



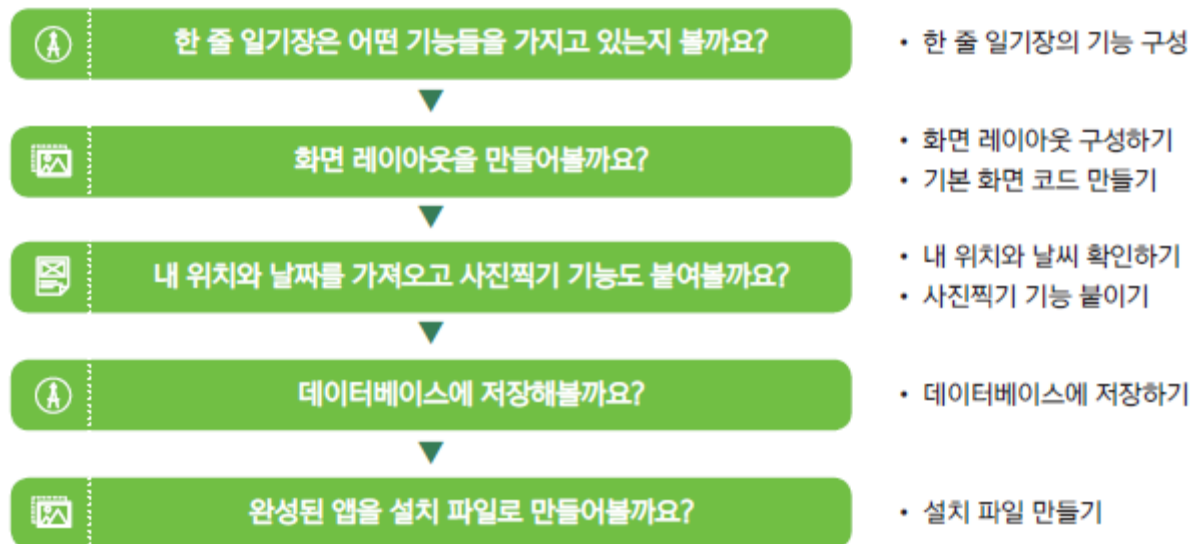
안드로이드 앱 프로그래밍

강사 : 정지영



안드로이드 앱 프로그래밍

한 줄 일기장 앱 만들기



1.

한 줄 일기장 앱 제작에 필요한 기획



데이터베이스 처리 방식

- 리스트 화면과 입력 화면에서 데이터베이스 접근





앱 구성 과정

• 단계별로 앱 제작 진행

1단계

- ▶ 화면 구성
 - 리스트 화면
 - 입력 화면
 - 통계 화면

2단계

- ▶ 내 위치 확인
- ▶ 날씨 연동
 - Volley & Gson
- ▶ 사진 연동
 - 사진 찍기
 - 앨범에서 선택

3단계

- ▶ 데이터베이스 연동
 - 리스트 조회
 - 저장
 - 수정
 - 삭제
 - 통계 조회

4단계

- ▶ 마무리
 - 앱 이름
 - 앱 아이콘
 - 스플래시 화면



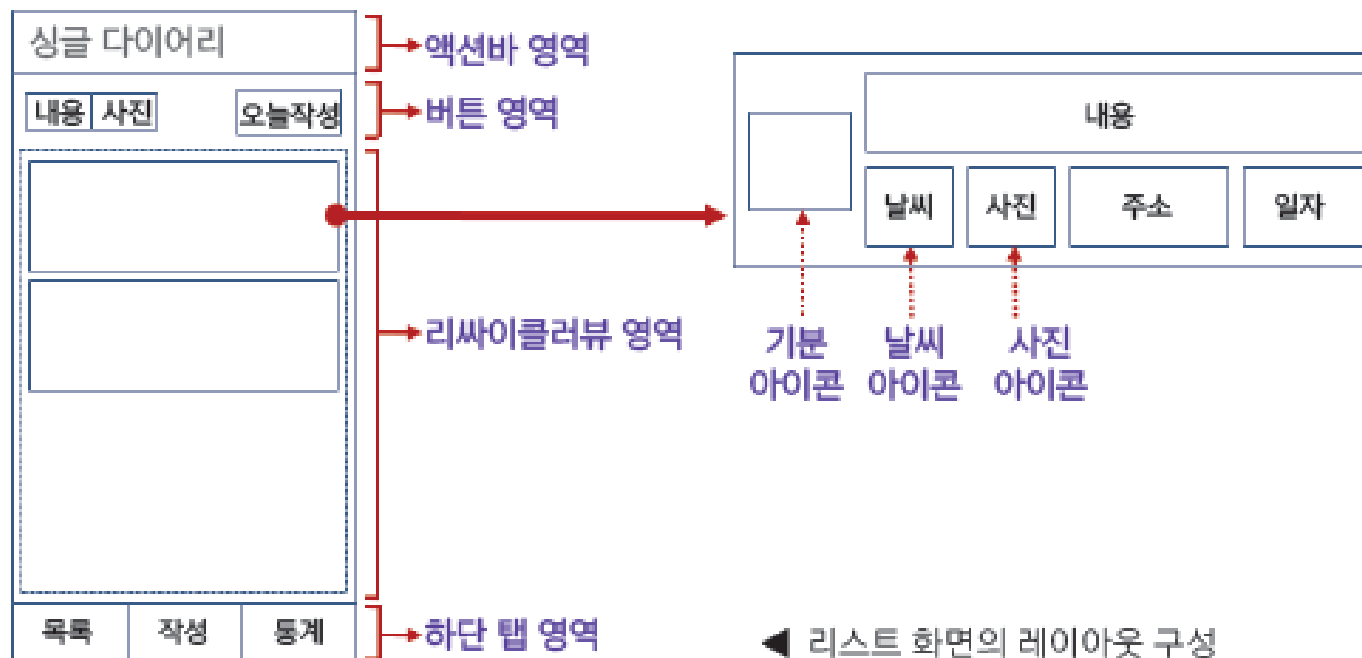
2.

화면 구성하기



리스트 화면의 레이아웃 구성

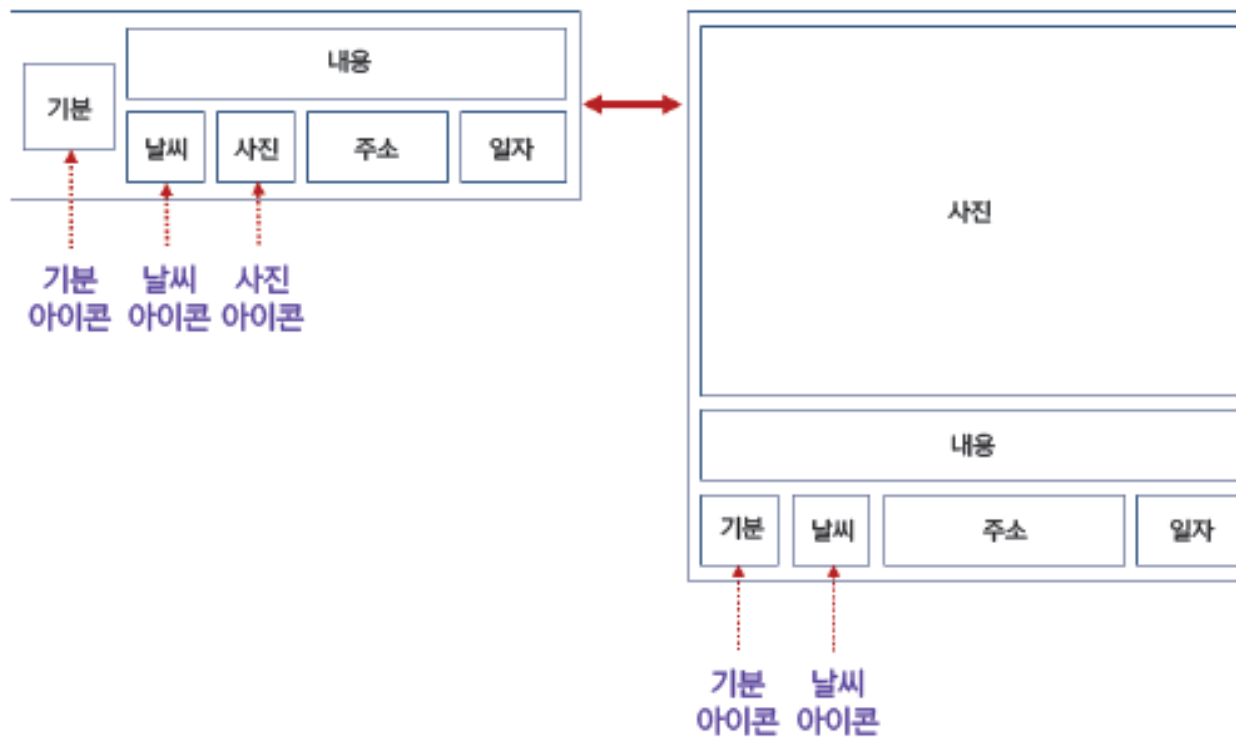
- 리싸이클러뷰를 이용해 일기 리스트 표시





리스트 화면의 레이아웃 구성

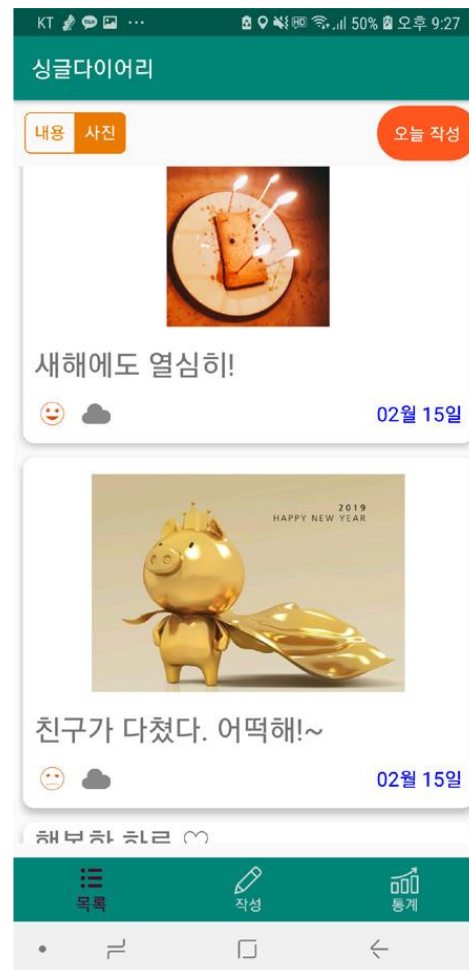
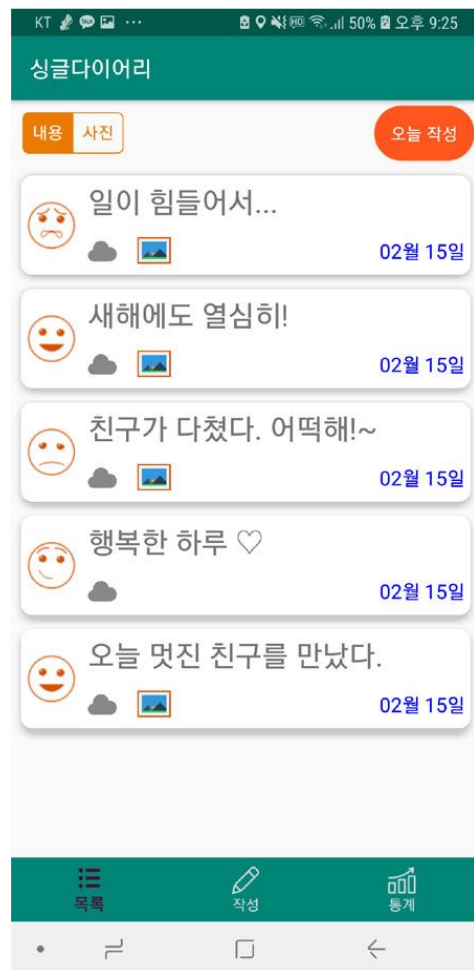
- 레이아웃을 두 가지로 표시





리스트 화면의 레이아웃 구성

- 레이아웃을 두 가지로 표시





외부 라이브러리 추가

- 리사이클러뷰, 카드뷰 등을 위한 외부 라이브러리 추가

참조파일 SingleDiary>/Gradle Scripts/build.gradle (Module: app)

중략...

dependencies {

중략...

implementation 'com.android.support:design:28.0.0'

implementation 'com.android.support:recyclerview-v7:28.0.0'

implementation 'com.android.support:cardview-v7:28.0.0'

}



메인 화면을 위한 XML 레이아웃 구성

- 프레임 레이아웃과 하단 탭

참조파일 SingleDiary>/app/res/layout/activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <FrameLayout
        android:id="@+id/container"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_behavior="@string/appbar_scrolling_view_behavior" />

    <android.support.design.widget.BottomNavigationView
        android:id="@+id/bottom_navigation"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginEnd="0dp"
        android:layout_marginStart="0dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent">
```



메인 화면을 위한 XML 레이아웃 구성

- 하단 탭의 메뉴를 위한 XML 구성

참조파일 SingleDiary>/app/res/menu/menu_bottom.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">

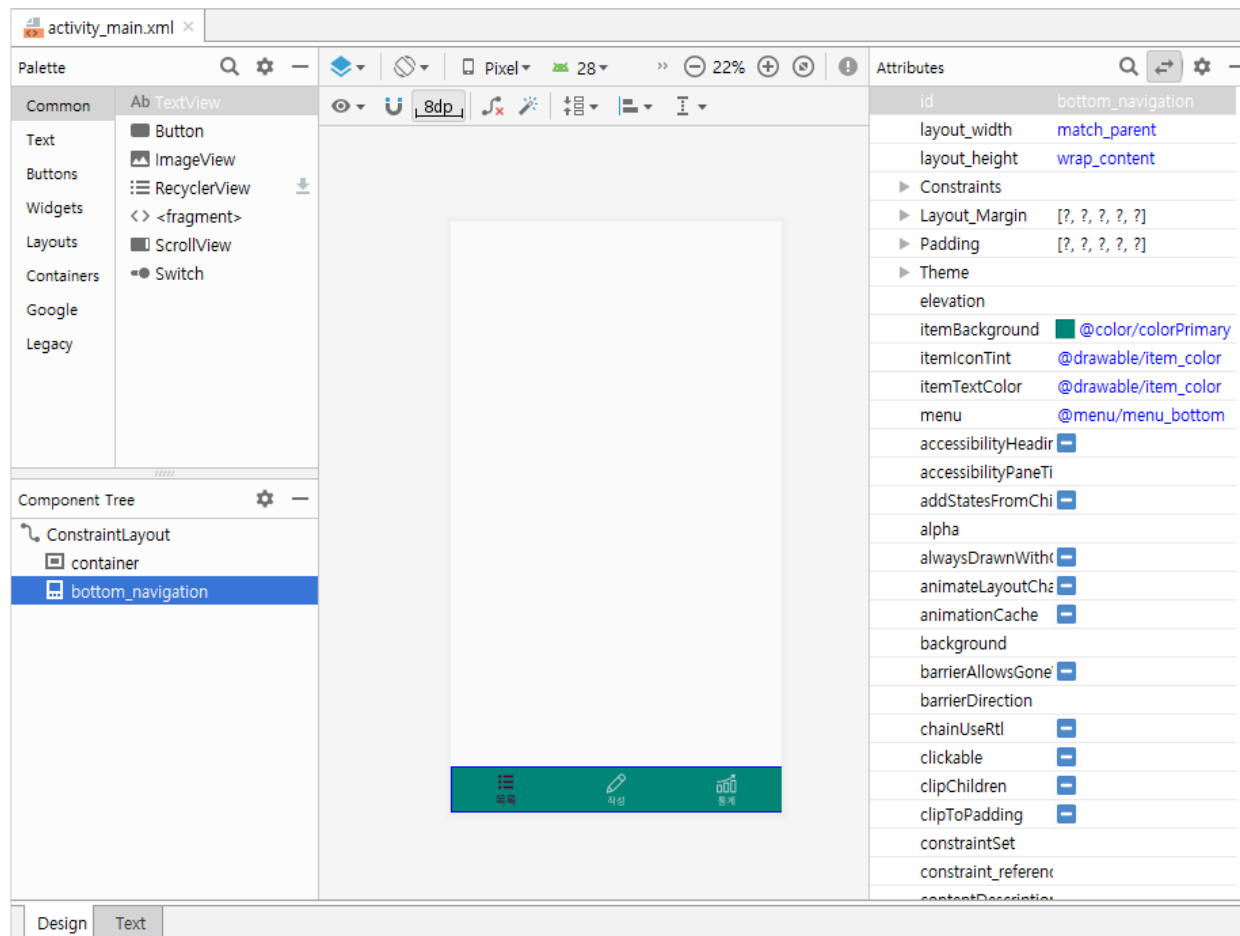
    <item
        android:id="@+id/tab1"
        app:showAsAction="ifRoom"
        android:enabled="true"
        android:icon="@drawable/list_48"
        android:title="목록" />
    <item
        android:id="@+id/tab2"
        app:showAsAction="ifRoom"
        android:icon="@drawable/write_48"
        android:title="작성" />
    <item
        android:id="@+id/tab3"
        app:showAsAction="ifRoom"
        android:enabled="true"
        android:icon="@drawable/graph_48"
        android:title="통계" />

</menu>
```



메인 화면을 위한 XML 레이아웃 구성

- 보이는 형태





메인 화면을 위한 XML 레이아웃 구성

- 프래그먼트 3개 만들어 추가

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    fragment1 = new Fragment1();
    fragment2 = new Fragment2();
    fragment3 = new Fragment3();

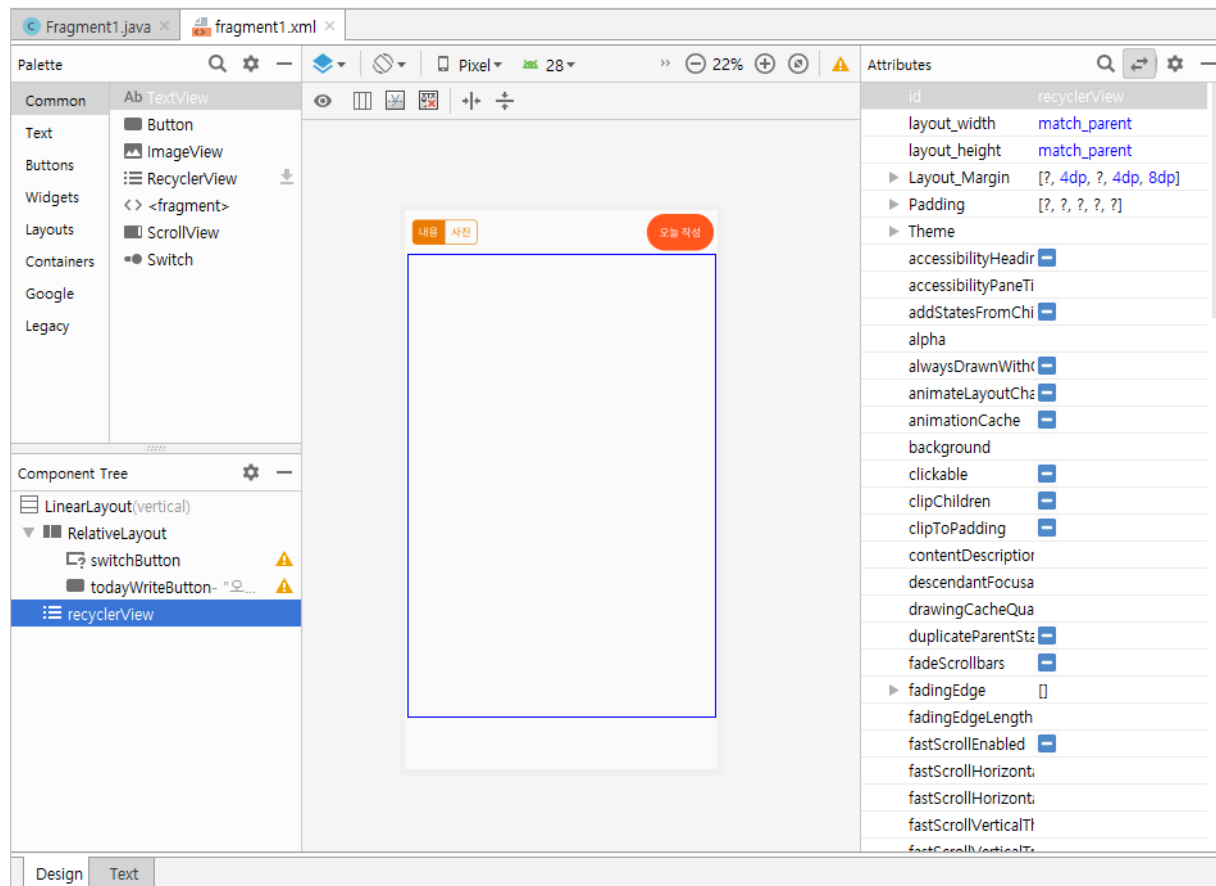
    getSupportFragmentManager().beginTransaction().replace(R.id.container, fragment1).commit();

    bottomNavigation = findViewById(R.id.bottom_navigation);
    bottomNavigation.setOnNavigationItemSelectedListener(
        new BottomNavigationView.OnNavigationItemSelectedListener() {
            @Override
            public boolean onNavigationItemSelected(@NonNull MenuItem item) {
                switch (item.getItemId()) {
                    case R.id.tab1:
                        Toast.makeText(getApplicationContext(), "첫 번째 탭 선택됨", Toast.LENGTH_LONG).show();
                        getSupportFragmentManager().beginTransaction().replace(R.id.container, fragment1).commit();
```




리스트 화면 구성

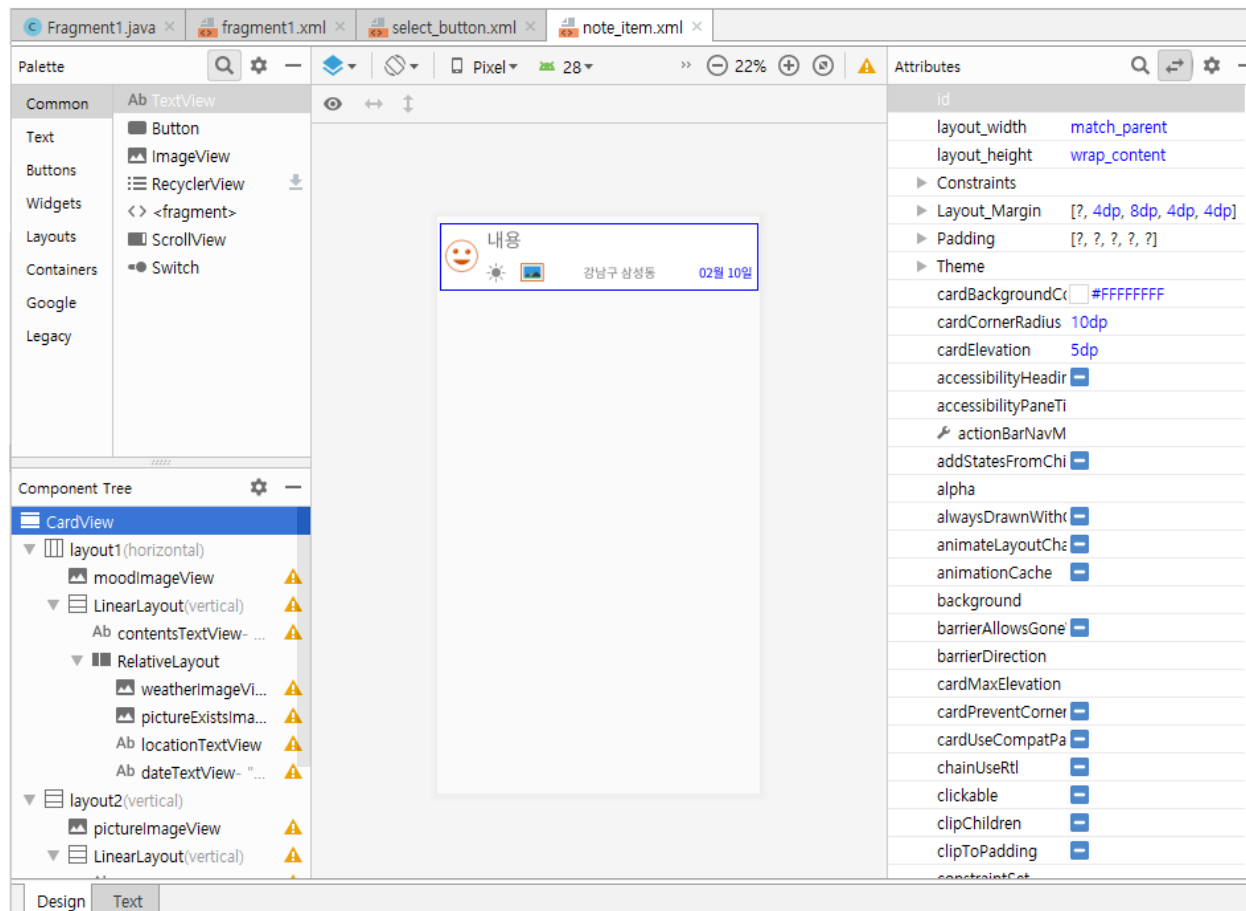
- 리싸이클러뷰로 보여줄 화면 구성





리스트 화면 구성

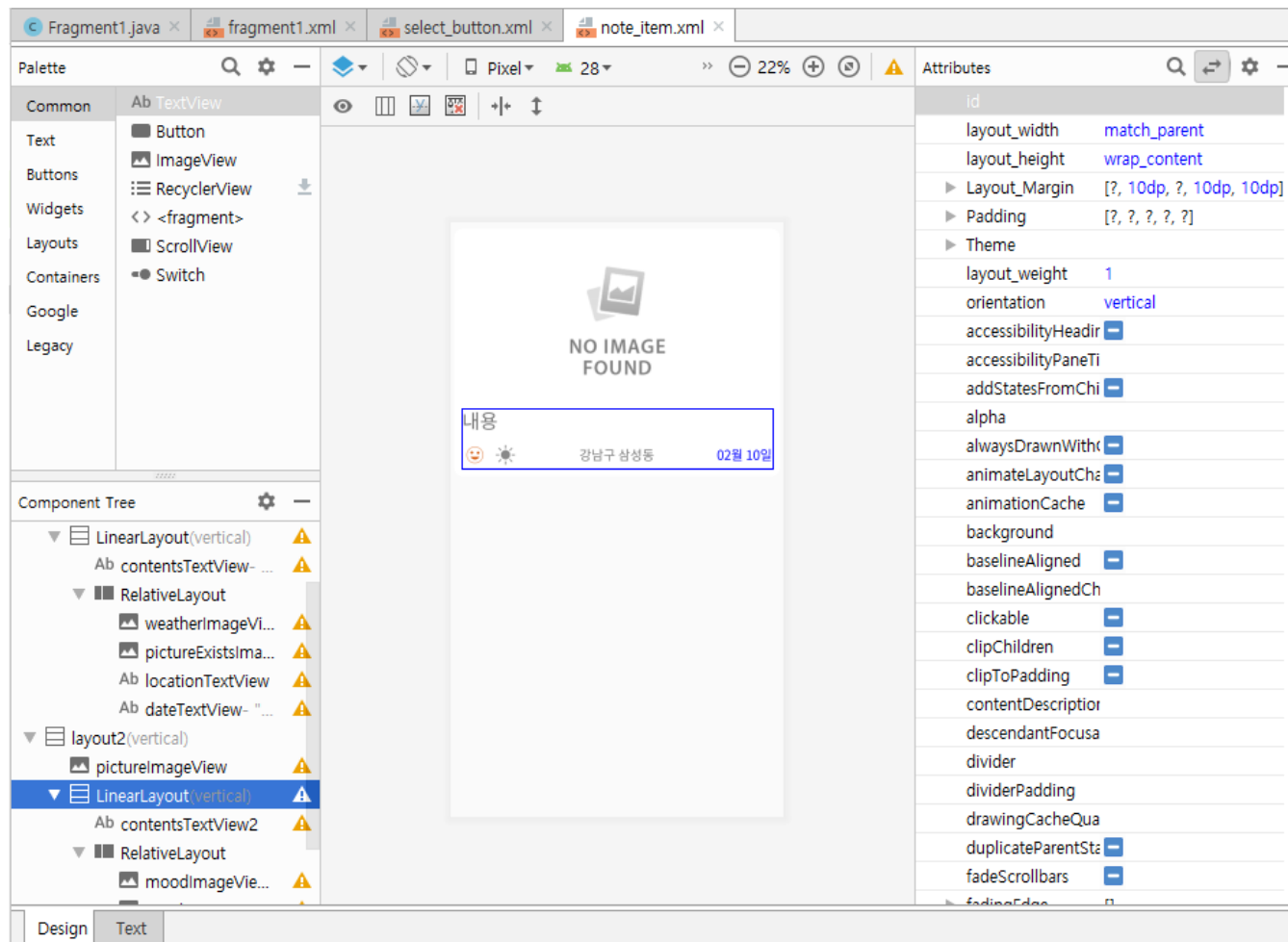
- 리싸이클러뷰의 아이템을 위한 레이아웃





리스트 화면 구성

- 리싸이클러뷰의 아이템을 위한 레이아웃 2





리스트 화면 구성

- 어댑터 코드 구성

참조파일 SingleDiary>/app/java/org.techtown.diary/NoteAdapter.java

```
public class NoteAdapter extends RecyclerView.Adapter<NoteAdapter.ViewHolder>
    implements OnNoteItemClickListener {
    ArrayList<Note> items = new ArrayList<Note>();

    OnNoteItemClickListener listener;

    int layoutType = 0;

    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int viewType) {
        LayoutInflater inflater = LayoutInflater.from(viewGroup.getContext());
        View itemView = inflater.inflate(R.layout.note_item, viewGroup, false);

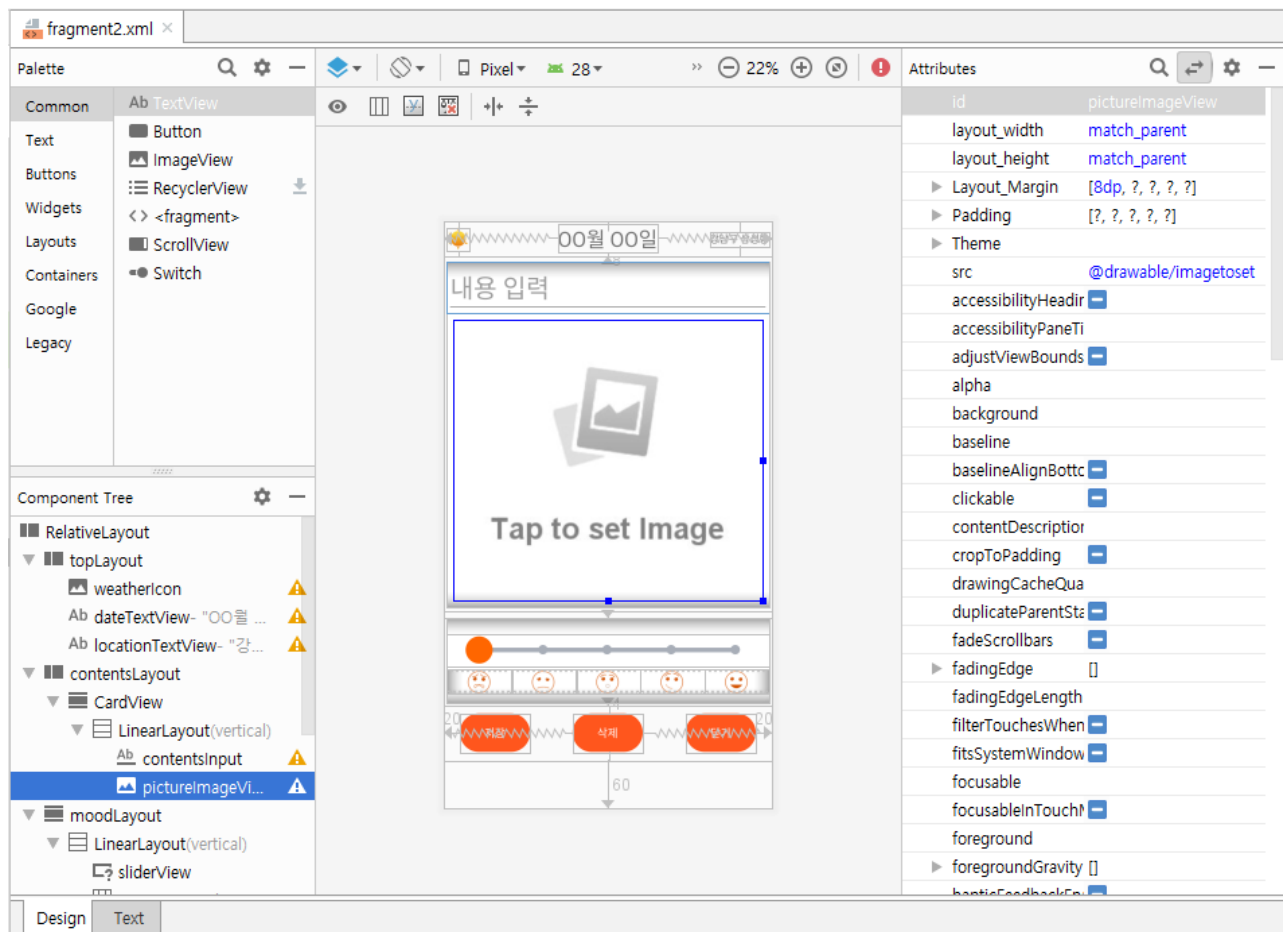
        return new ViewHolder(itemView, this, layoutType);
    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolder viewHolder, int position) {
        Note item = items.get(position);
        viewHolder.setItem(item);
        viewHolder.setLayoutType(layoutType);
    }
}
```



입력 화면 구성

- 입력 화면을 위한 레이아웃 구성





입력 화면을 위한 코드 추가

- 입력 화면의 코드 구성

```
Button closeButton = rootView.findViewById(R.id.closeButton);
closeButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (listener != null) {
            listener.onTabSelected(0);
        }
    }
});

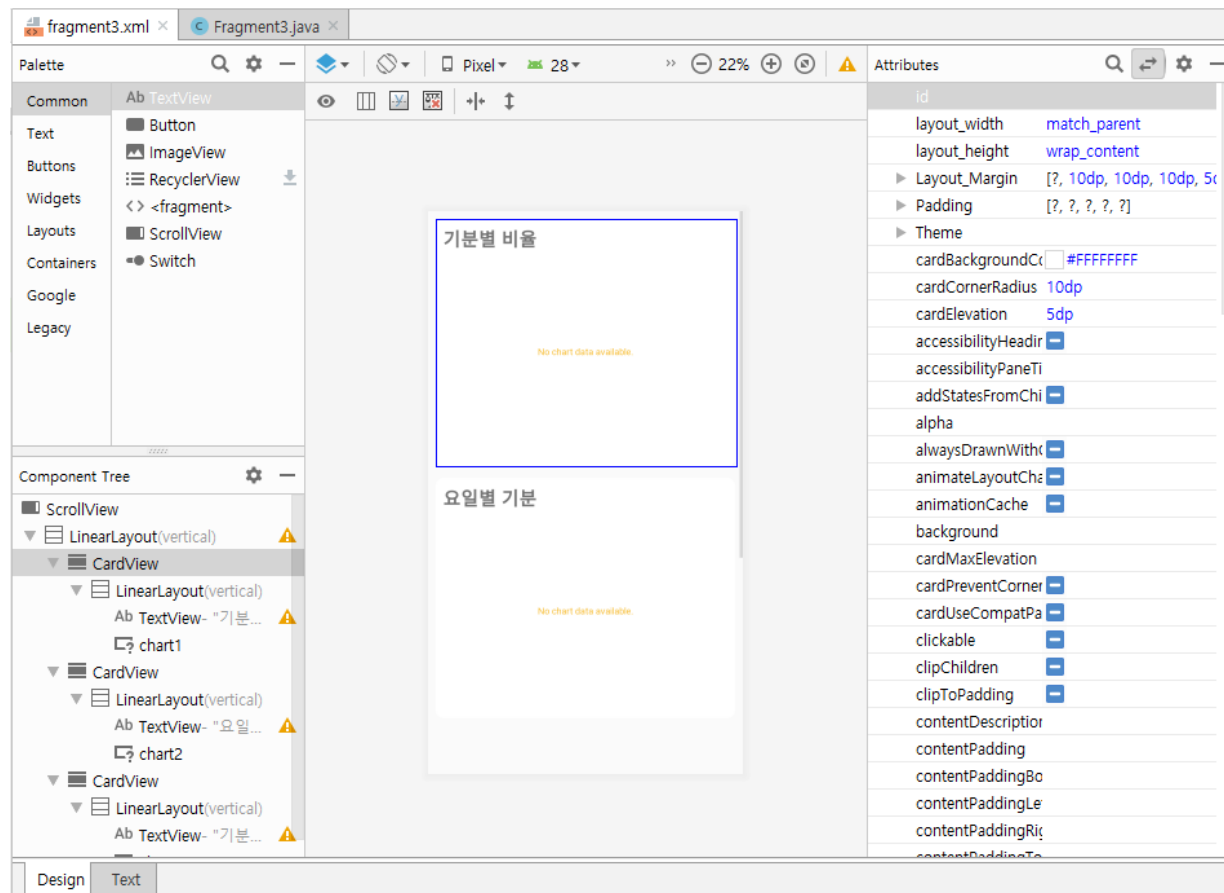
RangeSliderView sliderView = rootView.findViewById(R.id.sliderView);
sliderView.setOnSlideListener(new RangeSliderView.OnSlideListener() {
    @Override
    public void onSlide(int index) {
        Toast.makeText(context, "moodIndex changed to " + index, Toast.LENGTH_LONG).show();
    }
});

sliderView.setInitialIndex(2);
}
```



통계 화면 만들기

- 통계 화면 레이아웃 구성





통계 화면 만들기

- 그래프 사용 코드 추가

참조파일 SingleDiary>/app/java/org.techtown.diary/Fragment3.java

```
public class Fragment3 extends Fragment {
    PieChart chart;
    BarChart chart2;
    LineChart chart3;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        ViewGroup rootView = (ViewGroup) inflater.inflate(R.layout.fragment3, container, false);

        initUI(rootView);

        return rootView;
    }

    private void initUI(ViewGroup rootView) {
        chart = rootView.findViewById(R.id.chart1);
        chart.setUsePercentValues(true);
        chart.getDescription().setEnabled(false);

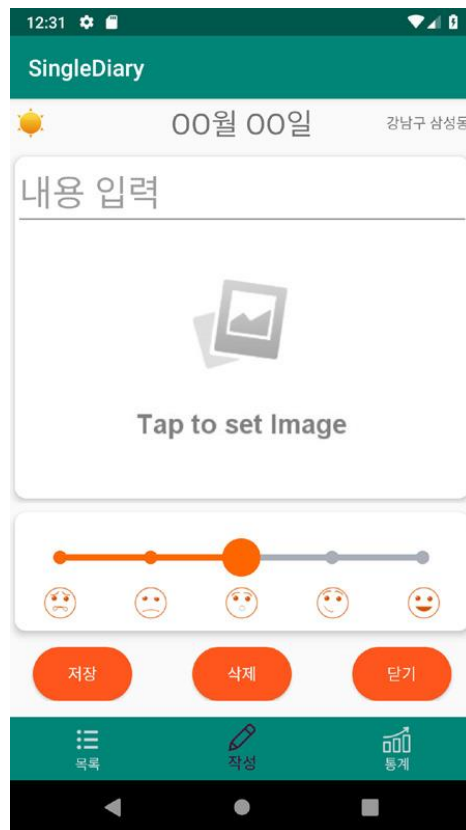
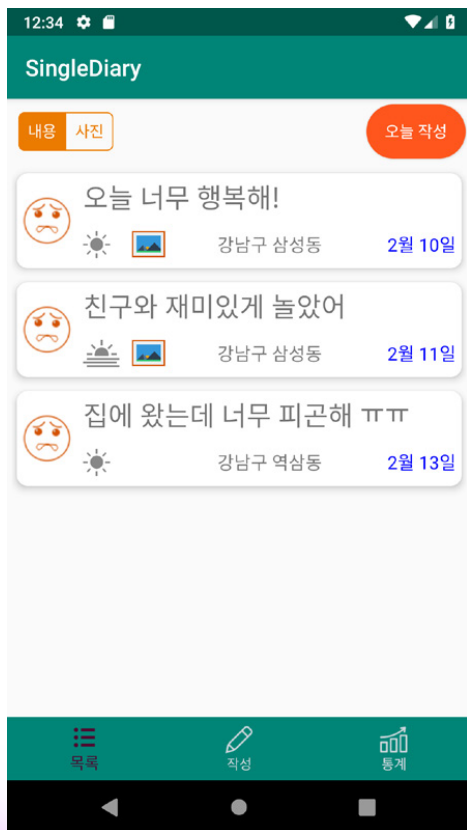
        chart.setCenterText("기분별 비율");

        chart.setTransparentCircleColor(Color.WHITE);
        chart.setTransparentCircleAlpha(110);
    }
}
```




앱 실행 후 화면 확인

- 만들어진 화면 확인



3.

내 위치와 날씨 확인하고 사진 찍기



내 위치 확인

- LocationManager를 위한 위치 확인 코드 추가

```
public void getCurrentLocation() {
    currentDate = new Date();
    currentDateString = AppConstants.dateFormat3.format(currentDate);
    if (fragment2 != null) {
        fragment2.setDateString(currentDateString);
    }

    LocationManager manager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);

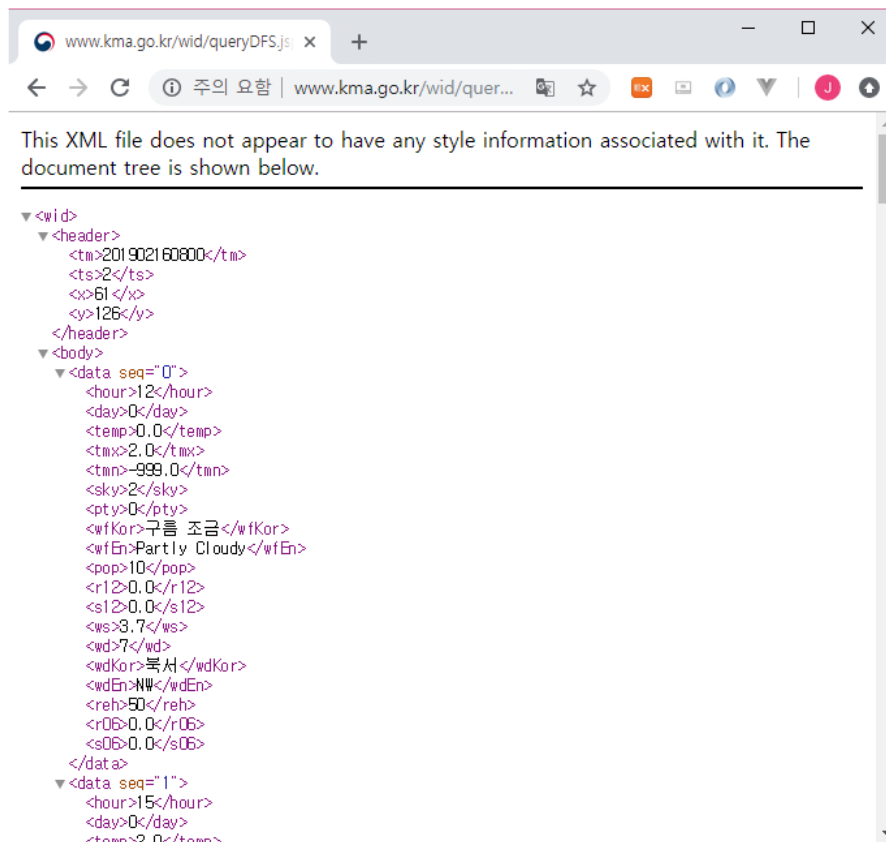
    try {
        currentLocation = manager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
        if (currentLocation != null) {
            double latitude = currentLocation.getLatitude();
            double longitude = currentLocation.getLongitude();
            String message = "Last Location -> Latitude : " + latitude + " \nLongitude:" + longitude;
            println(message);

            getCurrentWeather();
            getCurrentAddress();
        }
    }
```



날씨 확인

- 기상청 날씨 확인 사이트
- <http://www.kma.go.kr/wid/queryDFS.jsp?gridx=61 &gridy=126>





- 외부 라이브러리 추가

참조파일 SingleDiary2>/Gradle Scripts/build.gradle (Module: app)

중략...

```
repositories {  
    maven { url 'https://jitpack.io' }  
}
```

```
dependencies {
```

중략...

```
    implementation 'com.android.volley:volley:1.1.0'  
    implementation 'com.google.code.gson:gson:2.8.2'  
    implementation 'com.stanfy:gson-xml-java:0.1.7'  
    implementation 'com.github.pedroSG94:AutoPermissions:1.0.3'  
}
```



날씨 확인

- 날씨 확인 요청 코드 추가

참조파일 SingleDiary2>/app/java/org.techtown.diary/MainActivity.java

```
public class MainActivity extends AppCompatActivity
    implements OnTabItemSelectedListener, OnRequestListener,
        AutoPermissionsListener, MyApplication.OnResponseListener {
    private static final String TAG = "MainActivity";
    중략...

    public void getCurrentWeather() {

        Map<String, Double> gridMap = GridUtil.getGrid(currentLocation.getLatitude(),
                                                    currentLocation.getLongitude());

        double gridX = gridMap.get("x");
        double gridY = gridMap.get("y");
        println("x -> " + gridX + ", y -> " + gridY);

        sendLocalWeatherReq(gridX, gridY);
    }

    public void sendLocalWeatherReq(double gridX, double gridY) {
        String url = "http://www.kma.go.kr/wid/queryDFS.jsp";
        url += "?gridx=" + Math.round(gridX);
        url += "&gridy=" + Math.round(gridY);

        Map<String,String> params = new HashMap<String,String>();

        MyApplication.send(AppConstants.REQ_WEATHER_BY_GRID, Request.Method.GET, url, params, this);
    }
}
```



날씨 확인

- 날씨 확인 응답 처리 코드 추가

```
WeatherResult weather = gsonXml.fromXml(response, WeatherResult.class);

try {
    Date tmDate = AppConstants.dateFormat.parse(weather.header.tm);
    String tmDateText = AppConstants.dateFormat2.format(tmDate);
    println("기준 시간: " + tmDateText);

    for (int i = 0; i < weather.body.datas.size(); i++) {
        WeatherItem item = weather.body.datas.get(i);
        println("#" + i + " 시간: " + item.hour + "시, " + item.day + "일째");
        println(" 날씨: " + item.wfKor);
        println(" 기온: " + item.temp + " C");
        println(" 강수확률: " + item.pop + "%");

        println("debug 1 : " + (int)Math.round(item.ws * 10));
        float ws = Float.valueOf(String.valueOf((int)Math.round(item.ws * 10))) / 10.0f;
        println(" 풍속: " + ws + " m/s");
    }

    WeatherItem item = weather.body.datas.get(0);
    currentWeather = item.wfKor;
    if (fragment2 != null) {
        fragment2.setWeather(item.wfKor);
    }
}
```



- 현재 주소 확인

참조파일 SingleDiary2>/app/java/org.techtown.diary/MainActivity.java

중략...

```
public void getCurrentAddress() {
    Geocoder geocoder = new Geocoder(this, Locale.getDefault());
    List<Address> addresses = null;

    try {
        addresses = geocoder.getFromLocation(currentLocation.getLatitude(), currentLocation.getLongitude(), 1);
    } catch (Exception e) {
        e.printStackTrace();
    }

    if (addresses != null && addresses.size() > 0) {
        Address address = addresses.get(0);
        currentAddress = address.getLocality() + " " + address.getSubLocality();
        String adminArea = address.getAdminArea();
        String country = address.getCountryName();
        println("Address : " + country + " " + adminArea + " " + currentAddress);

        if (fragment2 != null) {
            fragment2.setAddress(currentAddress);
        }
    }
}
```

4.

데이터베이스 연동하기



데이터베이스 만들기

- 데이터베이스 만들어 사용하기

```
private class DatabaseHelper extends SQLiteOpenHelper {

    public DatabaseHelper(Context context) {
        super(context, AppConstants.DATABASE_NAME, null, DATABASE_VERSION);
    }

    public void onCreate(SQLiteDatabase db) {
        println("creating database [" + AppConstants.DATABASE_NAME + "].");

        println("creating table [" + TABLE_NOTE + "].");

        String DROP_SQL = "drop table if exists " + TABLE_NOTE;
        try {
            db.execSQL(DROP_SQL);
        } catch (Exception ex) {
            Log.e(TAG, "Exception in DROP_SQL", ex);
        }

        String CREATE_SQL = "create table " + TABLE_NOTE + "("
            + " _id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT, "
            + " WEATHER TEXT DEFAULT '', "
            + " ADDRESS TEXT DEFAULT '', "
            + " LOCATION_X TEXT DEFAULT '', "
```



데이터베이스 조회하기

- 리스트 화면에 보여주기

```
public int loadNoteListData() {
    AppConstants.println("loadNoteListData called.");
    String sql = "select _id, WEATHER, ADDRESS, LOCATION_X, LOCATION_Y, CONTENTS, MOOD, PICTURE,
                  CREATE_DATE, MODIFY_DATE from " + NoteDatabase.TABLE_NOTE +
                  " order by CREATE_DATE desc";

    int recordCount = -1;
    NoteDatabase database = NoteDatabase.getInstance(context);
    if (database != null) {
        Cursor outCursor = database.rawQuery(sql);

        recordCount = outCursor.getCount();
        AppConstants.println("record count : " + recordCount + "\n");

        ArrayList<Note> items = new ArrayList<Note>();

        for (int i = 0; i < recordCount; i++) {
            outCursor.moveToNext();

            int _id = outCursor.getInt(0);
            String weather = outCursor.getString(1);
            String address = outCursor.getString(2);
            String locationX = outCursor.getString(3);
```



데이터베이스 저장하기

- 입력화면에서 저장하기

```
private void saveNote() {
    String address = locationTextView.getText().toString();
    String contents = contentsInput.getText().toString();

    String picturePath = savePicture();

    String sql = "insert into " + NoteDatabase.TABLE_NOTE +
        "(WEATHER, ADDRESS, LOCATION_X, LOCATION_Y, CONTENTS, MOOD, PICTURE) values(" +
        "" + weatherIndex + ", " +
        "" + address + ", " +
        "" + "" + ", " +
        "" + "" + ", " +
        "" + contents + ", " +
        "" + moodIndex + ", " +
        "" + picturePath + ")";

    Log.d(TAG, "sql : " + sql);
    NoteDatabase database = NoteDatabase.getInstance(context);
    database.execSQL(sql);
}
```



데이터베이스 조회하기

- 통계화면에서 조회하기

```
public void loadStatData() {
    NoteDatabase database = NoteDatabase.getInstance(context);

    // 기분별 비율
    String sql = "select mood " +
        " , count(mood) " +
        "from " + NoteDatabase.TABLE_NOTE + " " +
        "where create_date >= '2019-02-01' " +
        " and create_date < '2019-03-01' " +
        "group by mood";

    Cursor cursor = database.rawQuery(sql);
    int recordCount = cursor.getCount();
    AppConstants.println("recordCount : " + recordCount);

    HashMap<String,Integer> dataHash1 = new HashMap<String,Integer>();
    for (int i = 0; i < recordCount; i++) {
        cursor.moveToNext();

        String moodName = cursor.getString(0);
        int moodCount = cursor.getInt(1);

        AppConstants.println("#" + i + " -> " + moodName + ", " + moodCount);
        dataHash1.put(moodName, moodCount);
    }
}
```

5.

마무리하기



앱 아이콘 설정 및 다국어 설정

- 앱 아이콘 추가
- 다국어 설정을 위해 파일 생성

참조파일 SingleDiary4>/app/manifests/AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="org.techtown.diary">

    중략...
    <application
        android:name=".MyApplication"
        android:usesCleartextTraffic="true"
        android:allowBackup="true"
        android:icon="@drawable/appicon"
        android:label="@string/app_name"
    중략...
```

참조파일 SingleDiary4>/app/res/values-ko/strings.xml

```
<resources>
    <string name="app_name">싱글 다이어리</string>

    <string name="tab_title1">목록</string>
    <string name="tab_title2">작성</string>
    <string name="tab_title3">통계</string>
</resources>
```



스플래시 화면 추가

- 액티비티 생성

참조파일 SingleDiary4>/app/java/org.techtown.diary/SplashActivity.java

```
public class SplashActivity extends AppCompatActivity {  
  
    Handler handler = new Handler();  
  
    @Override  
    protected void onCreate(@Nullable Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        handler.postDelayed(new Runnable() {  
            public void run() {  
                Intent intent = new Intent(getApplicationContext(), MainActivity.class);  
                startActivity(intent);  
            }  
        }, 2000);  
    }  
}
```



스플래시 화면 추가

- 스플래시 배경을 위한 XML 정의

참조파일 SingleDiary4>/app/res/drawable/splash_background.xml

```
<?xml version="1.0" encoding="utf-8"?>
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@drawable/splash_base" />
    <item android:top="210dp"
        android:gravity="center_horizontal">
        <bitmap android:src="@drawable/applogo" android:gravity="top" />
    </item>
</layer-list>
```

참조파일 SingleDiary4>/app/res/drawable/splash_base.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <gradient
        android:startColor="#FF3E50B4"
        android:centerColor="#FF7288DB"
        android:endColor="#FF7288DB"
        android:angle="90"
        android:centerY="0.5" />
    <corners android:radius="0dp" />
</shape>
```

