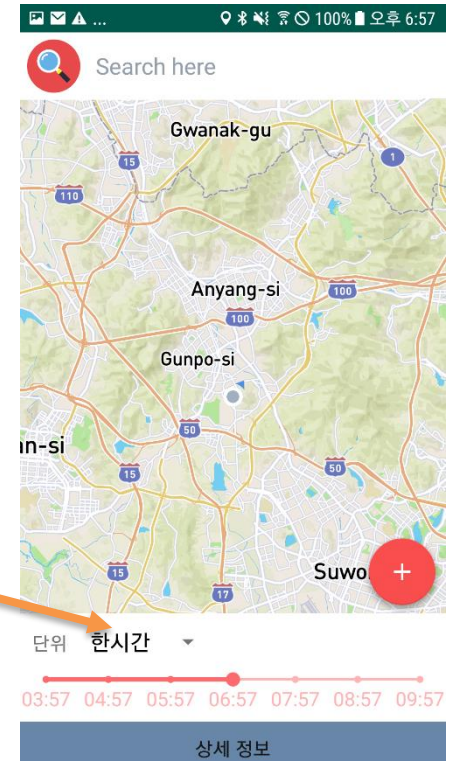


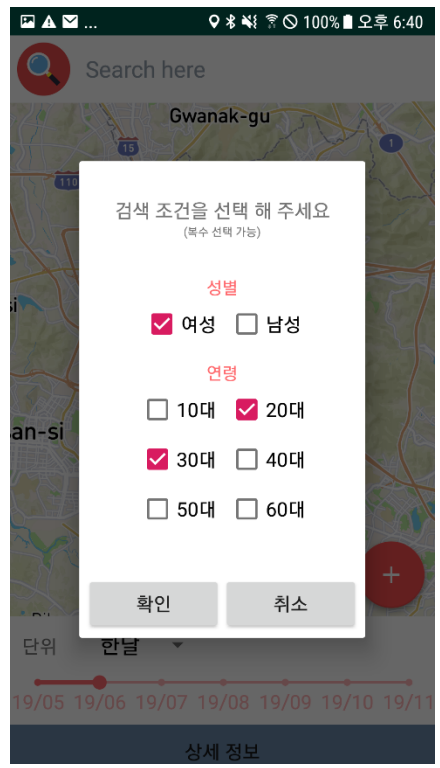
SK텔레콤 AI Fellowship

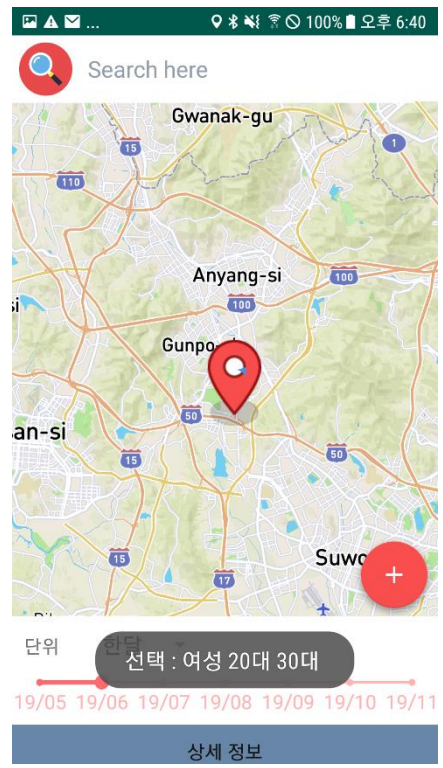
02. 목표 OUTPUT


Search here
단위
하루
08/05 08/06 08/07 08/08 08/09 08/10 08/11
상세 정보

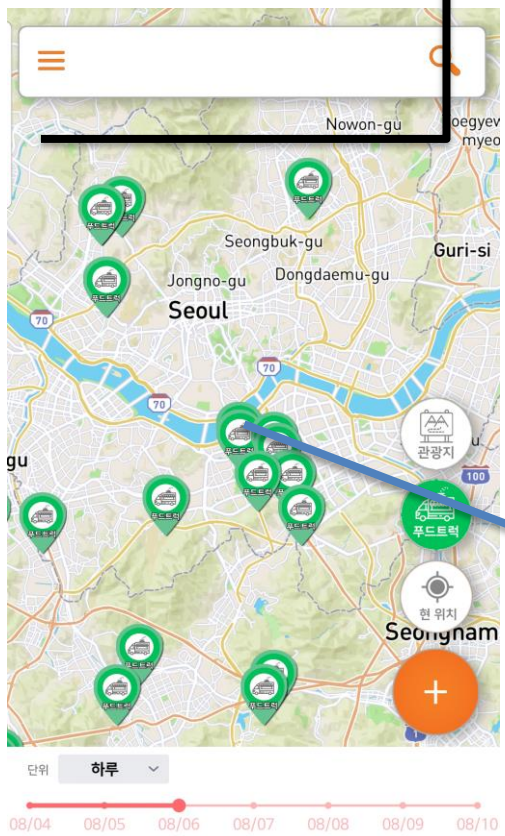

Search here
Gwanak-gu
Anyang-si
Gunpo-si
한시간
하루
한달
단위
하루
08/05 08/06 08/07 08/08 08/09 08/10 08/11
상세 정보

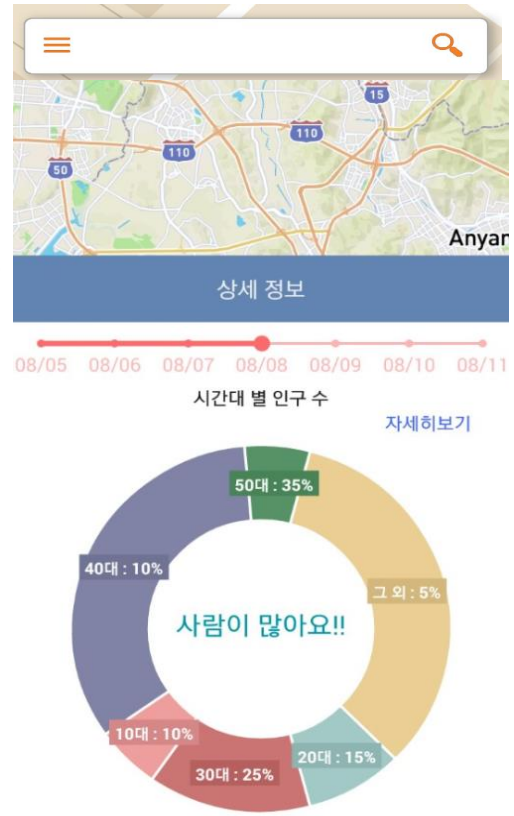

Search here
Gwanak-gu
Anyang-si
Gunpo-si
단위
한시간
03:57 04:57 05:57 06:57 07:57 08:57 09:57
상세 정보

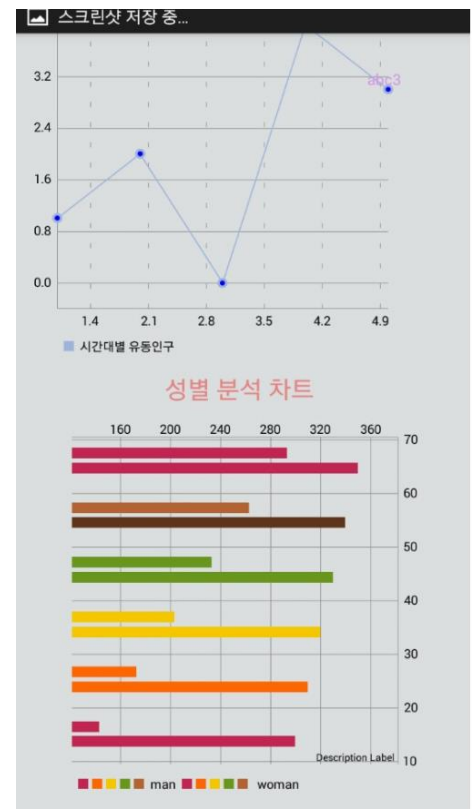

Search here
Gwanak-gu
검색 조건을 선택 해 주세요
(복수 선택 가능)
성별
여성
남성
연령
10대
20대
30대
40대
50대
60대
확인
취소
단위
한달
19/05 19/06 19/07 19/08 19/09 19/10 19/11
상세 정보


Search here
Gwanak-gu
Anyang-si
단위
선택 : 여성 20대 30대
19/05 19/06 19/07 19/08 19/09 19/10 19/11
상세 정보

02. 목표 OUTPUT

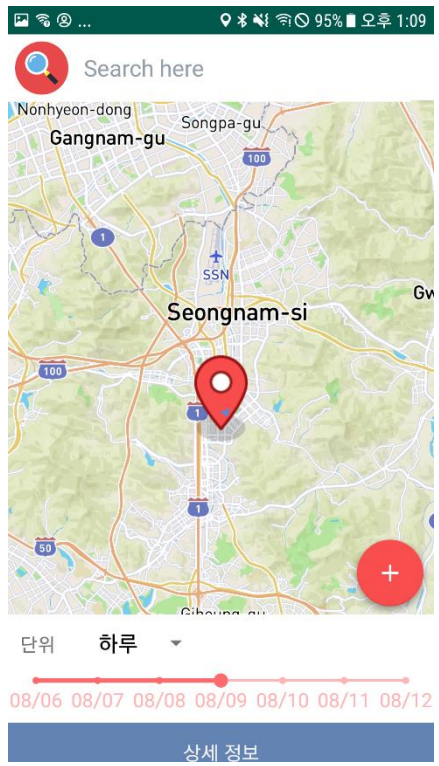




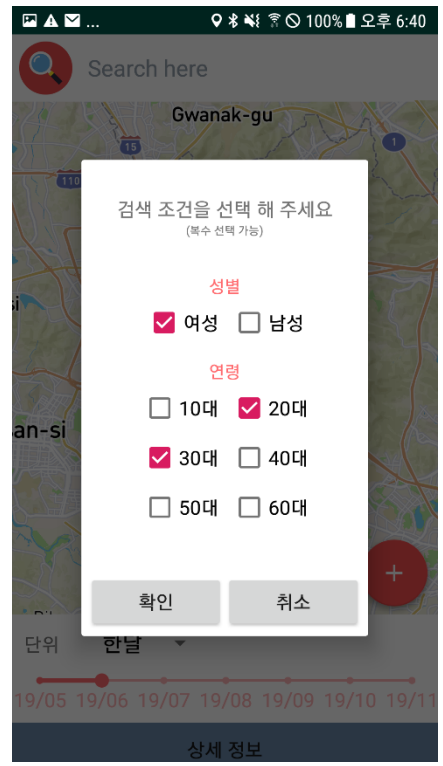


간단한 차트와 함께
화면에 대한
실시간 유동인구 정도를
사람이 많아요!
사람이 별로 없어요!
등으로 예쁘게 표현해줄 예정!

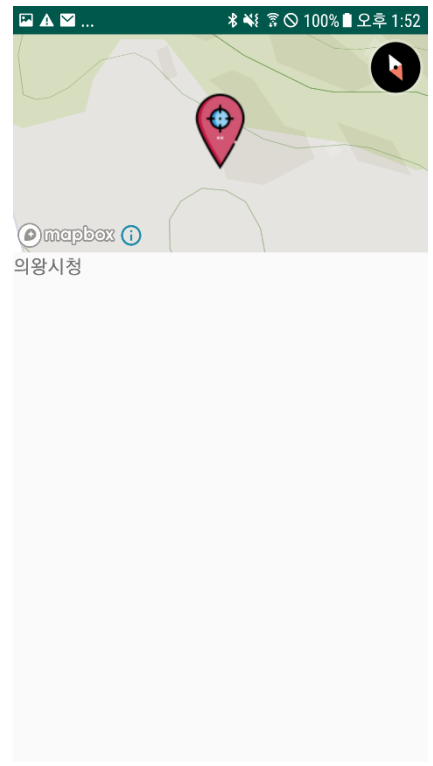
03. Study

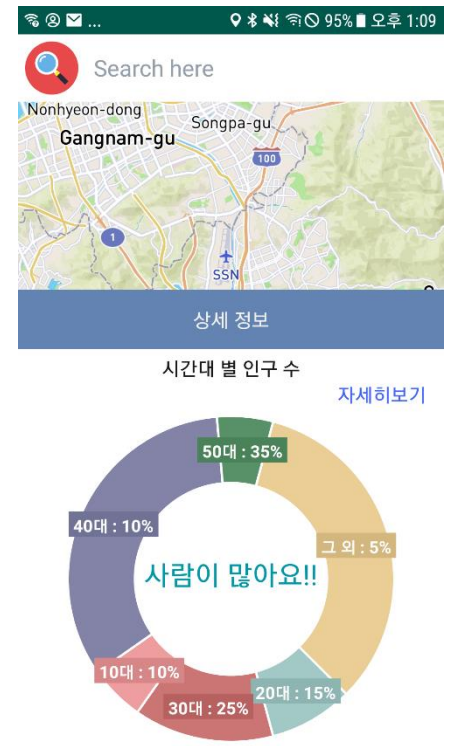


1) Progress Bar
2) Search Bar
3) VPN
- 이 솔



1) Mapbox 관련 함수
2) 상세 검색 (custom Dialog)
3) 상세 페이지
(DetailInfoActivity)
4) 현재 위치
- 이지은





1) chart API
- 이승은

03. STUDY

3-(

```
//jieun - mapbox on fling & on move events
mapboxMap.addOnMoveListener(new MapboxMap.OnMoveListener() {
    @Override
    public void onMoveBegin(MoveGestureDetector detector) {
        // user started moving the map
    }
    @Override
    public void onMove(MoveGestureDetector detector) {
        // user is moving the map
    }
    @Override
    public void onMoveEnd(MoveGestureDetector detector) {
        // user stopped moving the map
        int viewportWidth = mapView.getWidth();
        int viewportHeight = mapView.getHeight();
        Toast.makeText(HomeActivity.this, "onMoveEnd",
Toast.LENGTH_SHORT).show();
    }
});
mapboxMap.addOnFlingListener(new MapboxMap.OnFlingListener() {
    @Override
    public void onFling() {
        Toast.makeText(HomeActivity.this, "onFling",
Toast.LENGTH_SHORT).show();
    }
});
mapboxMap.addOnMapClickListener(new MapboxMap.OnMapClickListener() {
    @Override
    public boolean onMapClick(@NonNull LatLng point) {
    }
});
mapboxMap.addOnMapLongClickListener(new MapboxMap.OnMapLongClickListener() {
    @Override
    public boolean onMapLongClick(@NonNull LatLng point) {
});
```

Map box click event

(1) onMoveLister

- onMoveBegin
- onMove
- onMoveEnd

-> 움직임이 끝날때 마다 호출
(여기서 나중에 서버와 통신?) (toast message)

(2) onFlingListener

-> zoom 바뀔 때 마다 호출
(toast message)

(3) onMapClickLister

- **>지도 클릭 시 호출**

(4) onMapLongClickLister

-> 길게 클릭시 호출

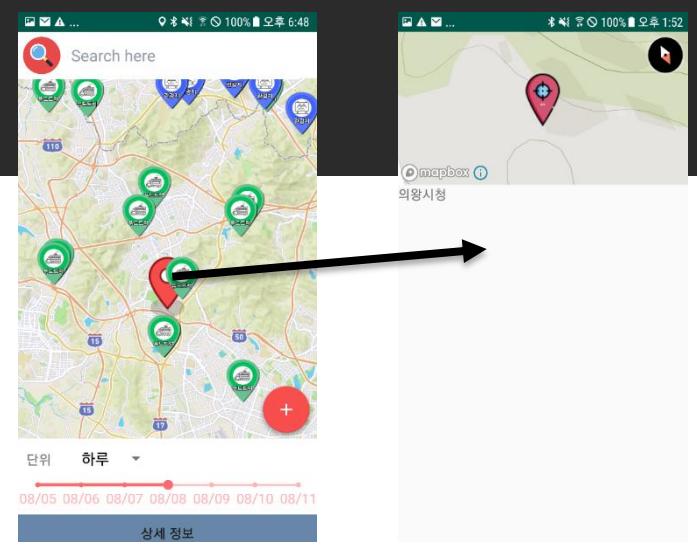
03. STUDY

3-(1)-1 Mapbox event (marker Click event)

Map marker 클릭 시 상세정보 화면으로 전환 및 데이터 전송
(해당 mark가 무엇인지 알기 위해 index 값 전달)

```
mapboxMap.addOnMapClickListener(new MapboxMap.OnMapClickListener() {
    @Override
    public boolean onMapClick(@NonNull LatLng point) {
        PointF pointf = mapboxMap.getProjection().toScreenLocation(point);
        RectF rectF = new RectF(pointf.x - 10, pointf.y - 10, pointf.x + 10, pointf.y + 10);
        List<Feature> Tourfeatures = mapboxMap.queryRenderedFeatures(rectF, LAYER_ID_Tour);
        if (!Tourfeatures.isEmpty()) {
            String name = Tourfeatures.get(0).getStringProperty("idx");
            Intent intent = new Intent(getApplicationContext(), DetailInfoActivity.class);
            startActivity(intent);

            return true;
        }
```

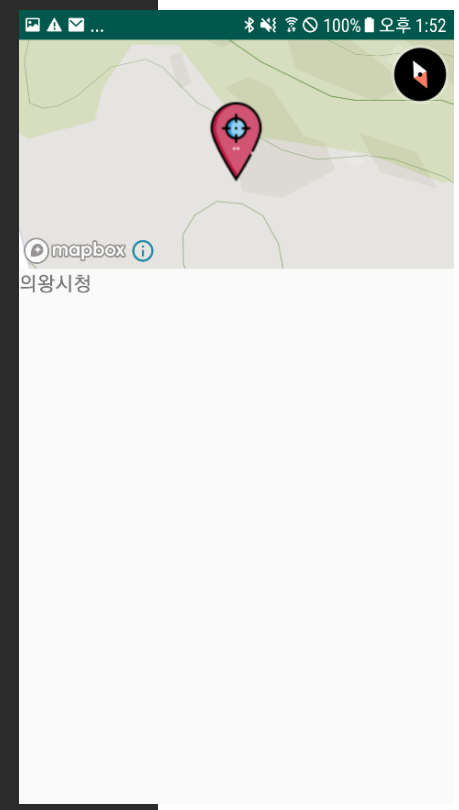
HomeActivity.java



03. STUDY

3-(1)-1 Mapbox event (marker Click event)

```
@Override
public void onResume() {
    super.onResume();
    mapView.onResume();
    Intent intent = getIntent();
    final TextView detailInfoTextView;
    detailInfoTextView = (TextView) findViewById(R.id.detailInfo_name_textView);
    idx = intent.getExtras().getString("idx");
    if (idx.equals("currentLocation")) {
        name = "현위치";
        longtitude = intent.getExtras().getDouble("Longitude");
        latitude = intent.getExtras().getDouble("Latitude");
        detailInfoTextView.setText(name);
    } else {
        mPlaceViewModel.getPlaceInfo(Integer.parseInt(idx)).observe(this, new Observer<Place>() {
            @Override
            public void onChanged(@Nullable Place place) {
                name = place.getPName();
                longtitude = place.getPLongitude();
                latitude = place.getPLatitude();
                detailInfoTextView.setText(name);

            }
        });
    }
}
```

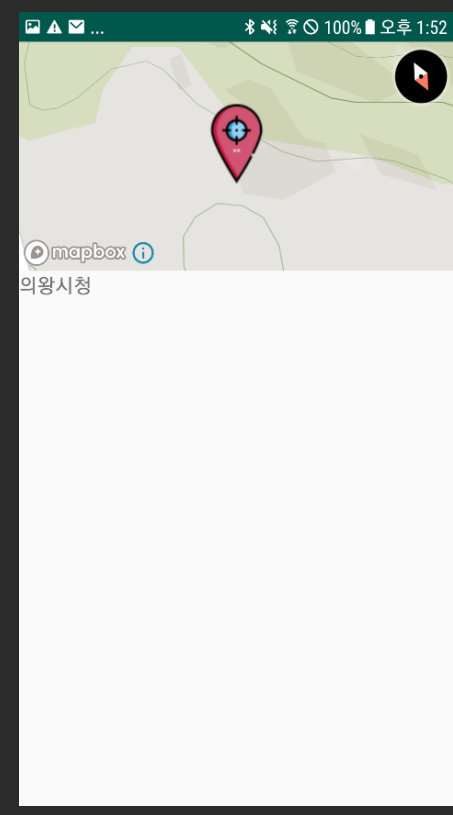


DetailInfoActivity.java

03. STUDY

3-(1)-1 Mapbox event (marker Click event)

```
@Override
public void onMapReady(@NonNull MapboxMap mapboxMap) {
    List<Feature> symbolLayerIconFeatureList = new ArrayList<>();
    symbolLayerIconFeatureList.add(Feature.fromGeometry(Point.fromLngLat(longtitude, latitude)));

    CameraPosition position = new CameraPosition.Builder()
            .target(new LatLng(latitude, longtitude))
            .zoom(13)
            .bearing(180)
            .tilt(30)
            .build();
    mapboxMap.animateCamera(CameraUpdateFactory
            .newCameraPosition(position), 5000);

    mapboxMap.setStyle(Style.OUTDOORS, new Style.OnStyleLoaded() {
        @Override
        public void onStyleLoaded(@NonNull Style style) {
            style.addImageAsync(ICON_ID, BitmapUtils.getBitmapFromDrawable(
                    getResources().getDrawable(R.drawable.blue_marker)));
            Source source = new GeoJsonSource(SOURCE_ID,
                    FeatureCollection.fromFeatures(symbolLayerIconFeatureList));
            style.addSource(source);
            SymbolLayer layer = new SymbolLayer(LAYER_ID, SOURCE_ID)
                    .withProperties(PropertyFactory.iconImage(ICON_ID), iconAllowOverlap(true), iconOffset(new Float[]{0f,
-9f}));
            style.addLayer(layer);
        }
    });
}
```



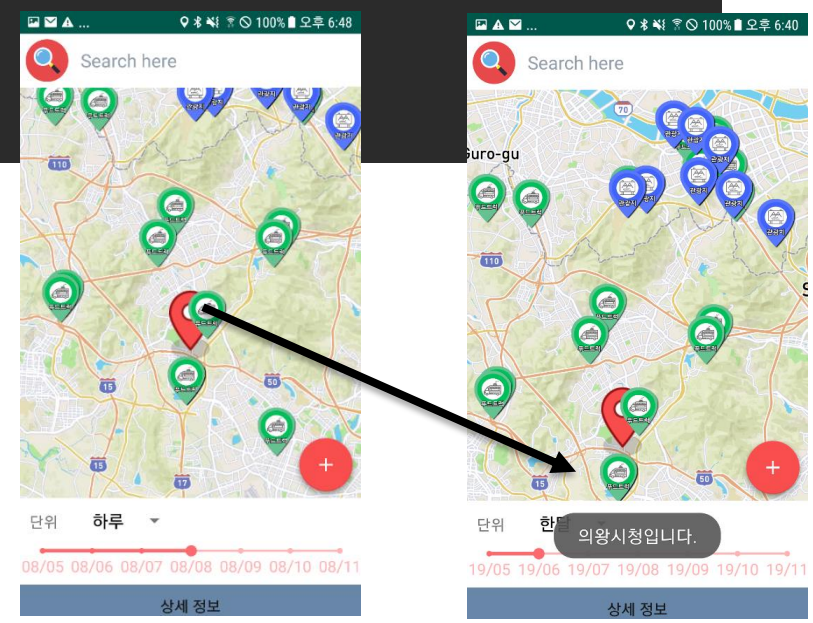
DetailInfoActivity.java

03. STUDY

3-(1)-1 Mapbox event (marker Click event)

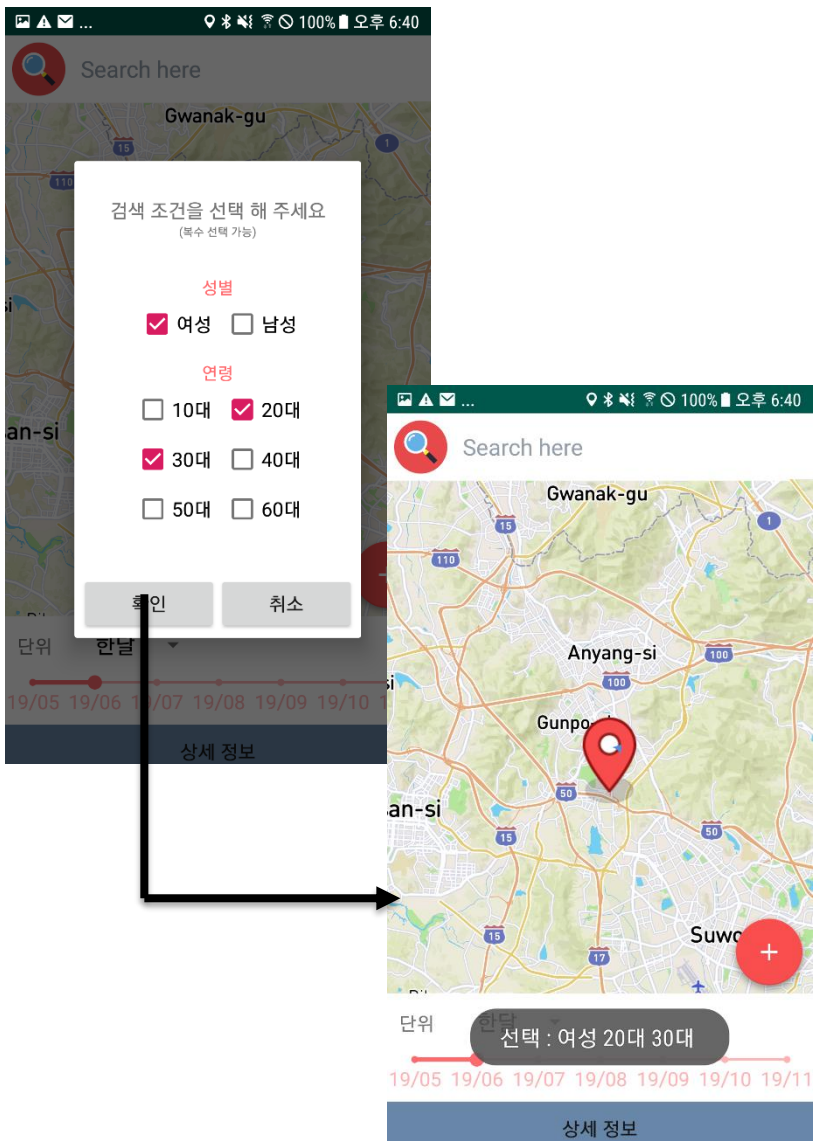
Map marker 길게 클릭 시 해당 marker의 이름 정보 출력

```
mapboxMap.addOnMapLongClickListener(new MapboxMap.OnMapLongClickListener() {
    @Override
    public boolean onMapLongClick(@NonNull LatLng point) {
        PointF pointf = mapboxMap.getProjection().toScreenLocation(point);
        RectF rectF = new RectF(pointf.x - 10, pointf.y - 10, pointf.x + 10, pointf.y + 10);

        List<Feature> Tourfeatures = mapboxMap.queryRenderedFeatures(rectF, LAYER_ID_Tour);
        if (!Tourfeatures.isEmpty()) {
            String name = Tourfeatures.get(0).getStringProperty("name");

            Toast.makeText(HomeActivity.this, name + "입니다.",
                    Toast.LENGTH_SHORT).show();
            return true;
        }
```



03. STUDY

3-(1)-2. 상세 검색 (custom Dialog)




```java
// jieun  custom Dialog
public void setCustomDialogSearh() {
    Button homeSearchBtn;
    homeSearchBtn = (Button) findViewByld(R.id.home_search_btn);
    homeSearchBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            CustomDialogSearch customDialogSearch = new CustomDialogSearch(HomeActivity.this);
            customDialogSearch.setDialogListener(new
CustomDialogSearch.CustomDialogSearchListener() {
                @Override
                public void onPositiveClicked(String result) {
                    if (result.equals("")) {
                        Toast.makeText(getApplicationContext(), "선택사항 없음",
Toast.LENGTH_SHORT).show();
                        customDialogSearch.dismiss();
                    } else {
                        Toast.makeText(getApplicationContext(), "선택 : " + result,
Toast.LENGTH_SHORT).show();
                        customDialogSearch.dismiss();
                    }
                }

                @Override
                public void onNegativeClicked() {
                    Toast.makeText(getApplicationContext(), "취소", Toast.LENGTH_SHORT).show();
                    customDialogSearch.dismiss();
                }
            });
            customDialogSearch.show();
        }
    });
}
```

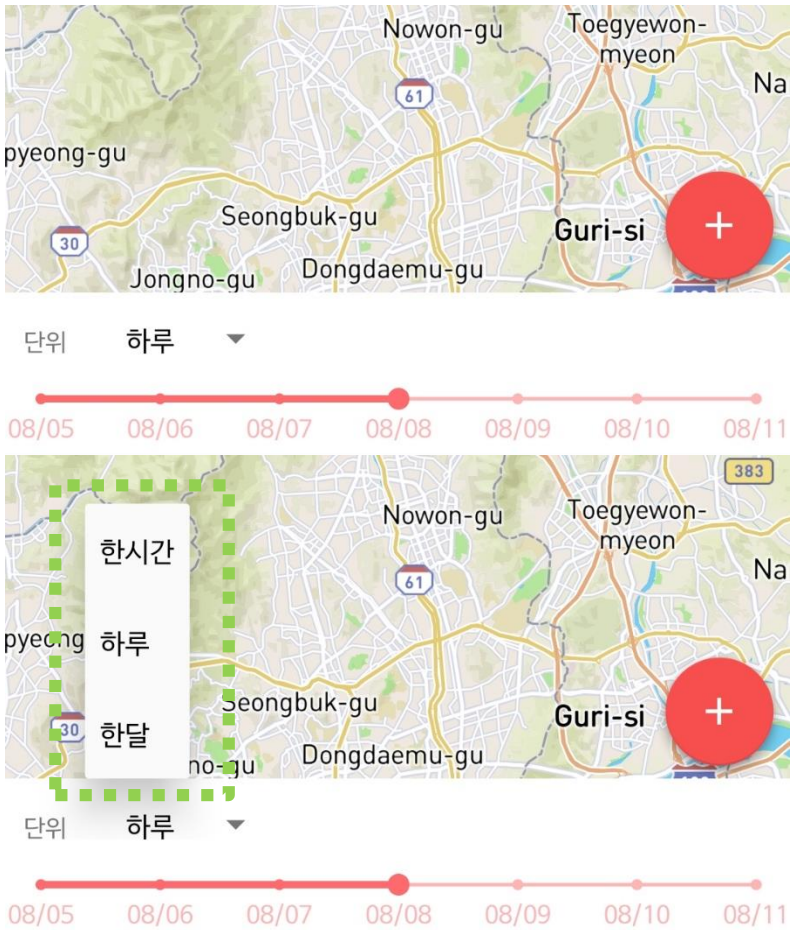
HomeActivity.java

- CustomDialogSearch.java -> 커스텀 다이얼 로그 구현 (값 전달 위해 리스너, 인터페이스 구현)

- 선택한 값 toast massage 나옴

03. STUDY

3-(2)-1. Custom Progress Bar & Combo Box



1. BubbleSeekbar.java
   – 커스텀 프로그래스바 구현

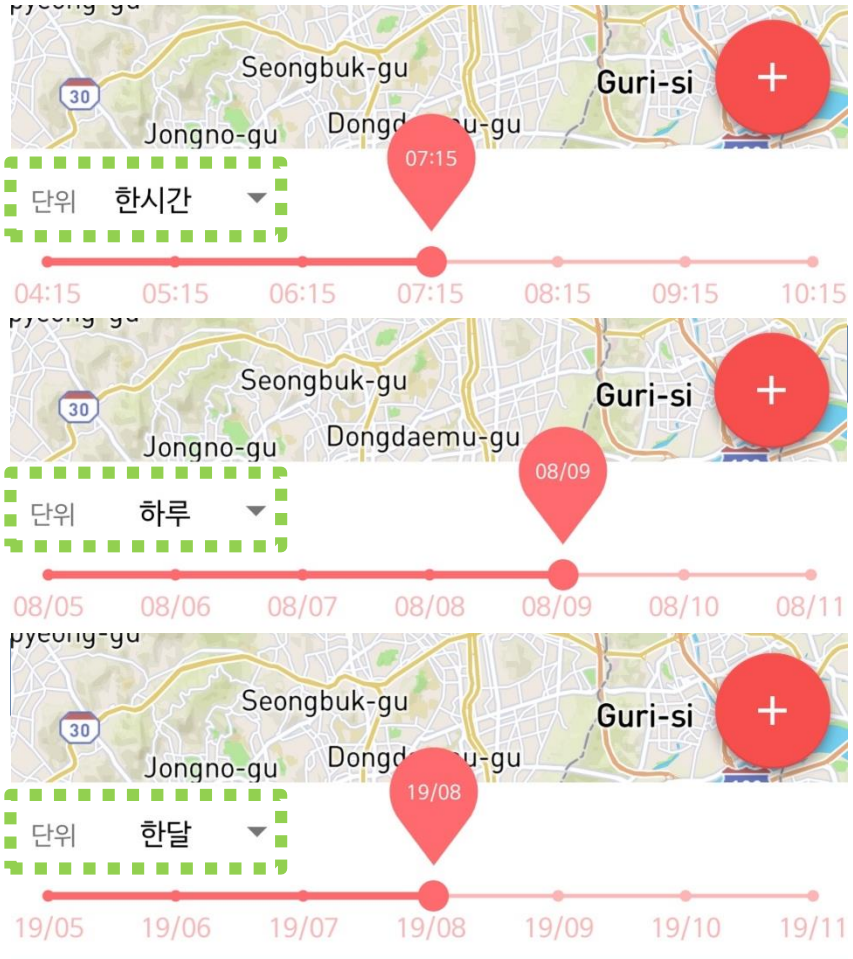
```
for (int i = 0; i <= mSectionCount; i++) {
     if ((getProgress() >= mSectionValue * i) && (getProgress() < mSectionValue * i + 1)) {
        mBubbleView.setProgressText(mSectionTextArray.get(i));
     }
}
```

2. HomeActivity.java – BubbleSeekBar
   – Default로 ‘하루’ 단위의 유동 인구를 볼 수 있게 함

```
// sol BubbleSeekBar
bottomSheetBehavior.getPeekHeight();

final BubbleSeekBar bubbleSeekBar3 = findViewById(R.id.demo_4_seek_bar_3);

bubbleSeekBar3.setProgress(storedValue);

SimpleDateFormat sdf_hour = new SimpleDateFormat("hh:mm");
SimpleDateFormat sdf_day = new SimpleDateFormat("MM/dd");
SimpleDateFormat sdf_month = new SimpleDateFormat("yy/MM");
Calendar cal = Calendar.getInstance();

bubbleSeekBar3.setCustomSectionTextArray(new BubbleSeekBar.CustomSectionTextArray() {
    @NonNull
    @Override
    public SparseArray<String> onCustomize(int sectionCount, @NonNull
SparseArray<String> array) {
        array.clear();
        cal.add(Calendar.DAY_OF_MONTH, -3);
        array.put(0, sdf_day.format(cal.getTime()));
        for (int i = 1; i < 7; i++) {
            cal.add(Calendar.DAY_OF_MONTH, +1);
            array.put(i, sdf_day.format(cal.getTime()));
        }
        return array;
    }
});
```

03. STUDY

3-(2)-1. Custom Progress Bar & Combo Box

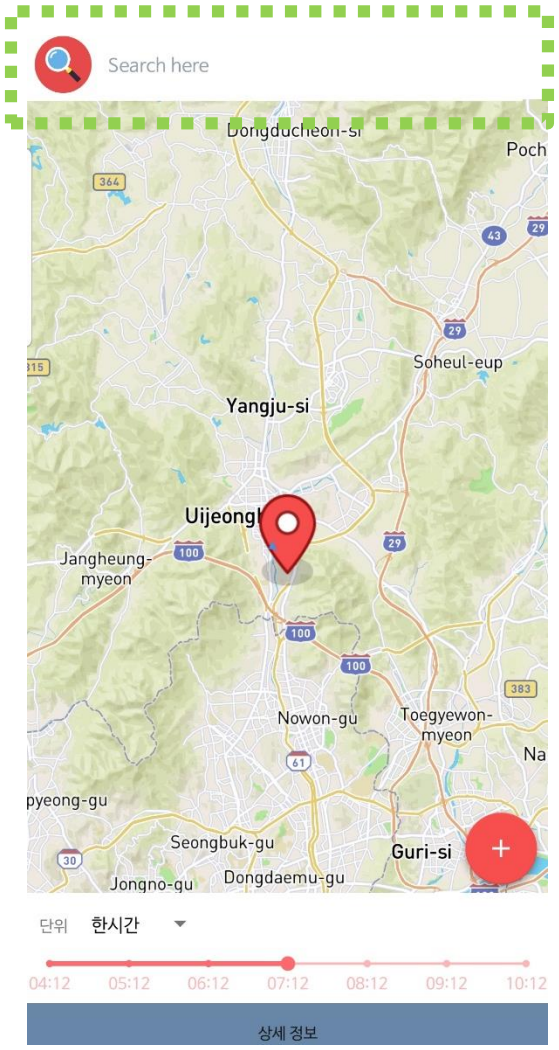


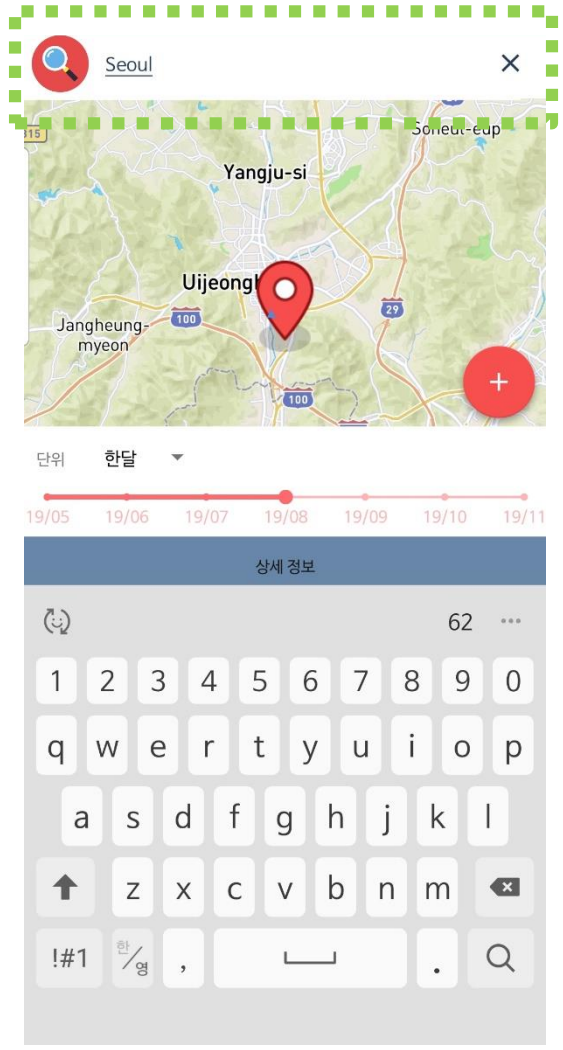
3. HomeActivity.java – Spinner
   – 콤보 박스의 값에 따라,
   프로그래스 바의 TextArray를
   수정할 수 있는 함수를 호출

```java
@Override
public void onItemSelected(AdapterView<?> adapterView, View view, int position, long id) {
    Toast.makeText(HomeActivity.this, (String) sAdapter.getItem(position), Toast.LENGTH_SHORT).show();
    if(spinner.getSelectedItemPosition() == 0) {
        bubbleSeekBar3.setCustomSectionTextArray(new BubbleSeekBar.CustomSectionTextArray() {
            @NonNull
            @Override
            public SparseArray<String> onCustomize(int sectionCount, @NonNull SparseArray<String> array) {
                array.clear();
                Calendar cal = Calendar.getInstance();
                cal.add(Calendar.HOUR, -3);
                array.put(0, sdf_hour.format(cal.getTime()));
                for (int i = 1; i < 7; i++) {
                    cal.add(Calendar.HOUR, +1);
                    array.put(i, sdf_hour.format(cal.getTime()));
                }
                return array;
            }
        });
    } else if (spinner.getSelectedItemPosition() == 1) {
        bubbleSeekBar3.setCustomSectionTextArray(new BubbleSeekBar.CustomSectionTextArray() {
            @NonNull
            @Override
            public SparseArray<String> onCustomize(int sectionCount, @NonNull SparseArray<String> array) {
                array.clear();
                Calendar cal = Calendar.getInstance();
                cal.add(Calendar.DAY_OF_MONTH, -3);
                array.put(0, sdf_day.format(cal.getTime()));
                for (int i = 1; i < 7; i++) {
                    cal.add(Calendar.DAY_OF_MONTH, +1);
                    array.put(i, sdf_day.format(cal.getTime()));
                }
                return array;
            }
        });
    } else if (spinner.getSelectedItemPosition() == 2) {
        bubbleSeekBar3.setCustomSectionTextArray(new BubbleSeekBar.CustomSectionTextArray() {
            @NonNull
            @Override
            public SparseArray<String> onCustomize(int sectionCount, @NonNull SparseArray<String> array) {
                array.clear();
                Calendar cal = Calendar.getInstance();
                cal.add(Calendar.MONTH, -3);
                array.put(0, sdf_month.format(cal.getTime()));
                for (int i = 1; i < 7; i++) {
                    cal.add(Calendar.MONTH, +1);
                    array.put(i, sdf_month.format(cal.getTime()));
                }
                return array;
            }
        });
```

03. STUDY

3-(2)- 2. Mapbox Search Bar






```
PlaceAutocompleteFragment autocompleteFragment;

if (savedInstanceState == null) {
    autocompleteFragment =
PlaceAutocompleteFragment.newInstance(getString(R.string.mapbo
x_access_token));

    final FragmentTransaction transaction =
getSupportFragmentManager().beginTransaction();
    transaction.add(R.id.mapSearchBar, autocompleteFragment,
PlaceAutocompleteFragment.TAG);
    transaction.commit();
} else {
    autocompleteFragment =
(PlaceAutocompleteFragment)getSupportFragmentManager().findFra
gmentByTag(PlaceAutocompleteFragment.TAG);
}
autocompleteFragment.setOnPlaceSelectedListener(new
PlaceSelectionListener() {
    @Override
    public void onPlaceSelected(CarmenFeature carmenFeature) {
        Toast.makeText(HomeActivity.this,
                carmenFeature.text(),
Toast.LENGTH_LONG).show();
        finish();
    }
```

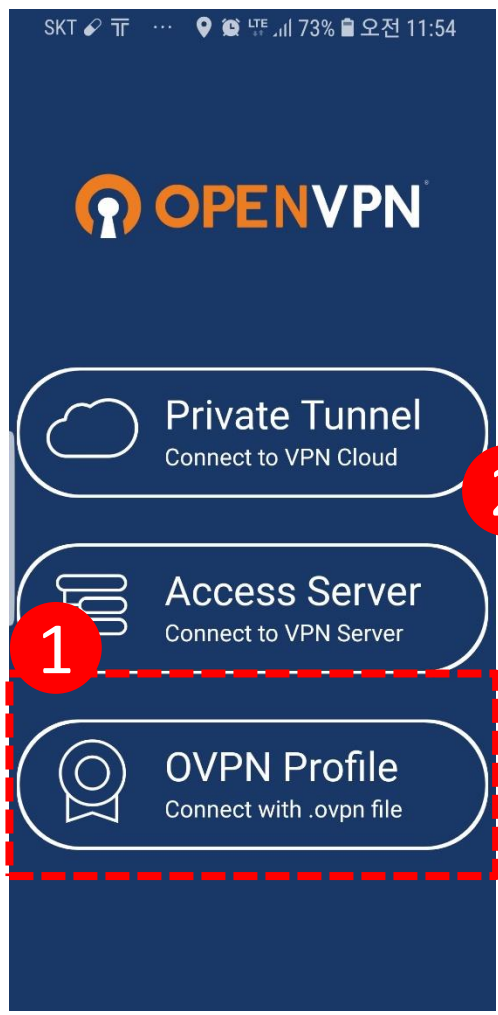
4. HomeActivity.java – Search Bar
   - 지금은 글자를 쓸 수 만 있는 서치바가 나옴
     맵박스 깃허브를 참조하여 코드를 수정할 예정

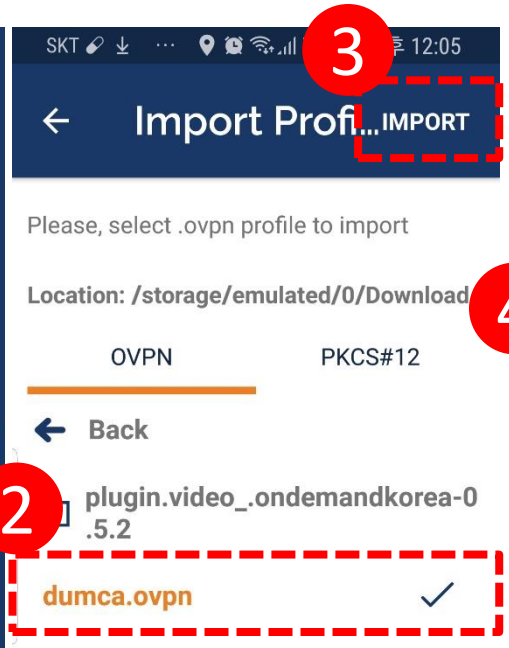
03. STUDY

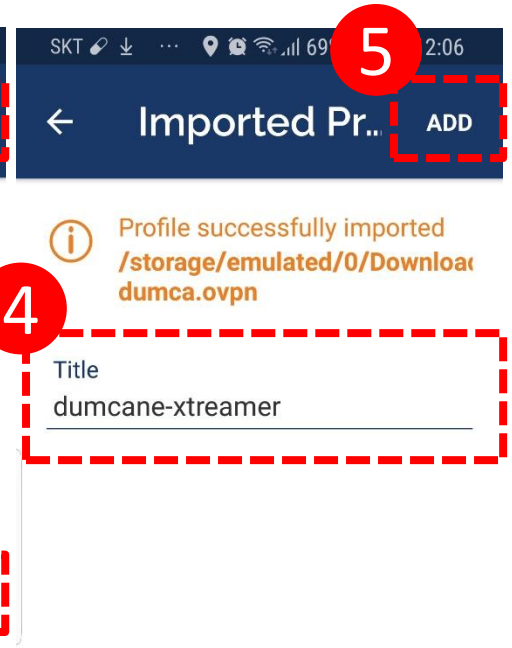
3-(2)-3. OpenVPN 서버를 안드로이드 스마트폰 클라이언트에서 접속하는 방법

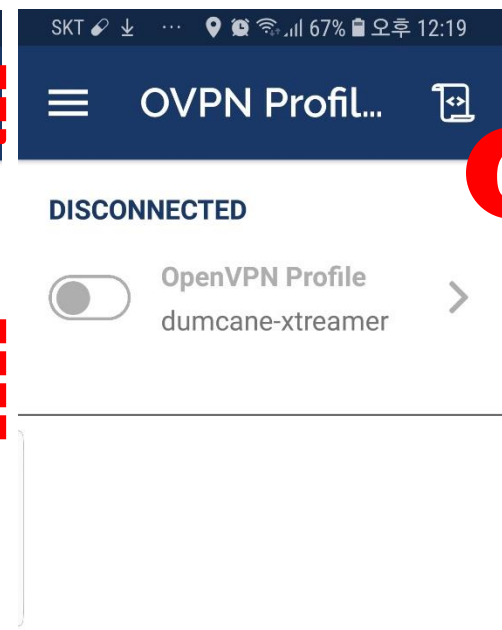
1. OpenVPN Connect 앱 설치

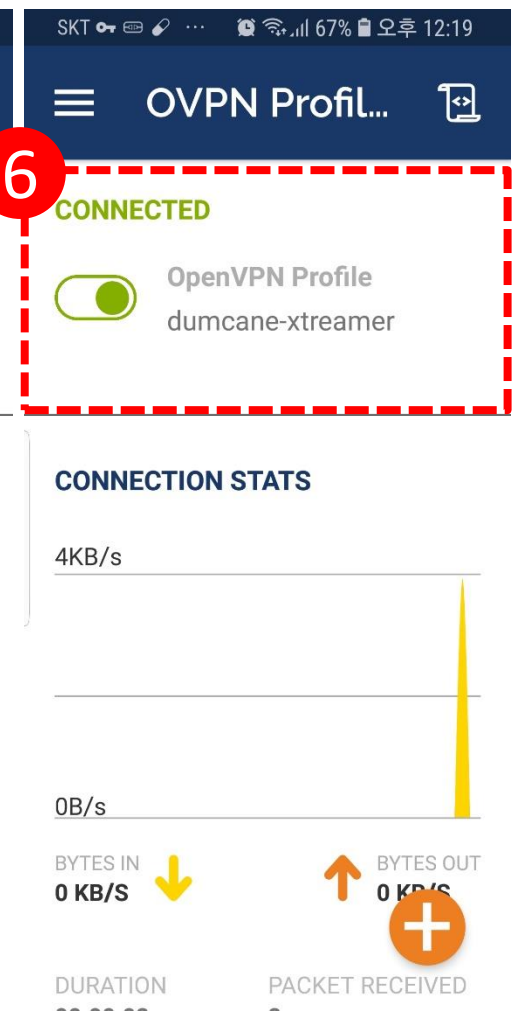
* OpenVPN Connect 앱에서 클라이언트 설정 파일을 .ovpn를 사용하므로, client.conf 파일을 client.ovpn으로 변경하여 스마트폰으로 옮긴 후에 OpenVPN 앱을 실행
(kdeconnect의 파일보내기 기능을 이용해도 됨)











2. 클라이언트 설정 파일 import

OVPN를 선택하여 Downloads 경로로
이동하여 다운로드한 .ovpn를 지정한 후 -〉 프로파일명을 지정한 후
오른쪽 상단에 있는 [IMPORT]를 누름 [ADD]를 터치

3. 접속 상태 페이지

추가된 프로파일에서 연결을 누르면
OpenVPN 서버에 접속이 이뤄지면서
접속 상태 페이지가 나타남

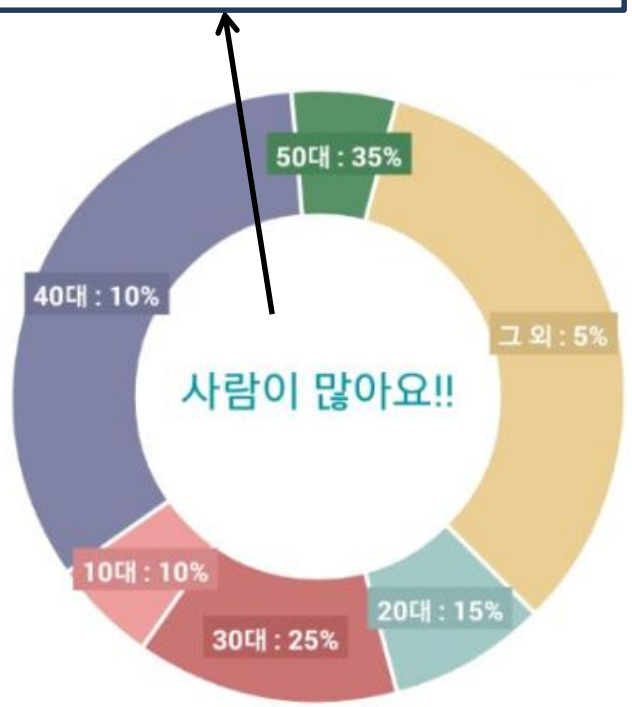
4. OpenVPN 서버 접속

스마트폰에서 OpenVPN 서버에
접속이 이뤄지게 되면 네트웍이
다른 스마트폰과 리눅스 서버 간에
kdeconnect를 연결 가능

03. STUDY

3-(3)-1 안드로이드 차트 라이브러리 MPAndroidChart - Pie Chart

이 안에 글을 작성할 수 있음.
화면 내 유동인구 수를 보여 줄 수도 있을 것으로 예상



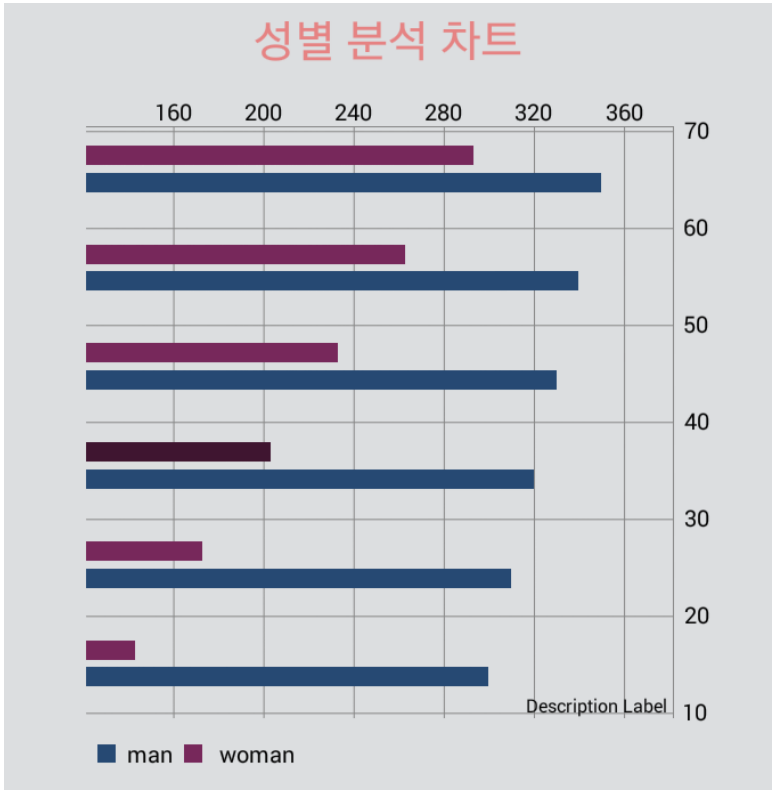

```
public void create_pie_chart(){
    PieChartView pieChartview;
    pieChartview= findViewById(R.id.pie_chart);

    List pieData = new ArrayList<>();
    pieData.add(new SliceValue( value: 15, Color.parseColor( colorString: "#a3c9c7")).setLabel("20대 : 15%"));
    pieData.add(new SliceValue( value: 25, Color.parseColor( colorString: "#cb7575")).setLabel("30대 : 25%"));
    pieData.add(new SliceValue( value: 10, Color.parseColor( colorString: "#ef9e9f")).setLabel("10대 : 10%"));
    pieData.add(new SliceValue( value: 60, Color.parseColor( colorString: "#8283a7")).setLabel("40대 : 10%"));
    pieData.add(new SliceValue( value: 10, Color.parseColor( colorString: "#589167")).setLabel("50대 : 35%"));
    pieData.add(new SliceValue( value: 60, Color.parseColor( colorString: "#ebce95")).setLabel("그 외 : 5%"));

    PieChartData pieChartData = new PieChartData(pieData);
    pieChartData.setHasLabels(true).setValueLabelTextSize(12);

    //원 안에 텍스트 넣을 수 있는 코드
    pieChartData.setHasCenterCircle(true).setCenterText1("사람이 많아요!!")
            .setCenterText1FontSize(20).setCenterText1Color(Color.parseColor( colorString: "#0097A7"));
    pieChartview.setPieChartData(pieChartData);

}
```

리스트에 값을 넣고, 그 값을 PieChartData 메소드에 넣어주면 자동으로 Pie chart가 생성됩니다.

또한, 파이 차트 원 안에 setCenterText()를 이용해 텍스트도 넣을 수 있습니다.

데이터 마다 색도 지정 가능해 예쁜 차트 디자인이 가능할 것으로 예상됩니다.

03. STUDY 3-(3)-2 안드로이드 차트 라이브러리 MPAndroidChart - BarChart



1. BarEntry 객체 리스트를 생성한 후 데이터 입력 <- 임시로 임의 데이터 저장

2. 차트내에 데이터가 있거나 데이터를 설정한 적 있으면 변경된 데이터에 대해서만 처리

3. BarDataSet 객체를 이용해 차트내에 데이터를 설정해줌

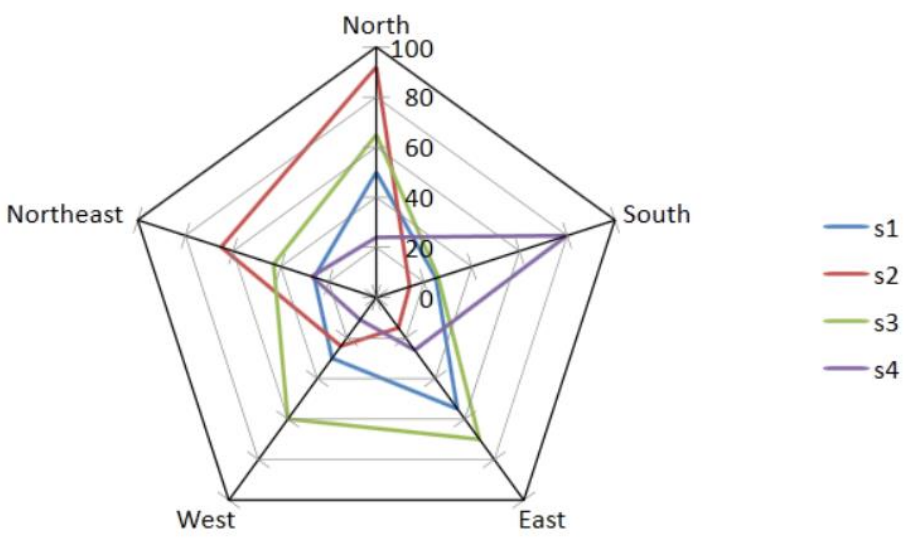
4. 해당 객체를 배열에 넣어 BarData 객체로 차트에 저장

```
List<BarEntry> woman_List = new ArrayList<~>();

for (int i = startYear; i <= endYear; i+=10) {
    woman_List.add(new BarEntry(i, n));
    n+= 10;
}

if (barChart.getData() != null && barChart.getData().getDataSetCount() > 0) {
    set1 = (BarDataSet) barChart.getData().getDataSetByIndex(0);
    set2 = (BarDataSet) barChart.getData().getDataSetByIndex(1);

    set1.setValues(man_List);
    set2.setValues(woman_List);

    barChart.getData().notifyDataChanged();
    barChart.notifyDataSetChanged();

} else {
    set1 = new BarDataSet(man_List, label: "man");
    set1.setColors(Color.parseColor( colorString: "#264973"));

    set2 = new BarDataSet(woman_List, label: " woman");
    set2.setColors(Color.parseColor( colorString: "#77285B"));

    ArrayList<IBarDataSet> dataSets = new ArrayList<~>();
    dataSets.add(set1);
    dataSets.add(set2);

    BarData data = new BarData(dataSets);
    barChart.setData(data);
}
barChart.getBarData().setBarWidth(barWidth);
barChart.groupBars(startYear, groupSpace, barSpace);
```

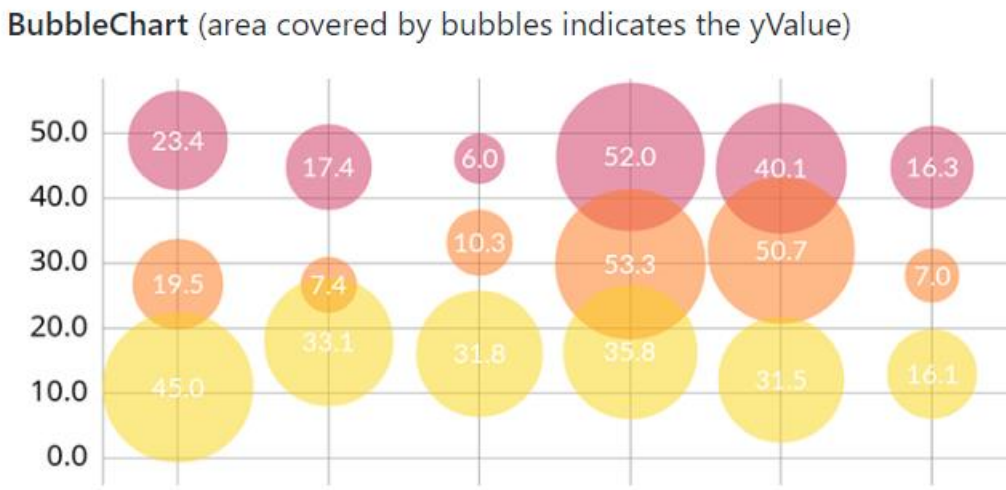
첫번째 인자는 x축 두번째 인자는 y축

03. STUDY

3-(3)-3 구현 가능한 추가 차트 종류



Radar chart



Bubble chart

07. Question

질문사항

1. Long Click Event, click Event 구분
2. 상세정보 페이지 작동 방식