

Report



경성대학교

과목명 : 종합설계프로젝트

학과 : 컴퓨터공학과

학번 : 2016642045

이름 : 강건우

제출일자 : 2019. 6. 20.

목 차

1. 가계부의 개요	3
2. 가계부 앱의 필요성	3
3. 목표	3
4. 기초설계(개념설계)	4
5. 상세설계	6
6. 구현	14
7. 테스트	32
8. 수정 및 보완	34
9. 프로젝트 결과	37
10. 결론	47
11. 부록	
11-1. 전체 소스코드	48
11-2. 사용설명서	90
11-3. 프로젝트 설치 환경	98
11-4. 프로젝트 결과 시연 환경	99
11-5. 참고 자료	100

1. 가계부의 개요

- 가계부의 사전적 의미는 가계 수입과 가계 지출을 중심으로 재산의 증가와 감소를 일정한 형식에 맞추어서 회기 동안 기록하는 장부이다.

2. 가계부 앱의 필요성

- 돈을 관리하는 것은 인생에 있어 가장 중요한 일입니다. 평소에 네이버 가계부를 사용하고 있지만 사용자로서 사용하기에 불편한 점이 조금 있습니다. 그렇기 때문에 저희 2조는 가계부를 보기 쉽게 그래프나 차트로 표현을 하여 보기 쉽게 만들기 위해 가계부의 필요성을 느꼈습니다. 가계부의 필요성으로 첫 번째 지출관리가 있습니다. 지출관리를 함으로써 돈이 어디서 어떤 이유로 돈이 빠져나가는지 본인이 명확하게 알고 있으면 그에 대한 피드백이 가능하고 불필요하게 세어나가는 돈을 손에 짚 수 있을 것입니다. 두 번째로 자산관리입니다. 소득과 소비가 일정하지 않기 때문에 지속 가능한 경제생활을 이루기 위해 자산에 대한 계획과 관리가 필요합니다. 더 나아가 예산관리까지 할 수 있기에 가계부를 쓰는 것은 매우 중요합니다.

3. 가계부 앱의 목표

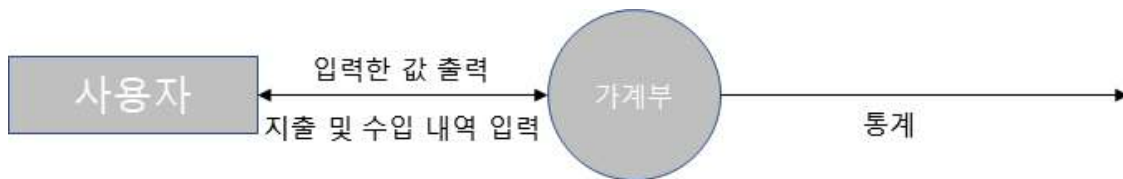
- 가계부 앱을 사용함으로써 사용자가 지출 관리, 자산 관리, 예산 관리 등을 쉽게 할 수 있게 만드는 것이 목표입니다. 저희 2조가 만든 가계부로 사용자 지출관리를 보다 편리하고 간단하게 표현을 할 수 있기에 기대효과가 클 것으로 보입니다.

4. 기초설계(개념설계)

4-1. 요구사항에 대한 설명

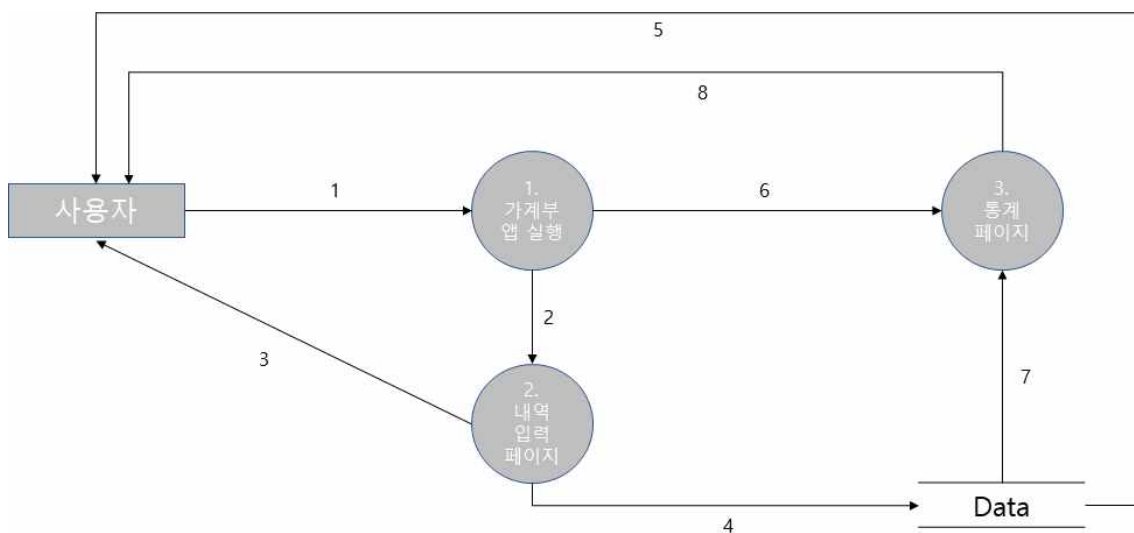
- 사용자가 지출 내역을 입력하면 입력한 값이 출력되게 한다.
- 사용자가 수입 내역을 입력하면 입력한 값이 출력되게 한다.
- 통계 보기 버튼을 누르면 지출과 수입 내역을 한 번에 확인되게 한다.

4-2. 요구분석_배경도



- 사용자가 가계부 앱을 실행하면 내역 입력을 하고 가계부 앱은 입력한 값을 사용자에게 출력을 해준다.
- 추가적으로 가계부 앱은 입력받은 금액들을 누적합계를 낸다.

4-3. 요구분석_자료흐름도



– 자료흐름도에 대한 설명

- 1) 사용자가 가계부 앱을 실행한다.
- 2) 내역 입력 페이지가 출력이 된다.
- 3) 사용자가 입력 페이지에서 입력을 한다.
- 4) 입력 페이지에서 입력한 데이터가 데이터베이스에 저장된다.
- 5) 입력한 값이 사용자에게 리스트로 출력된다.
- 6) 통계 버튼을 누르면 통계 페이지가 출력이 된다.
- 7) 통계 페이지가 출력되는 동시에 DB에서 입력한 데이터를 가져온다.
- 8) 가져온 데이터를 사용자에게 출력해준다.

4-4. 요구분석_소단위 명세서(Mini-Spec)

가계부 처리

1. FOR 내역 작성에서 지출, 수입별로 기록한다.
 - 1.1 IF 지출 내역 입력 THEN 내용, 금액, 사용처, 날짜를 기록한다.
 - 1.2 IF 수입 내역 입력 THEN 내용, 금액, 날짜를 기록한다.
 - 1.3 작성 완료된 내역서를 저장해서 처리한다.
2. 만일 위의 처리과정에서 잘못 작성한 경우
 - 2.1 오류 메시지를 출력한다.
3. 작성 완료된 내역서를 한번에 볼 수 있게 통계 보기를 출력한다.

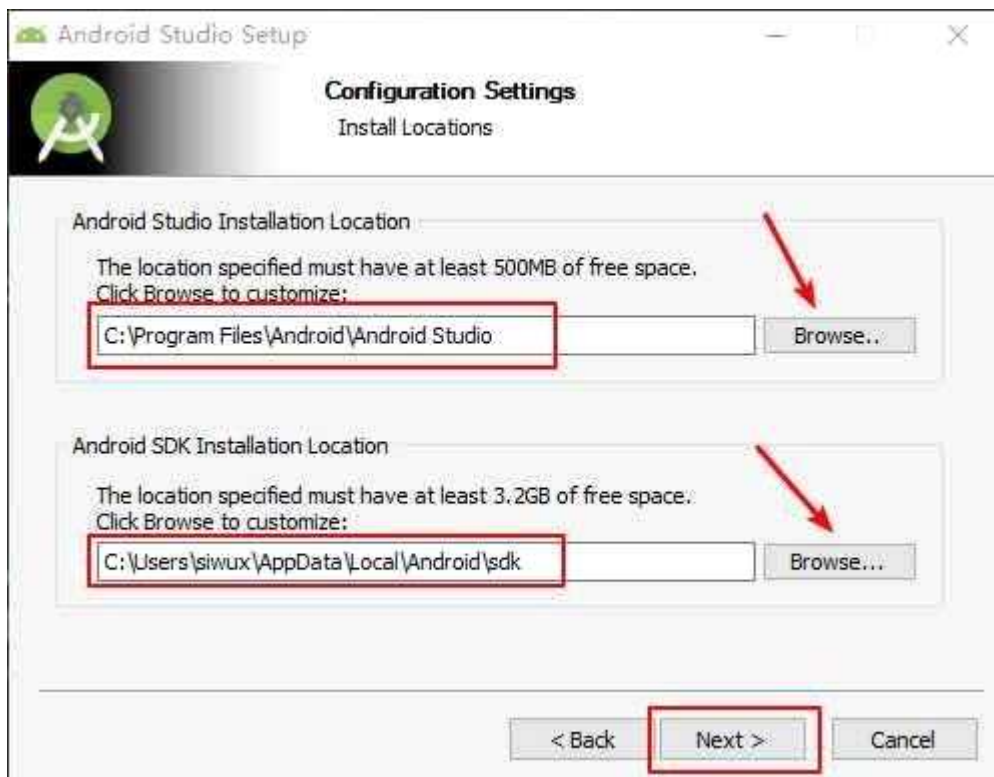
5. 상세설계

5-1. 안드로이드 스튜디오 설치과정

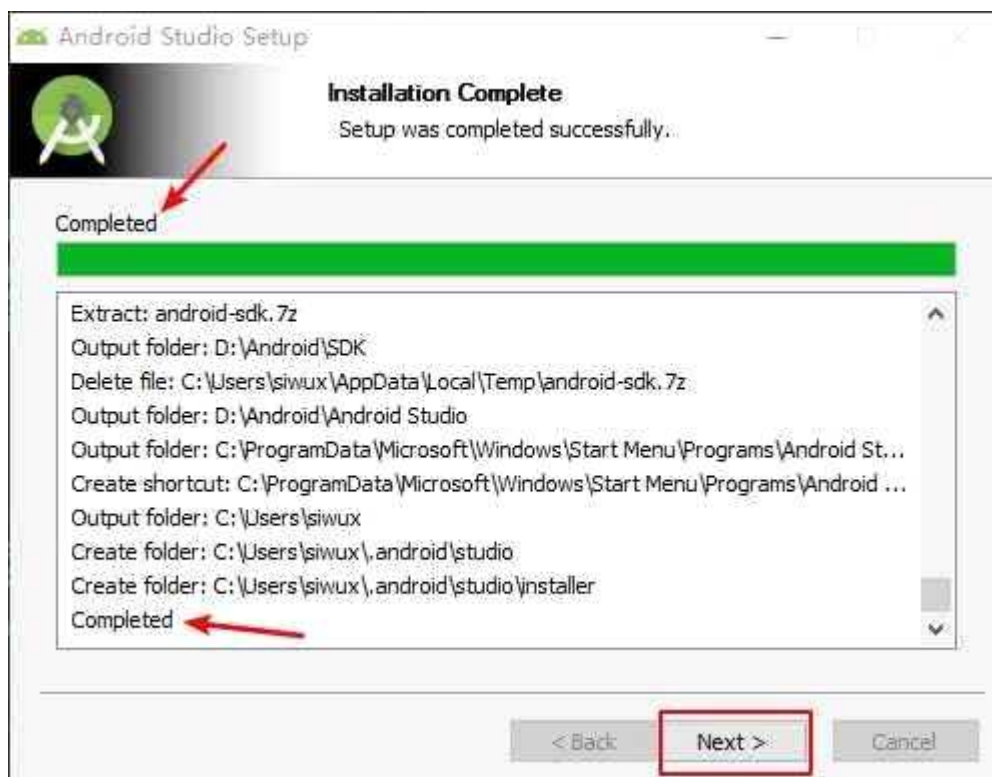
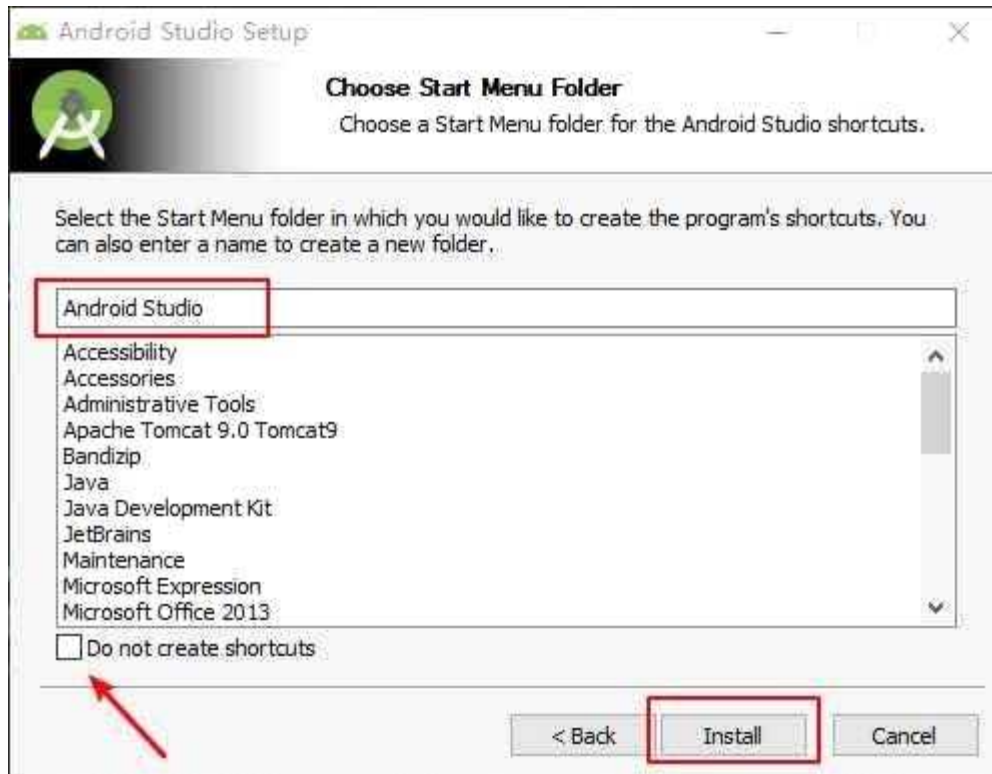
- 1) 안드로이드 공식 홈페이지에서 다운로드
 - <https://developer.android.com/studio/>
- 2) 다운로드 후 실행



3) 설치할 경로 지정



4) 설치 완료

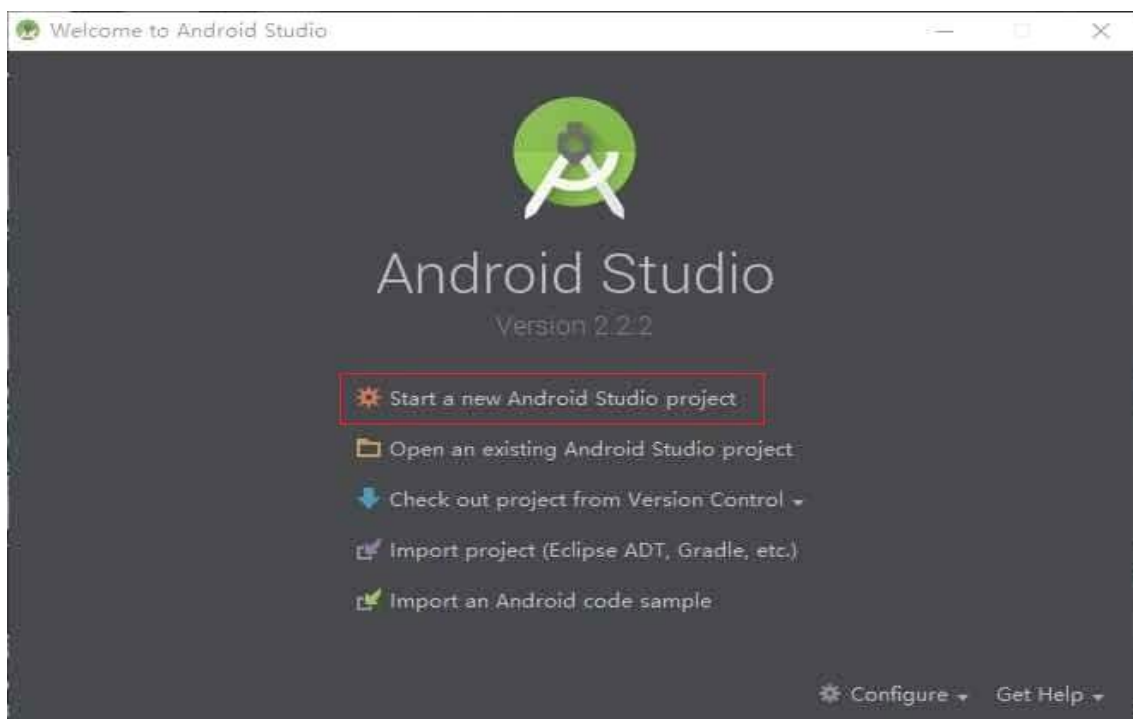


5) 설치 완료 후 안드로이드 스튜디오 실행

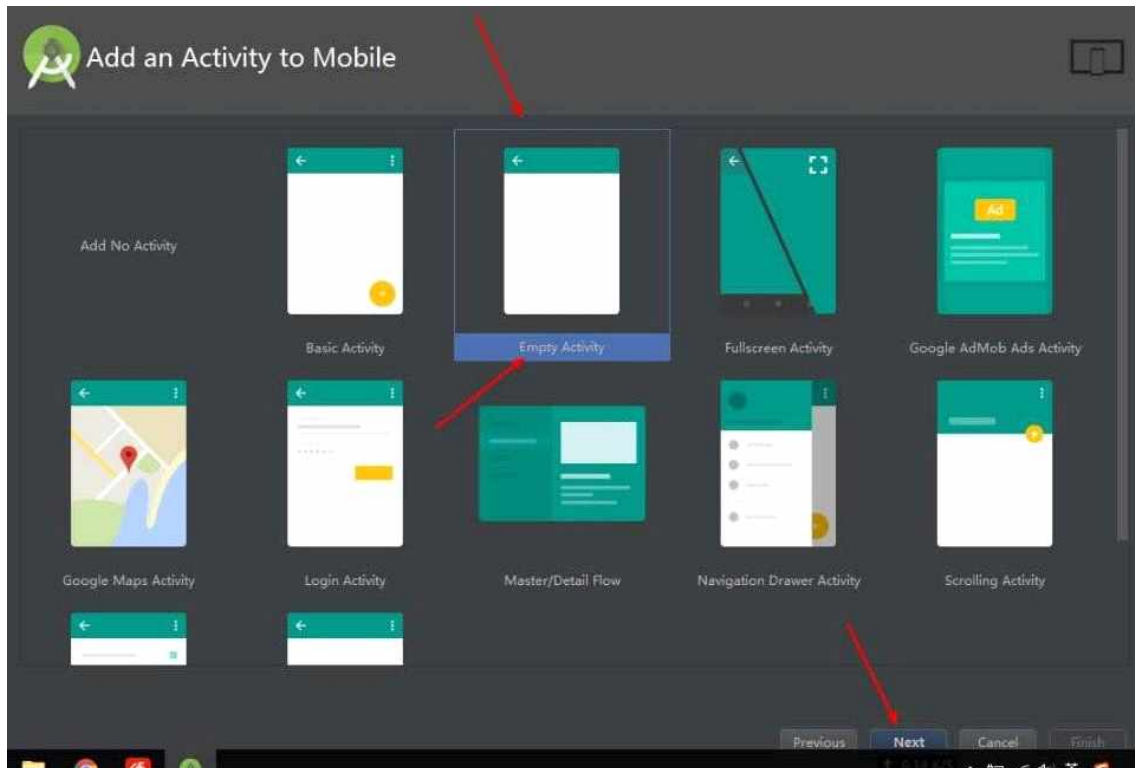


5-2. 새 프로젝트 생성

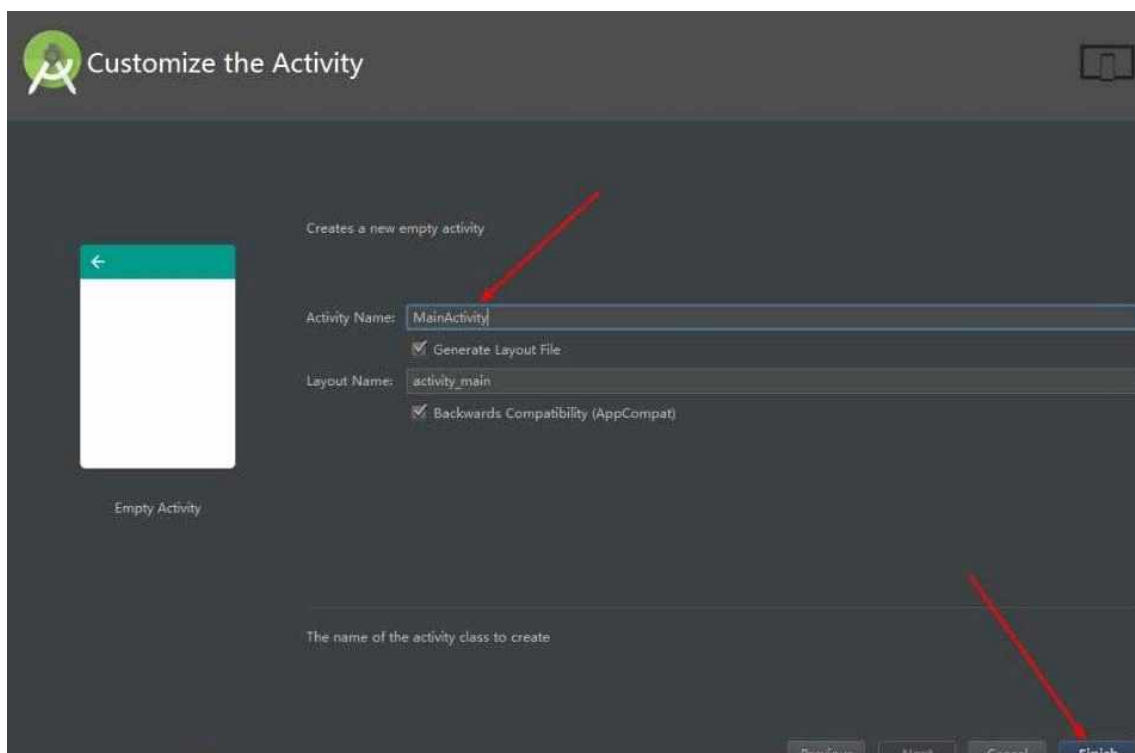
1) Start a new Android Studio project 선택



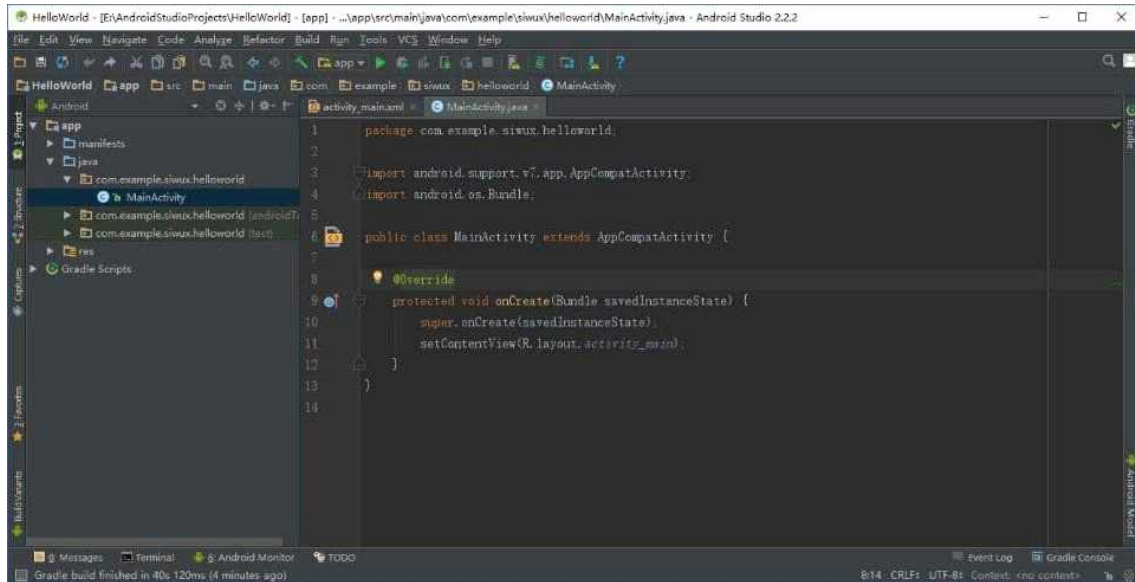
2) Empty Activity 선택 후 Next



3) Empty Activity의 MainActivity 이름 지정 후 Finish



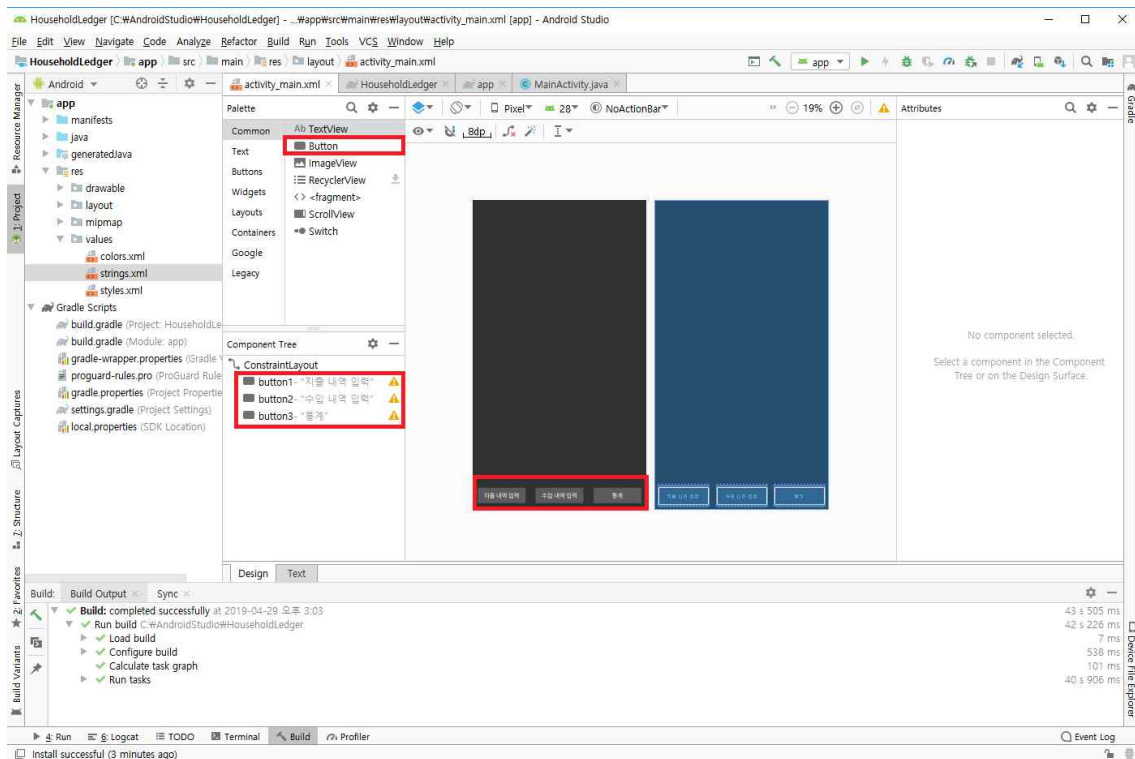
4) 새 프로젝트 Empty Activity가 생성된 것을 확인 할 수 있다.



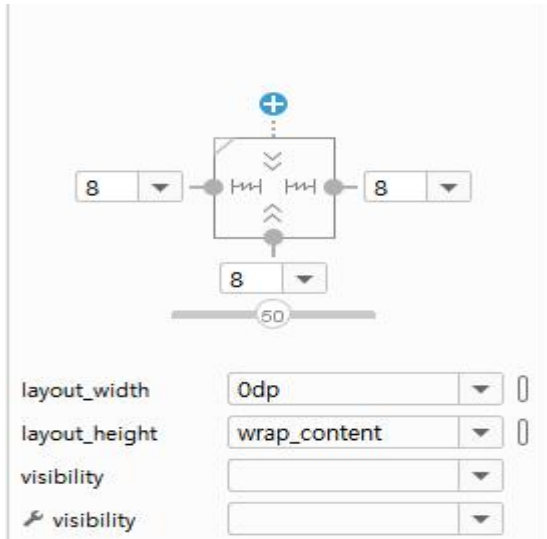
5-3. 메인 화면에서 사용할 틀 설계

1) 요구분석대로 적용하기 위해 버튼 세 개를 생성

- 지출 내역 입력 버튼, 수입 내역 입력 버튼, 통계 버튼



2) 버튼의 넓이와 높이 설정 후 버튼이 움직이지 않게 고정시킨다.



3) activity_main.xml파일 클릭 후 좌측 하단에 있는 Text를 선택

- onClick(버튼 이벤트)의 이름을 지정 후 생성

```
activity_main.xml x HouseholdLedger x app x MainActivity.java x
14 <Button
15     android:layout_marginEnd="8dp"
16     android:layout_marginBottom="8dp"
17     android:text="지출 내역 입력"
18     android:onClick="onButtonExpense"
19     app:layout_constraintBottom_toBottomOf="parent"
20     app:layout_constraintEnd_toStartOf="@+id/button2"
21     app:layout_constraintStart_toStartOf="parent" />
22
23 <Button
24     android:id="@+id/button2"
25     android:layout_width="0dp"
26     android:layout_height="wrap_content"
27     android:layout_marginStart="8dp"
28     android:layout_marginEnd="8dp"
29     android:layout_marginBottom="8dp"
30     android:text="수입 내역 입력"
31     android:onClick="onButtonImport"
32     app:layout_constraintBottom_toBottomOf="parent"
33     app:layout_constraintEnd_toStartOf="@+id/button3"
34     app:layout_constraintStart_toEndOf="@+id/button1" />
35
36 <Button
37     android:id="@+id/button3"
38     android:layout_width="0dp"
39     android:layout_height="wrap_content"
40     android:layout_marginStart="8dp"
41     android:layout_marginEnd="8dp"
42     android:layout_marginBottom="8dp"
43     android:text="통계 보기"
44     android:onClick="onButtonStats"
45     app:layout_constraintBottom_toBottomOf="parent"
```

4) onClick(버튼 이벤트)의 함수가 생성된 것을 확인할 수 있다.

- MainActivity.java에서 확인 가능

```
activity_main.xml x HouseholdLedger x app x MainActivity.java >
1 package com.example.householdledger;
2
3 import ...
6
7 public class MainActivity extends AppCompatActivity {
8
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.activity_main);
13    }
14
15    public void onButtonExpense(View view) {
16    }
17
18    public void onButtonImport(View view) {
19    }
20
21    public void onButtonStats(View view) {
22    }
23 }
24
```

5) Run 버튼을 눌러 컴파일 실행

- 안드로이드 가상 에뮬레이터가 부팅 후 앱 실행이 되는 것을 확인할 수 있다.



6. 구현

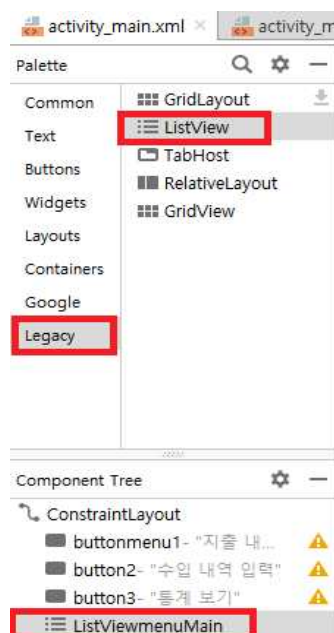
6-1. 주요 코드 구현 및 과정

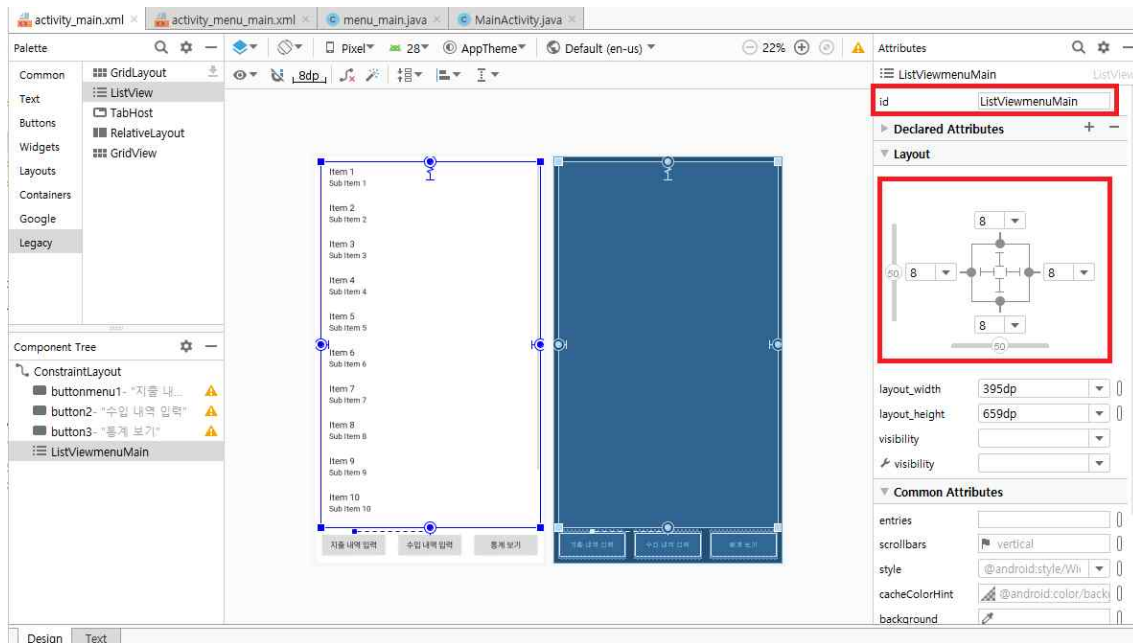
1) 메인 화면에서 내역 추가, 내역 수정할 때 값을 갱신해주는 코드



2) activity_main.xml에서 ListView를 추가

- ListView를 사용함으로써 입력한 값이 List에 담겨서 출력이 된다.





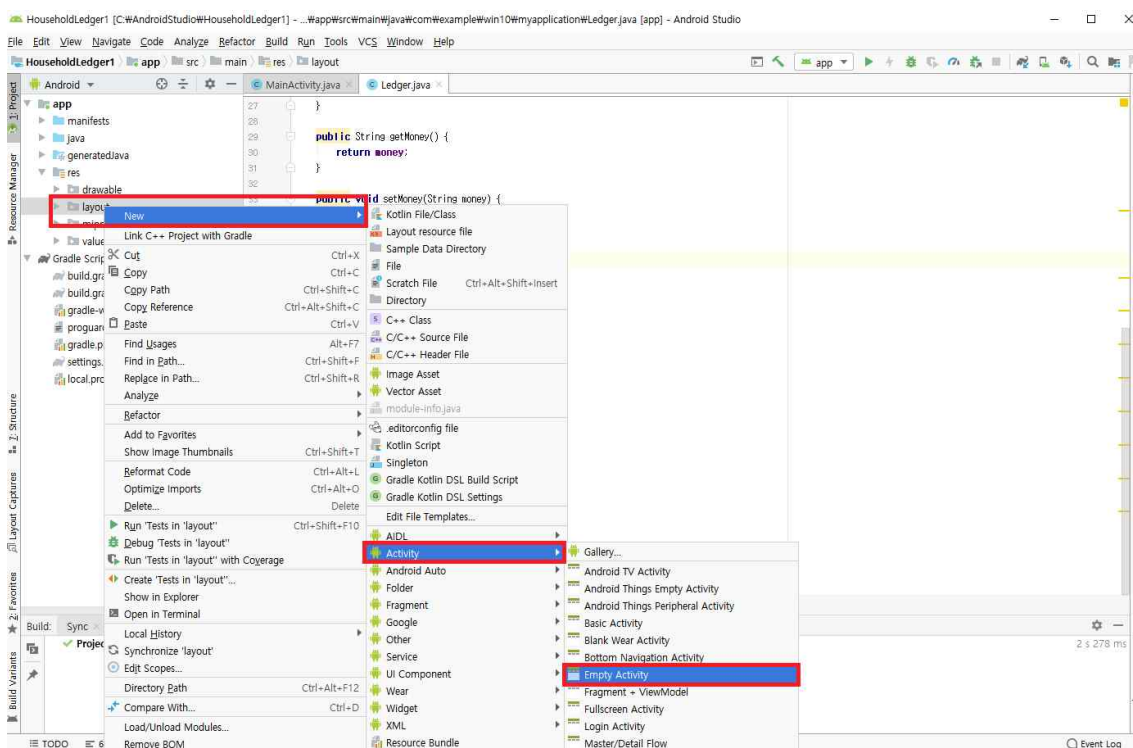
- 3) Ledger 클래스에서는 implements serializable을 사용하여 직렬화 시켜줌. 직렬화 시켜주는 이유는 전송 가능한 형태로 만들기 위함(파일, 버퍼 등에 저장하거나 네트워크로 전달할 수 있도록 하기 위함) 그리고 생성자와 접근자, 설정자 함수를 구현하는 코드를 구현



```
MainActivity.java x Ledger.java x
27 }
28
29 public String getMoney(){
30     return money;
31 }
32
33 public void setMoney(String money){
34     this.money = money;
35 }
36
37 public String getUsage(){
38     return usage;
39 }
40
41 public void setUsage(String usage){
42     this.usage = usage;
43 }
44
45 public String getDate(){
46     return date;
47 }
48
49 public void setDate(String date){
50     this.date = date;
51 }
52
53 }
```

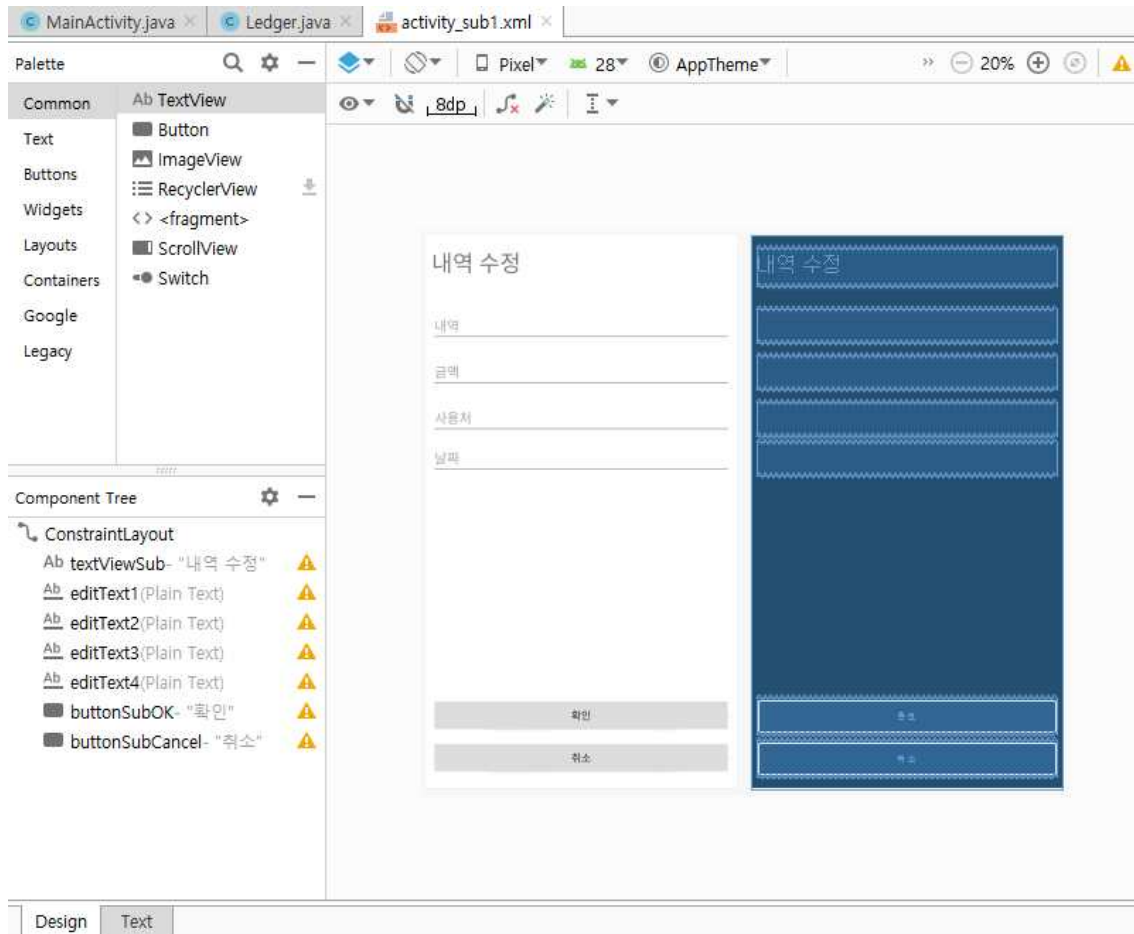
4) Sub1Activity 생성

- 지정 경로 우클릭 -> New -> Activity -> Empty Activity



5) activity_sub1.xml(Sub1Activity 생성 후 나온 xml파일)에서 틀 생성

- textView 1개, editText 4개, button 2개



- 6) Sub1Activity 클래스에서는 activity_sub1.xml을 가져오기 위해 activity_sub1.xml의 ID를 읽어서 가져오는 코드와 내역 추가 시 입력되는 값을 저장하여 넘겨주는 코드
- * 버튼 이벤트 구현할 때 중요한 점 : xml의 버튼 ID와 버튼 클래스 명을 일치시켜주는 것

```
1 package com.example.win10.myapplication;
2
3 import ...
4
5
6
7
8
9
10 public class Sub1Activity extends AppCompatActivity {
11
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_sub1);
16         TextView tv = findViewById(R.id.textViewSub);
17         tv.setText("내역 추가");
18     }
19
20     public void onButtonOK(View view) {
21         EditText et1 = findViewById(R.id.editText1);
22         String title = et1.getText().toString();
23
24         EditText et2 = findViewById(R.id.editText2);
25         String money = et2.getText().toString();
26
27         EditText et3 = findViewById(R.id.editText3);
28         String usage = et3.getText().toString();
29
30         EditText et4 = findViewById(R.id.editText4);
31         String date = et4.getText().toString();
32
33         Intent data = new Intent();
34         data.putExtra(name: "title", title);
35         data.putExtra(name: "money", money);
36         data.putExtra(name: "usage", usage);
37         data.putExtra(name: "date", date);
38
39         setResult(RESULT_OK, data);
40         finish();
41     }
42
43     public void onButtonCancel(View view) { finish(); }
44
45
46
47
```

7) LedgerAdapter 클래스에서는 안드로이드에서 제공하는
BaseAdapter를 사용하는 코드

```
package com.example.win10.myapplication;

import ...

class LedgerAdapter extends BaseAdapter {
    ArrayList<Ledger> items;
    Context context;
    int itemlayout;

    public LedgerAdapter(Context context, int layout_item_sc, ArrayList<Ledger> items) {
        this.items = items;
        this.itemlayout = layout_item_sc;
        this.context = context;
    }

    @Override
    public int getCount() { return items.size(); }

    @Override
    public Object getItem(int position) { return items.get(position); }

    @Override
    //item과2간의 고유한 값을 넘겨준다.
    public long getItemId(int position) { return position; }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        //convertView : 리스트를 담는 틀을 만드는 것
        if (convertView == null) {
            LayoutInflater inflater = (LayoutInflater)context.getSystemService(context.LAYOUT_INFLATER_SERVICE);
            convertView = inflater.inflate(itemlayout, root: null);
        }

        TextView tv1 = convertView.findViewById(R.id.textViewitem1);
        tv1.setText(items.get(position).getTitle());

        TextView tv2 = convertView.findViewById(R.id.textViewitem2);
        tv2.setText(items.get(position).getMoney());

        TextView tv3 = convertView.findViewById(R.id.textViewitem3);
        tv3.setText(items.get(position).getUsage());

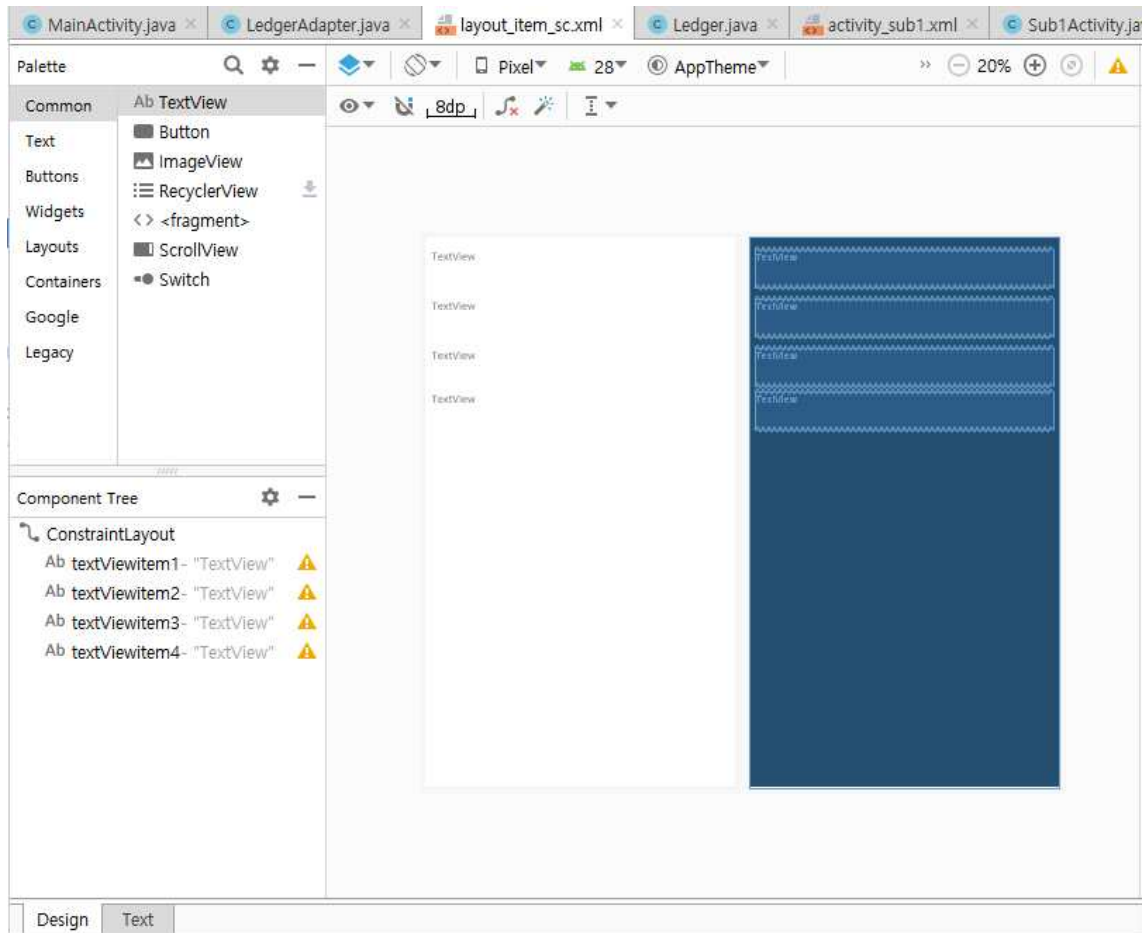
        TextView tv4 = convertView.findViewById(R.id.textViewitem4);
        tv4.setText(items.get(position).getDate());

        return convertView;
    }
}
```

- 8) LedgerAdapter 클래스에서 정의한 ArrayList와 어댑터를 MainActivity 클래스에서 호출하여 값을 저장하는 코드와 한 번 클릭 시 내역 수정 페이지로 넘어가는 코드, 길게 롱 클릭 시 내역이 삭제되는 코드를 구현

```
1 package com.example.win10.myapplication;
2
3 import ...
4
5
6
7
8
9
10
11
12
13 public class MainActivity extends AppCompatActivity {
14     private static final int CALL_FOR_UPDATE = 0;
15     private static final int CALL_FOR_CREATE = 1;
16     ArrayList<Ledger> items;
17     LedgerAdapter adapter;
18
19     @Override
20     protected void onCreate(Bundle savedInstanceState) {
21         super.onCreate(savedInstanceState);
22         setContentView(R.layout.activity_main);
23
24         //1. 집합데이터를 생성하기 위한 공간 확보 : ArrayList
25         items = new ArrayList<>();
26
27
28         //2. 객체를 저장하기 위한 Adapter 생성
29         //ArrayAdapter : 문자열만 저장함, ArrayAdapter는 사용하지 못한다.
30         adapter = new LedgerAdapter( context: this, R.layout.layout_item_sc, items);
31
32         //3. AdapterView(ListView)에 Adapter 등록
33         ListView lv1 = findViewById(R.id.ListViewMain);
34         lv1.setAdapter(adapter);
35
36         lv1.setOnItemClickListener((parent, view, position, id) -> {
37             Intent intent = new Intent( packageContext: MainActivity.this, Sub2Activity.class);
38             intent.putExtra( name: "index", position);
39             intent.putExtra( name: "ledger", items.get(position));
40             startActivityForResult(intent, CALL_FOR_UPDATE);
41
42         });
43
44         lv1.setOnItemLongClickListener((parent, view, position, id) -> {
45             items.remove(position); //현재 위치에서 삭제한다.
46             adapter.notifyDataSetChanged(); //변경된 데이터를 공지해준다.
47             return false;
48         });
49
50
51
52
53
54
55 public void onButtonMainClick(View view) {
56     Intent intent = new Intent( packageContext: this, Sub1Activity.class);
57     startActivityForResult(intent, CALL_FOR_CREATE);
58
59 }
```

9) ListView에 담길 Item을 textView로 생성 후 정렬



10) 내역 수정 클래스 이벤트 구현

- 내역 수정 시 사용할 틀(레이아웃)을 activity_sub1에서 가져온다.
- Intent를 사용하여 데이터를 넘긴다.
 - 자료형이 정수형이기 때문에 getIntExtra로 사용한다.
- 어떤 자료형인지 모르기 때문에 Ledger로 강제형변환을 한다.
 - 직렬화 되어서 돌아오는 객체는 순서대로 풀어준다.

```
package com.example.win10.myapplication;

import ...

public class Sub2Activity extends AppCompatActivity {
    int index;
    Ledger ledger;
    EditText et1, et2, et3, et4;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sub1);
        TextView tv = findViewById(R.id.textViewSub);
        tv.setText("내역 수정");

        Intent intent = getIntent();
        index = intent.getIntExtra("index", -1); //자료형이 정수형이기 때문에 getIntExtra("이름표", 초기값)로 사용한다.

        if (index != -1) {
            ledger = (Ledger)intent.getSerializableExtra("ledger"); //어떤 자료형인지 모르기 때문에 Ledger로 강제형변환 직렬화 되어서 돌아오는 객체는 순서대로
        }

        et1 = findViewById(R.id.editText1);
        et1.setText(ledger.getTitle());

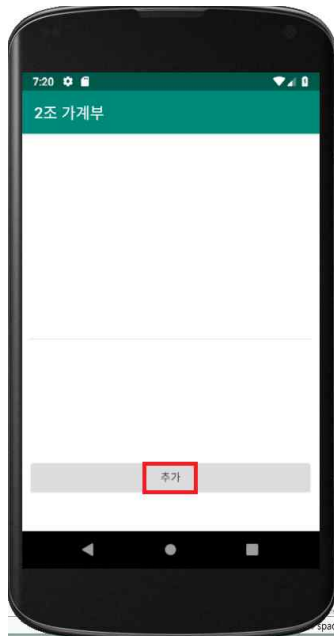
        et2 = findViewById(R.id.editText2);
        et2.setText(ledger.getMoney());

        et3 = findViewById(R.id.editText3);
        et3.setText(ledger.getUsage());

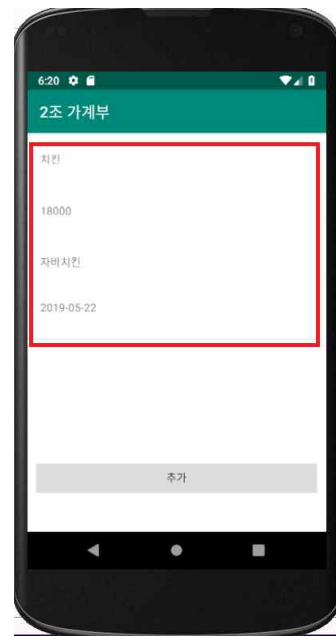
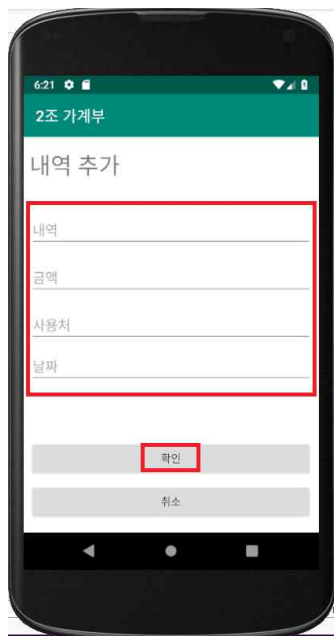
        et4 = findViewById(R.id.editText4);
        et4.setText(ledger.getDate());
    }
}
```

11) 컴파일 실행을 하여 확인

- 추가 버튼을 클릭



- 내역 추가 페이지로 넘어가게 되면 입력을 하고 확인 버튼을 클릭
- 확인 버튼을 클릭하면 메인 화면으로 돌아가 입력한 내용을 확인할 수 있다.

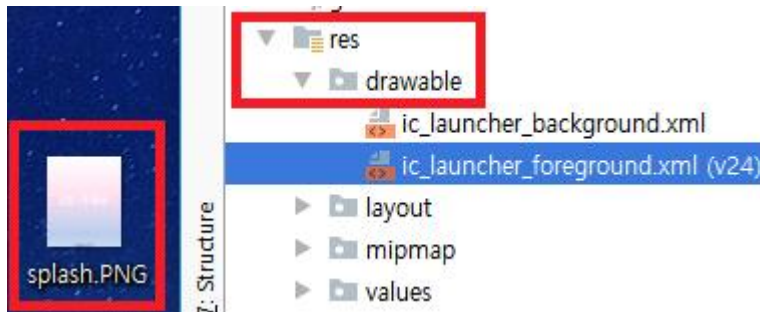


6-2. 스플래쉬(Splash) 화면 구현

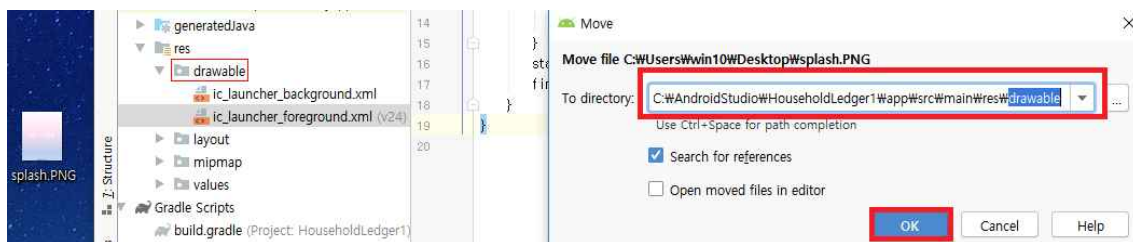
- 1) 앱 시작 로딩 화면에 사용될 이미지 생성
 - 본 이미지는 ppt로 제작



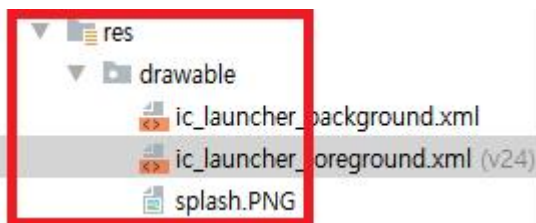
2) 바탕화면에 생성된 이미지를 지정 경로에 드래그



3) 지정 경로를 확인 후 OK



4) 지정 경로에 이미지가 생성된 것을 확인할 수 있다.

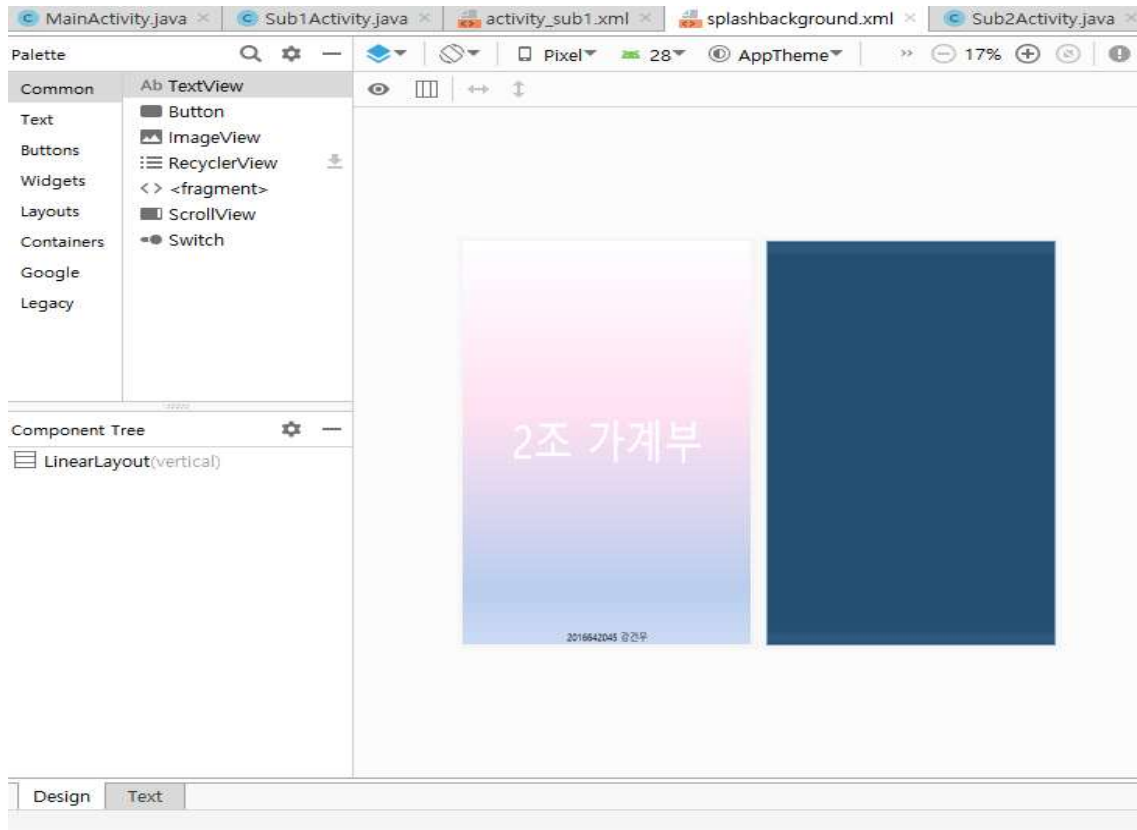


5) splashbackground.xml을 생성하여 사용할 이미지를 불러온다.

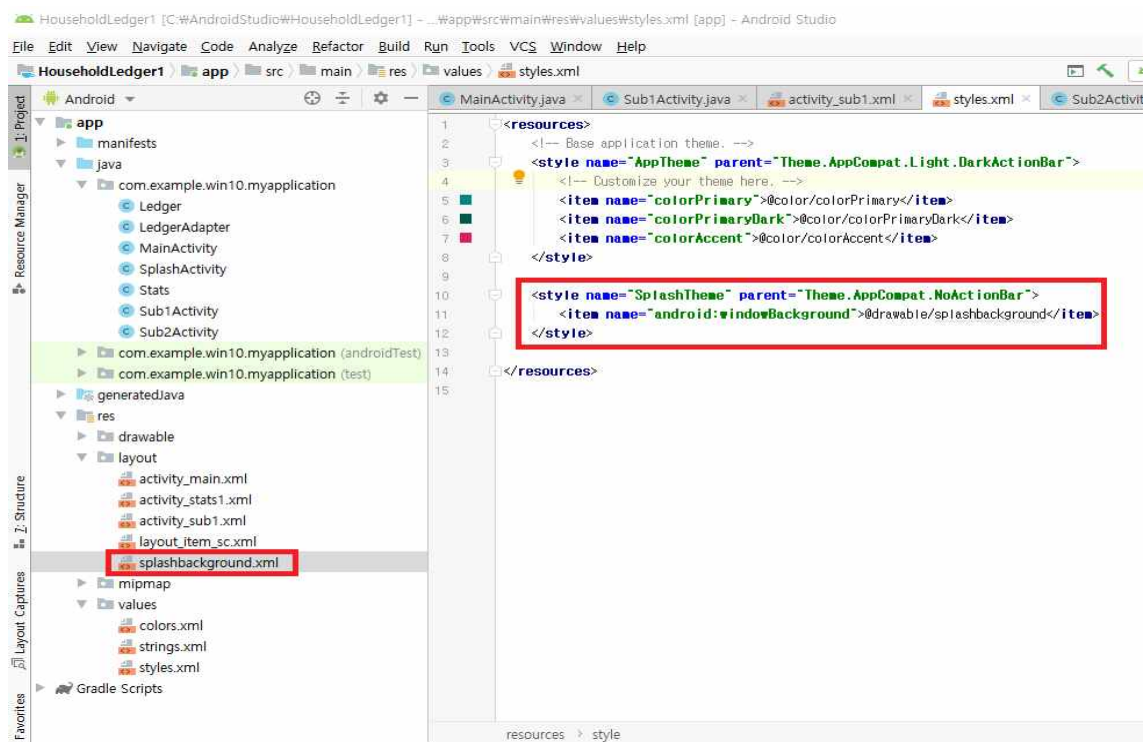
- 수직(세로), 넓이, 높이를 설정 후 불러올 이미지 경로 설정



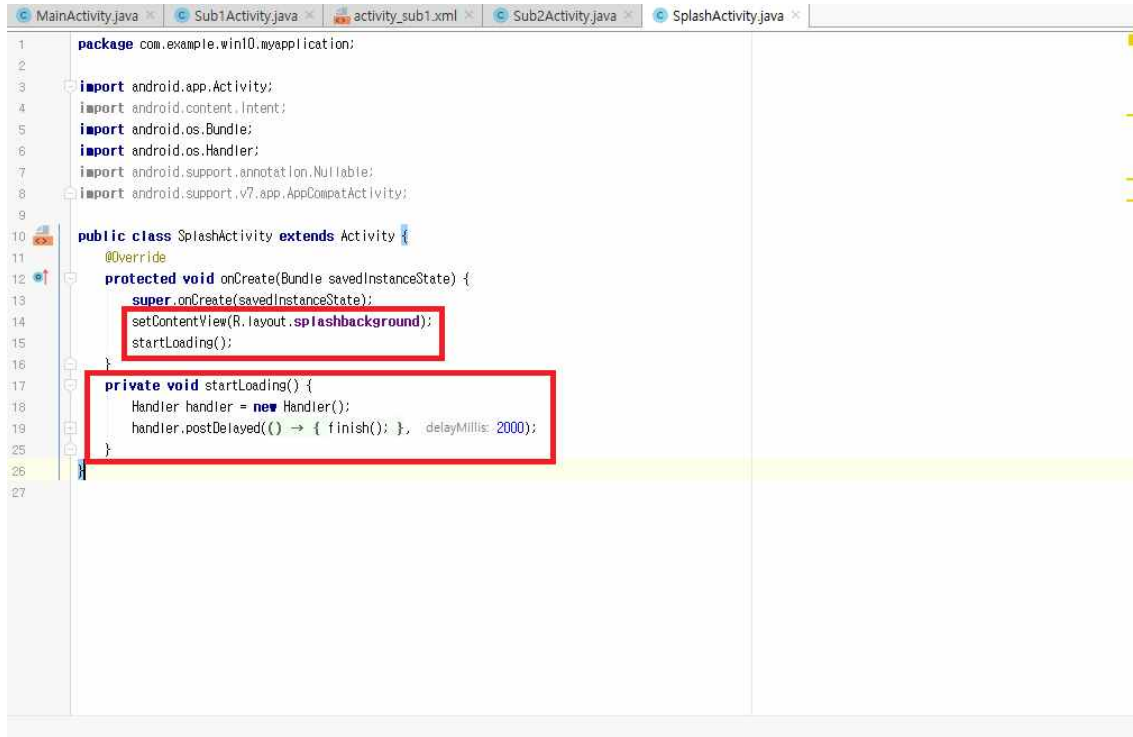
6) splashbackground.xml에서 Design탭을 눌러 확인



7) style.xml에서 splashbackground.xml를 불러온다.

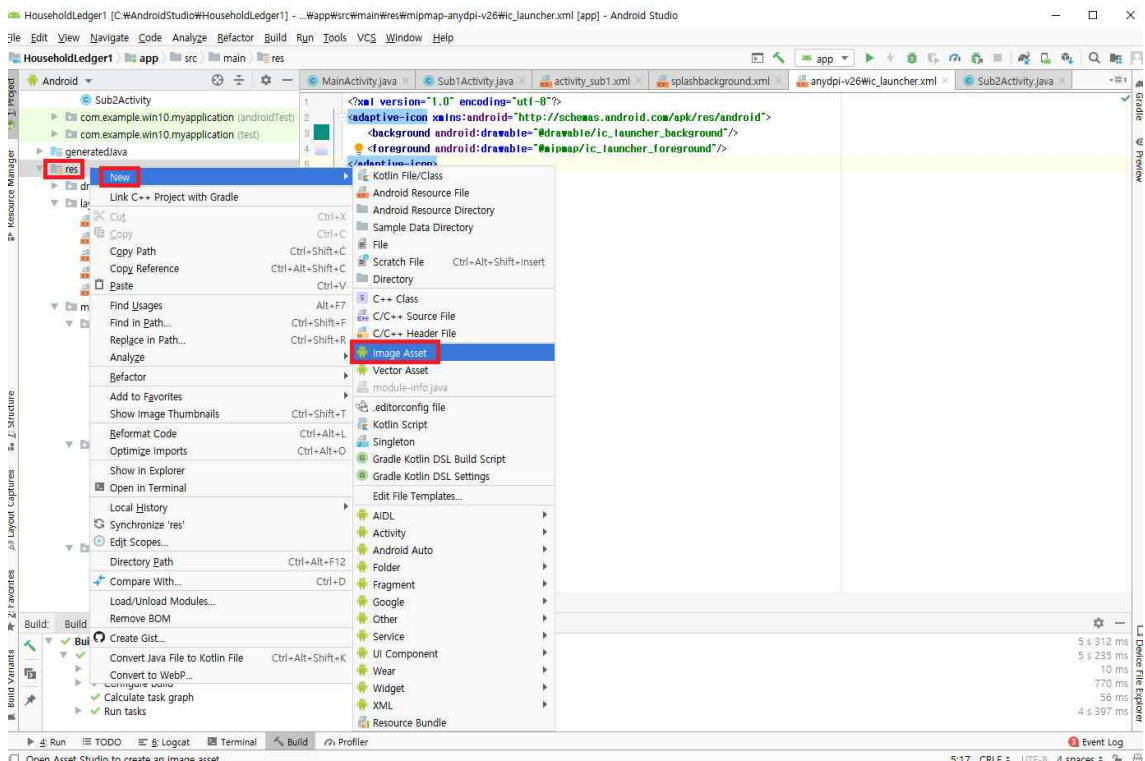


8) SplashActivity.java에서 로딩 시간 설정

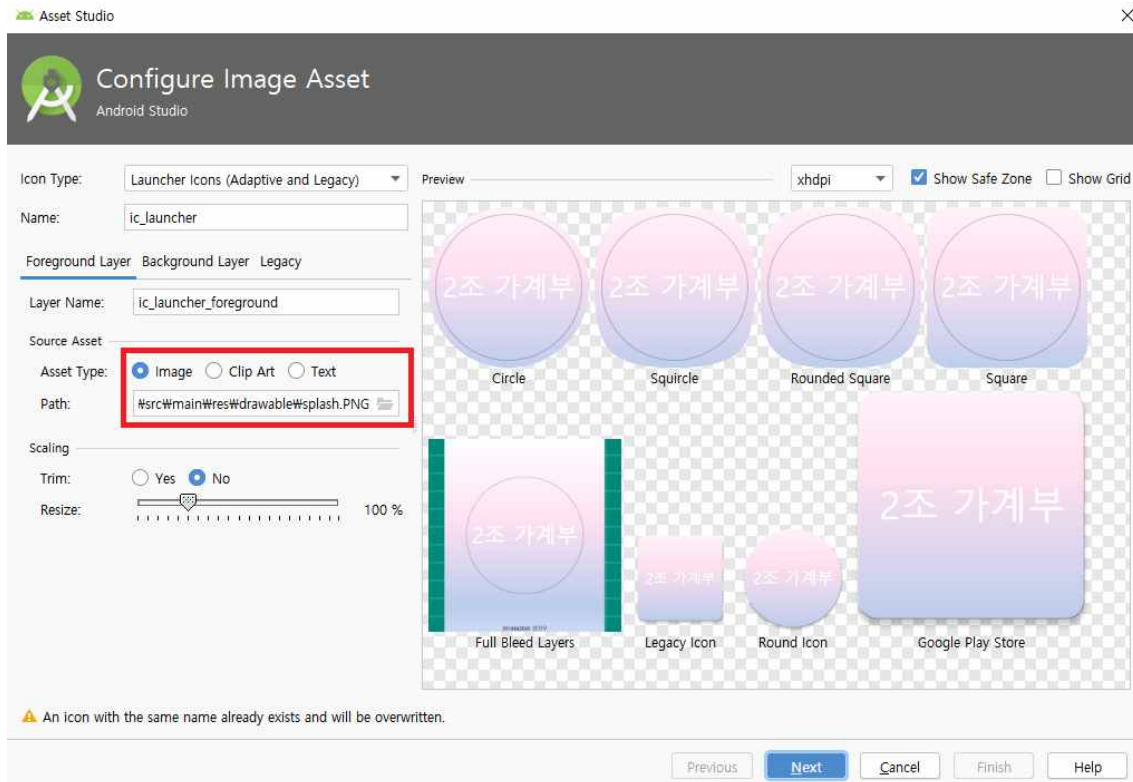


6-3. 앱 아이콘 디자인 구현

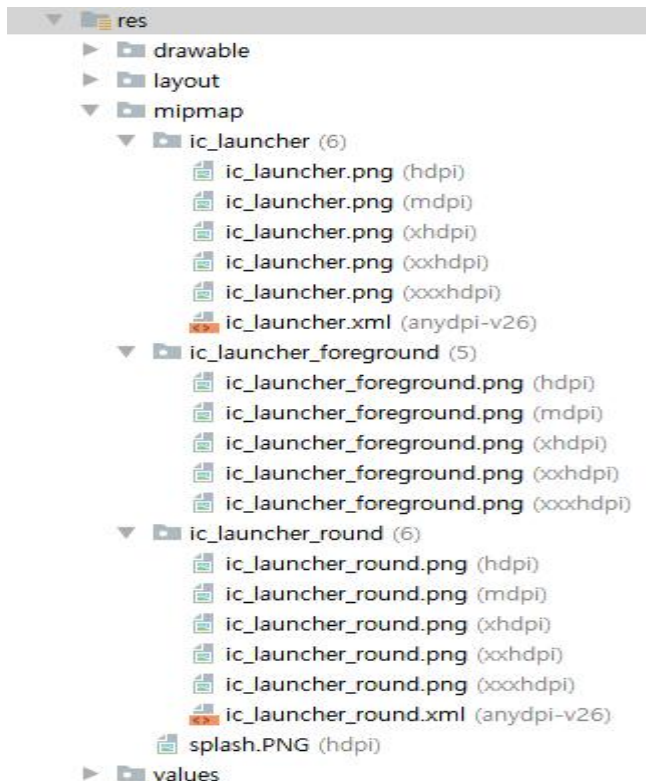
1) res 우클릭 -> New -> image Asset



2) 앱 아이콘에 사용할 이미지를 지정 경로로 불러온다.

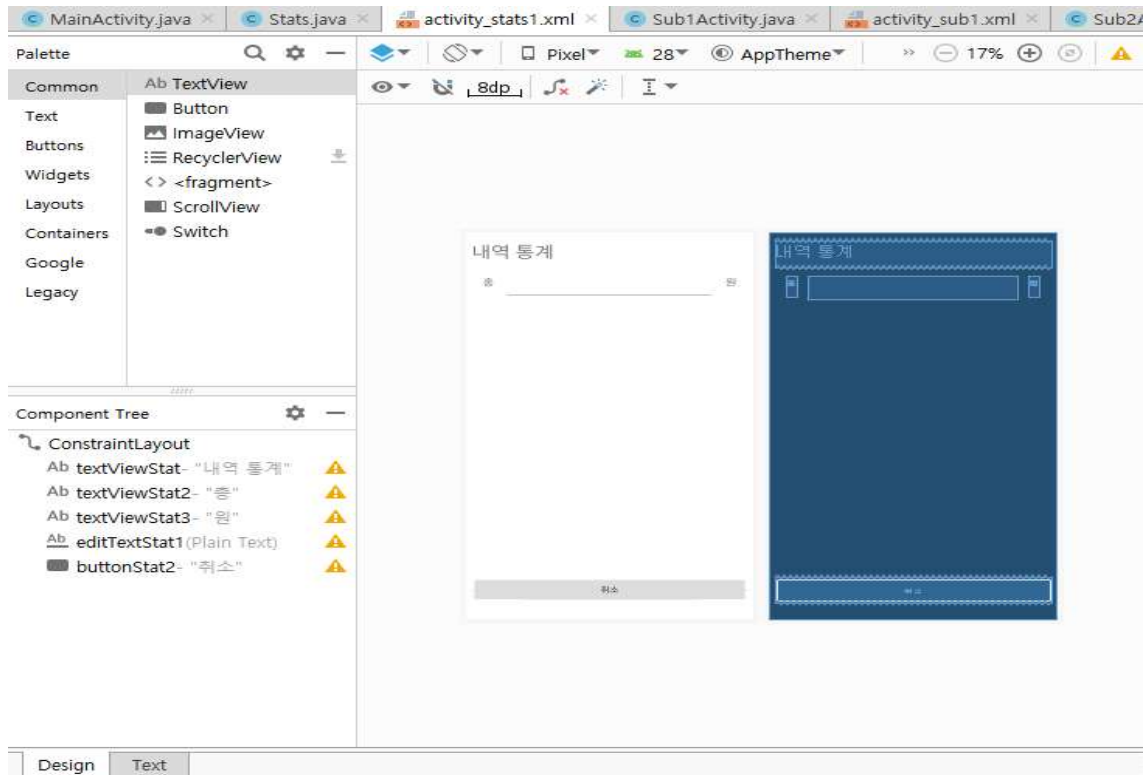


3) 앱 아이콘 이미지가 생성된 것을 확인할 수 있다.

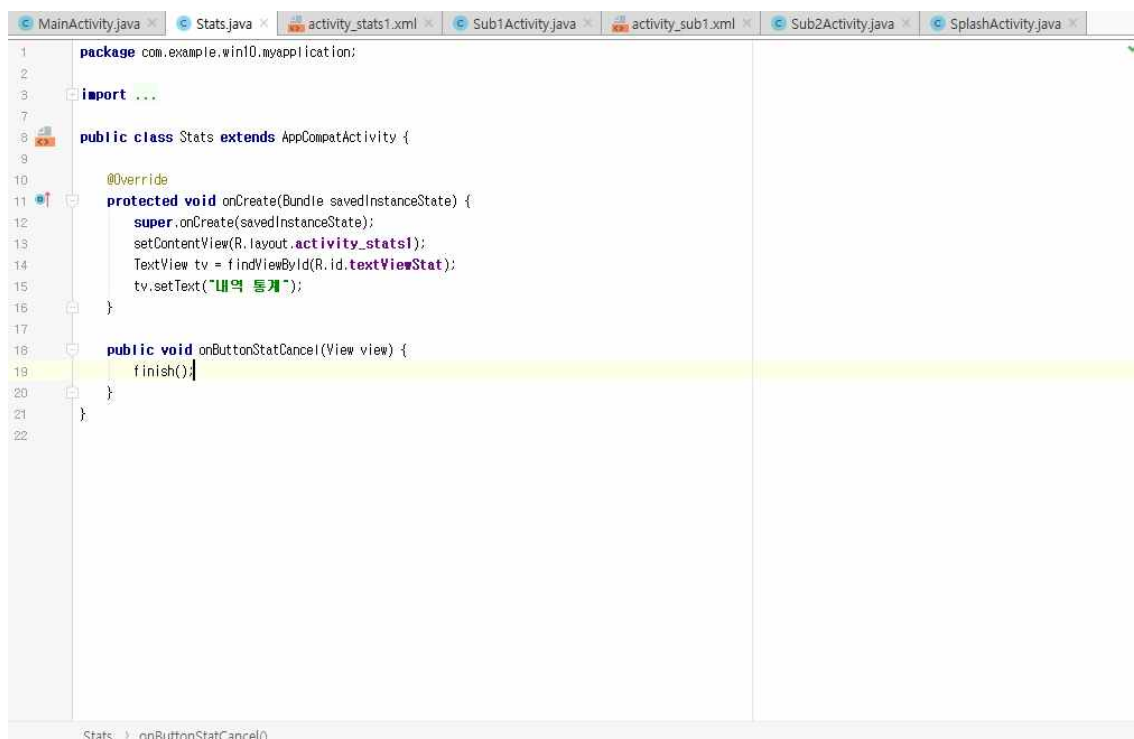


6-4. 통계 이벤트 구현

1) 통계 버튼을 누르면 발생하는 페이지의 틀 생성(activity_stats1.xml)



2) 레이아웃을 불러오기 위해 activity_stats1.xml ID를 가져온다.



3) MyDBHelper 클래스에서는 안드로이드에서 제공하는 데이터베이스 SQLiteOpenHelper 클래스를 상속받는 코드와 SQL쿼리문 작성

```
MyDBHelper.java x Stats.java x SplashActivity.java x MainActivity.java x Ledger.java x LedgerAdapter.java x
3 import android.content.Context;
4 import android.database.sqlite.SQLiteDatabase;
5 import android.database.sqlite.SQLiteOpenHelper;
6
7 import static com.example.win10.myapplication.Stats.KEY_DATE;
8 import static com.example.win10.myapplication.Stats.KEY_MONEY;
9 import static com.example.win10.myapplication.Stats.KEY_TITLE;
10 import static com.example.win10.myapplication.Stats.KEY_USAGE;
11 import static com.example.win10.myapplication.Stats.TABLE_NAME;
12
13 public class MyDBHelper extends SQLiteOpenHelper {
14     public MyDBHelper(Context context) { super(context, name: "My_Account_Data.db", factory: null, version: 4); }
15
16     @Override
17     public void onCreate(SQLiteDatabase db) {
18         //AUTOINCREMENT 속성 사용 시 PRIMARY KEY로 지정한다.
19         String query = String.format("CREATE TABLE %s ("
20             + "%s INTEGER PRIMARY KEY AUTOINCREMENT, "
21             + "%s TEXT, "
22             + "%s INTEGER, "
23             + "%s TEXT);", TABLE_NAME, KEY_TITLE, KEY_MONEY, KEY_USAGE, KEY_DATE);
24         db.execSQL(query);
25     }
26
27     @Override
28     public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
29         String query = String.format("DROP TABLE IF EXISTS %s", TABLE_NAME);
30         db.execSQL(query);
31         onCreate(db);
32     }
33 }
34
35
36
```


- 4) Stats 클래스에서는 한번 입력받고 더 이상 값을 변경하지 못하도록 final static 함수를 사용하는 코드와 데이터베이스 생성 코드, 누적합계를 내는 코드

* 누적합계 내는 코드는 어떻게 값을 가져와 읽어서 누적합계를 내는지 몰라서 자료를 찾아보고 여러 번 시도 끝에 구현 실패

```
Stats.java x SplashActivity.java x MainActivity.java x MyDBHelper.java x Ledger.java
10 public class Stats extends AppCompatActivity {
11     EditText editTextStat1;
12     MyDBHelper mHelper;
13     SQLiteDatabase db;
14
15     final static String TABLE_NAME = "HouseHoldLedger";
16     final static String KEY_TITLE = "title";
17     final static String KEY_MONEY = "money";
18     final static String KEY_USAGE = "usage";
19     final static String KEY_DATE = "date";
20
21     @Override
22     protected void onCreate(Bundle savedInstanceState) {
23         super.onCreate(savedInstanceState);
24         setContentView(R.layout.activity_stats1);
25         TextView tv = findViewById(R.id.textViewStat);
26         tv.setText("내역 통계");
27
28         editTextStat1 = (EditText) findViewById(R.id.editTextStat1);
29
30         //데이터베이스 생성
31         mHelper = new MyDBHelper( context: this);
32         db = mHelper.getWritableDatabase();
33
34         //통계 표시
35         //String queryPriceSum = String.format( " SELECT SUM(price) FROM %s", TABLE_NAME);
36         //db.rawQuery( queryPriceSum, null );
37         //int sum = 0;
38         //Log.d(TAG, "sum : " + sum);
39         //editTextStat1.setText(sum);
40     }
```

7. 테스트

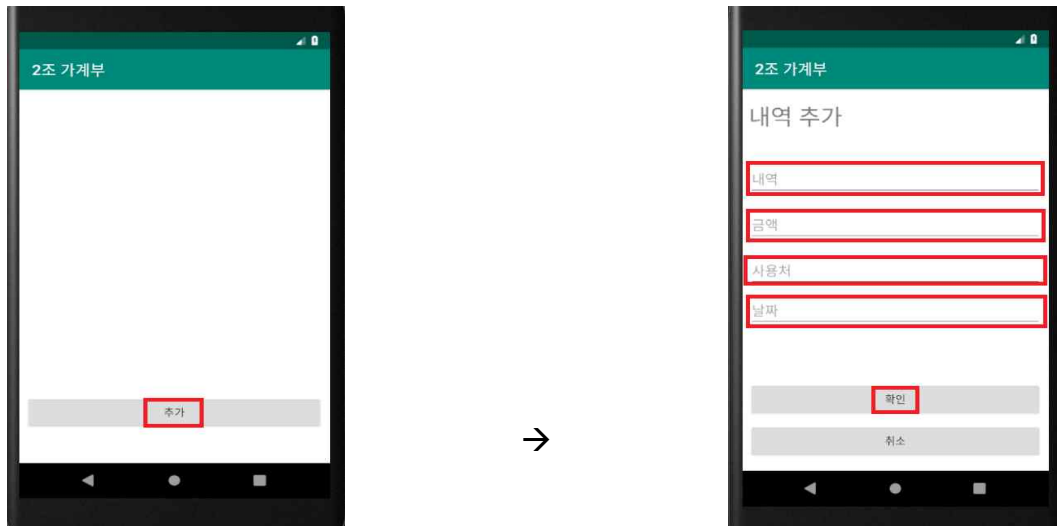
7-1. 프로젝트 초반

- 1) 초반에는 버튼 이벤트를 구현하지 않아 메인 페이지만 출력

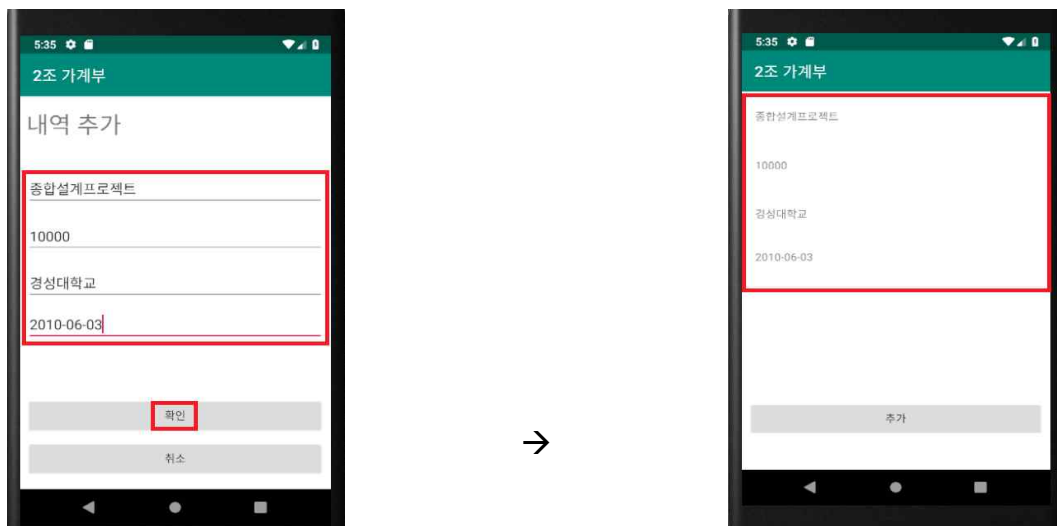


7-2. 프로젝트 중반

1) 추가 버튼을 누르면 내역 추가를 할 수 있다.



2) 내역 추가 페이지에서 입력을 하면 리스트로 담겨 메인 화면에 출력이 된다.



8. 수정 및 보완

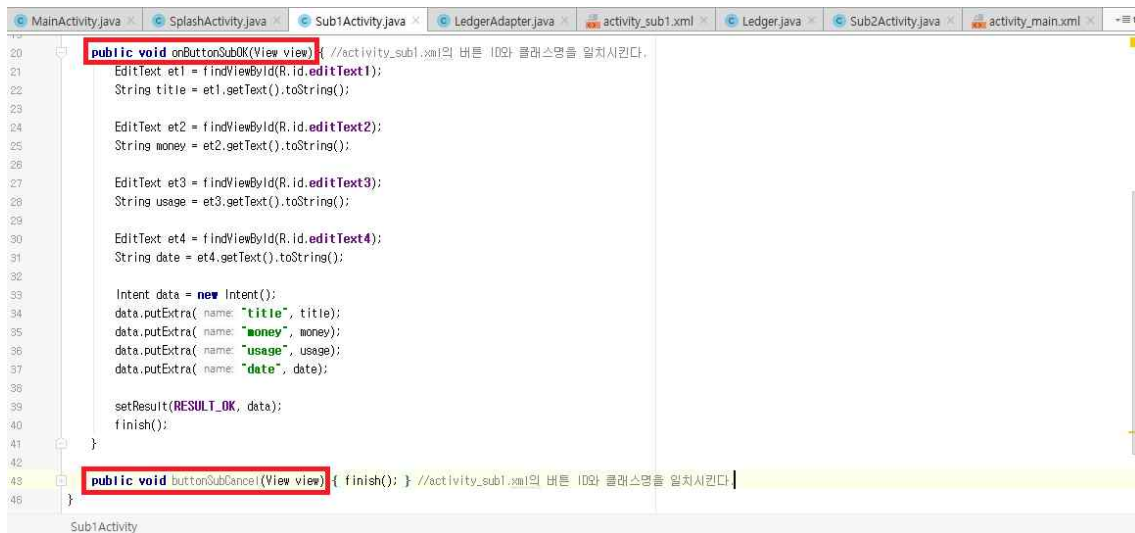
8-1. 내역 추가 시 버튼 이벤트 에러 수정

1) 내역 추가 시 버튼 이벤트 에러가 나는 것을 수정한다.

- 사용할 xml에서 버튼 ID와 사용할 클래스에서의 버튼 클래스 명을 일치시켜준다.



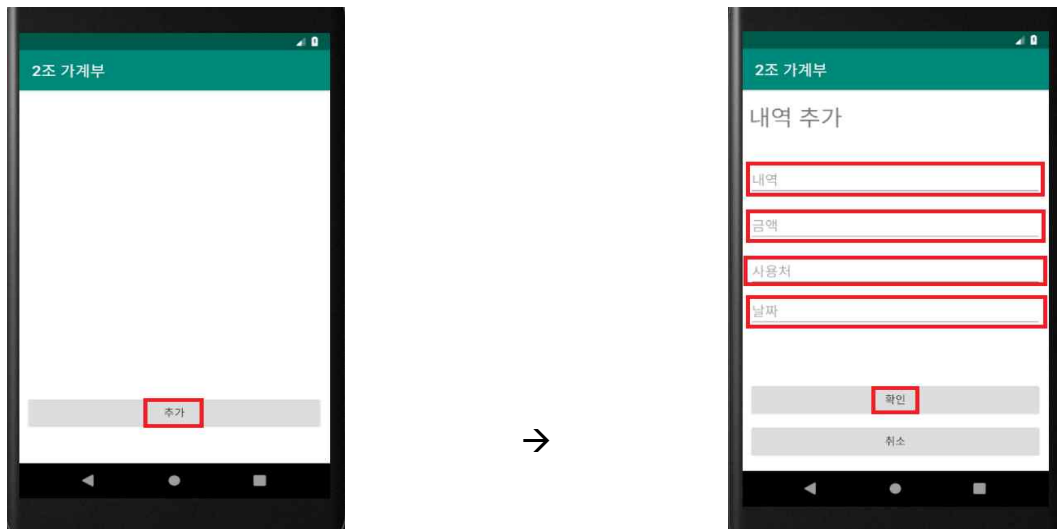
```
99 android:layout_height="wrap_content"
100 android:layout_marginStart="8dp"
101 android:layout_marginEnd="8dp"
102 android:layout_marginBottom="8dp"
103 android:onClick="onButtonSubOK"
104 android:text="확인"
105 app:layout_constraintBottom_toTopOf="@id/buttonSubCancel"
106 app:layout_constraintEnd_toEndOf="parent"
107 app:layout_constraintHorizontal_bias="0.0"
108 app:layout_constraintStart_toStartOf="parent"
109 tools:ignore="OnClick" />
110
111 <Button
112     android:id="@+id/buttonSubCancel"
113     android:layout_width="0dp"
114     android:layout_height="wrap_content"
115     android:layout_marginStart="8dp"
116     android:layout_marginEnd="8dp"
117     android:layout_marginBottom="16dp"
118     android:onClick="onButtonSubCancel"
119     android:text="취소"
120     app:layout_constraintBottom_toBottomOf="parent"
121     app:layout_constraintEnd_toEndOf="parent"
122     app:layout_constraintStart_toStartOf="parent" />
```



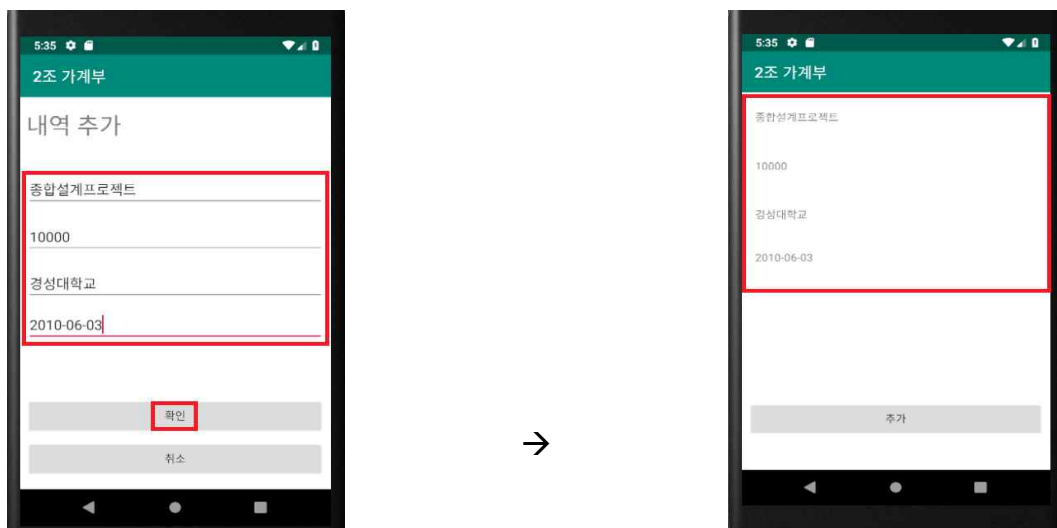
```
20 public void onButtonSubOK(View view) { //activity_sub1.xml의 버튼 ID와 클래스명을 일치시킨다.
21     EditText et1 = findViewById(R.id.editText1);
22     String title = et1.getText().toString();
23
24     EditText et2 = findViewById(R.id.editText2);
25     String money = et2.getText().toString();
26
27     EditText et3 = findViewById(R.id.editText3);
28     String usage = et3.getText().toString();
29
30     EditText et4 = findViewById(R.id.editText4);
31     String date = et4.getText().toString();
32
33     Intent data = new Intent();
34     data.putExtra( name: "title", title);
35     data.putExtra( name: "money", money);
36     data.putExtra( name: "usage", usage);
37     data.putExtra( name: "date", date);
38
39     setResult(RESULT_OK, data);
40     finish();
41 }
42
43 public void buttonSubCancel(View view) { finish(); } //activity_sub1.xml의 버튼 ID와 클래스명을 일치시킨다.
44 }
```

2) 정상적인 실행과정

- 추가 버튼을 누르면 내역 추가를 할 수 있게 된다.



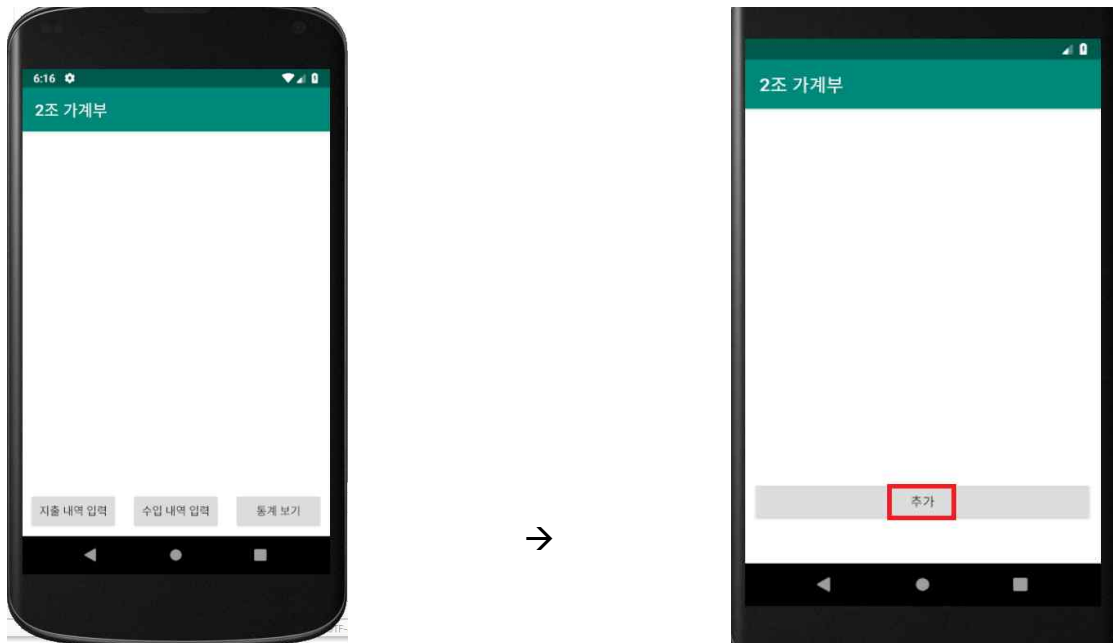
- 추가 내역을 입력하고 확인을 누르면 정상적으로 실행하는 것을 확인할 수 있다.



8-2. 메인 화면 보완

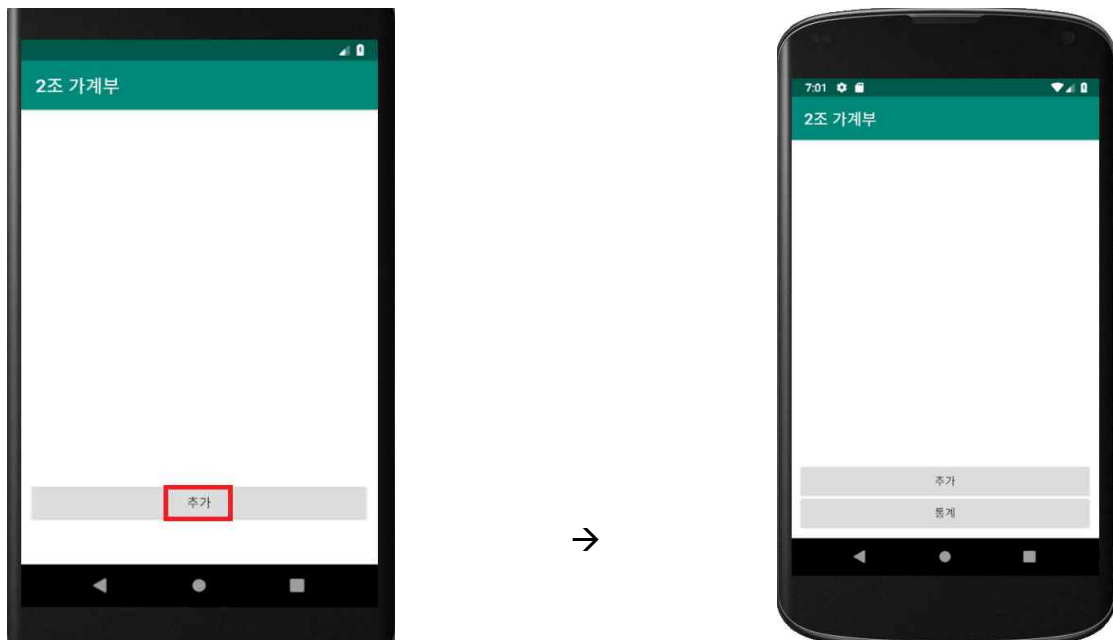
1) 프로젝트 초반의 메인 화면 구성을 변경

- 추가 버튼을 누르면 내역 입력을 할 수 있게 간단하게 변경



2) 프로젝트 중반의 메인 화면 구성을 변경

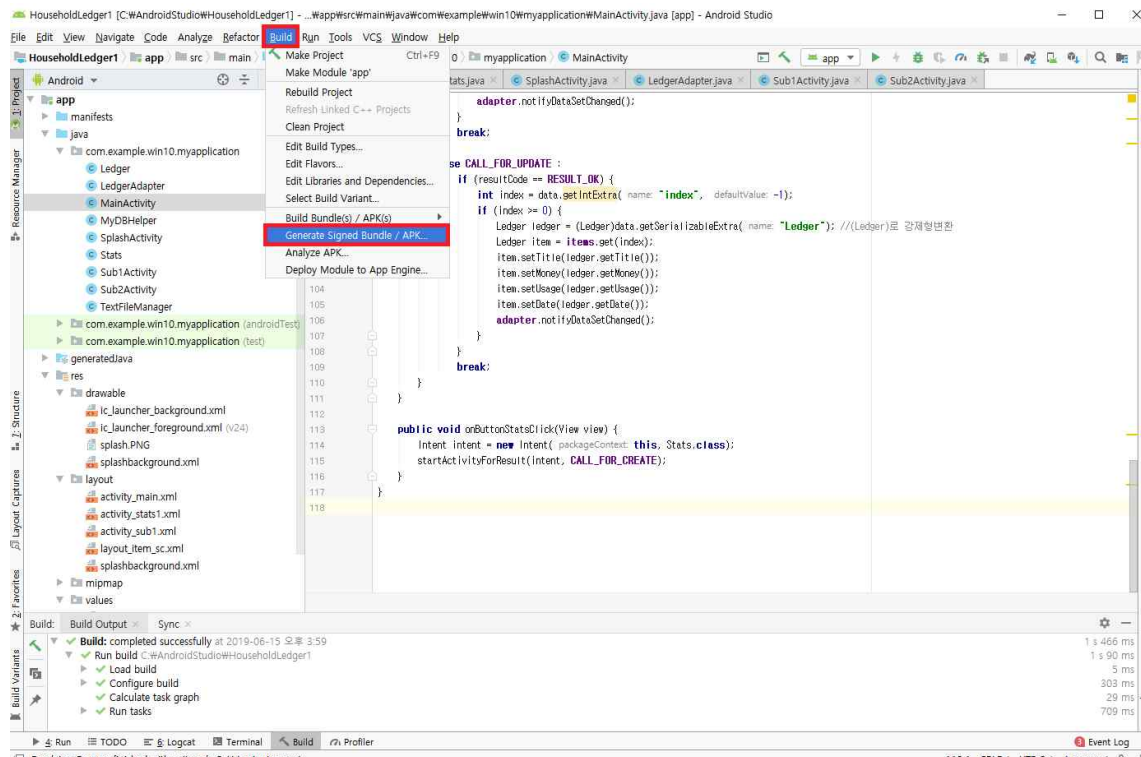
- 통계 버튼 추가



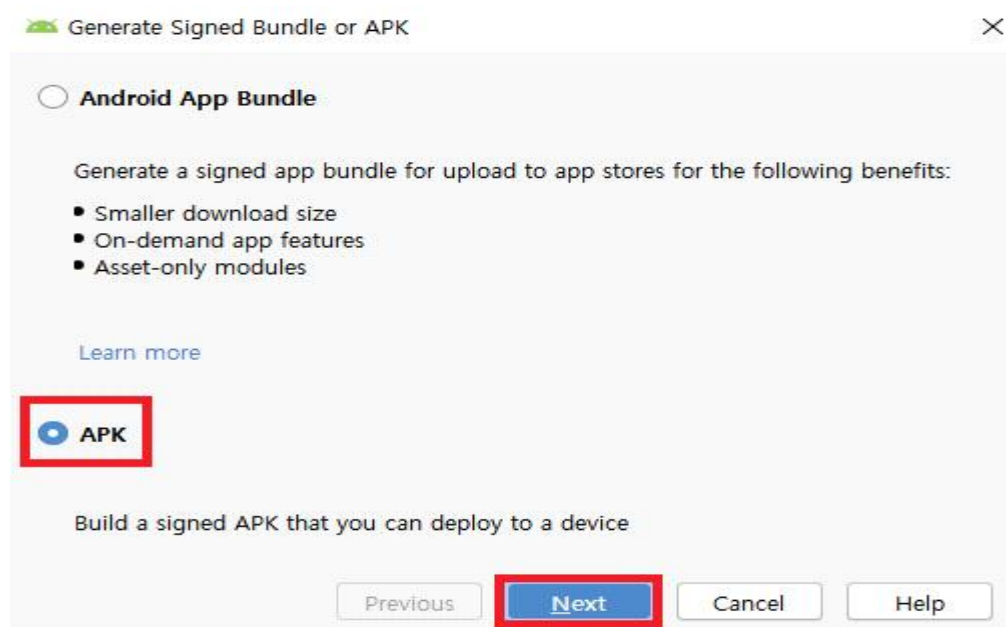
9. 프로젝트 결과

9-1. APK 파일 추출

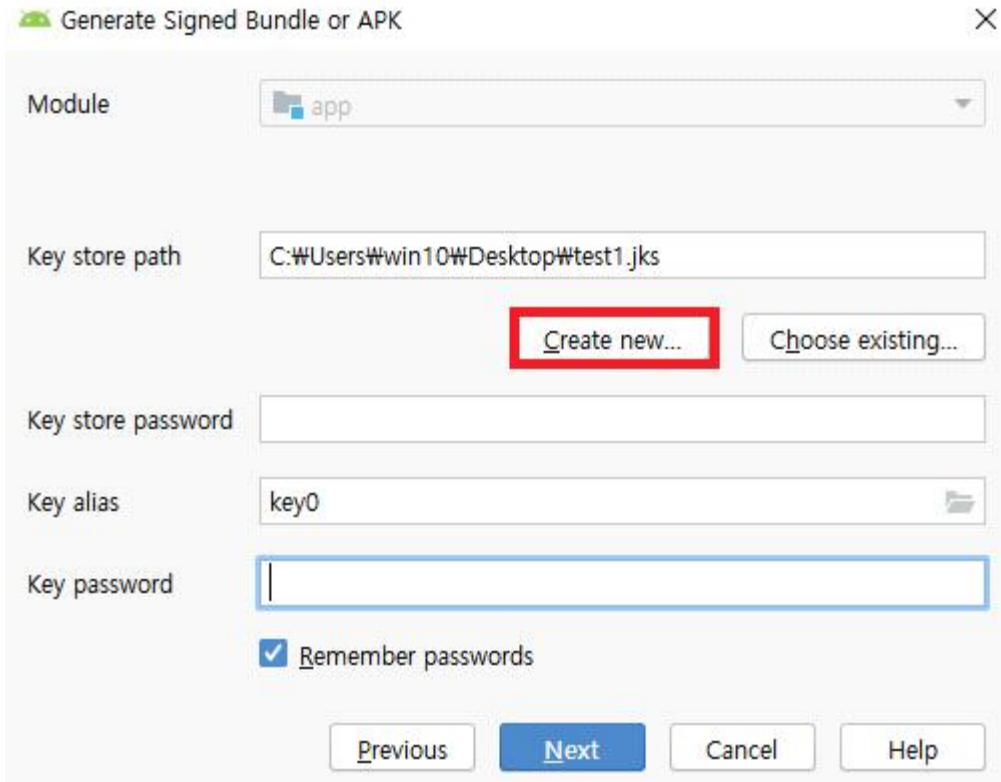
1) 메뉴 탭 -> Build -> Generate Signed Bundle/APK



2) APK 선택 후 Next



3) Key 값 저장할 폴더 생성



Generate Signed Bundle or APK

Module: app

Key store path: C:\Users\win10\Desktop\test1.jks

Create new... Choose existing...

Key store password:

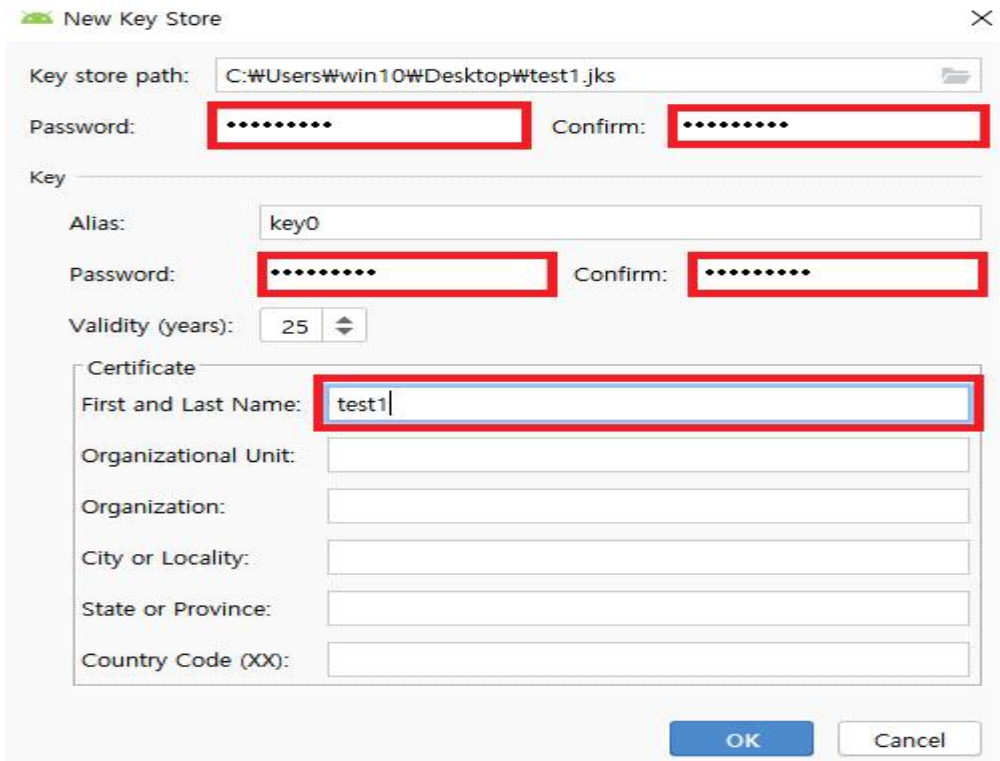
Key alias: key0

Key password:

☒ Remember passwords

Previous **Next** Cancel Help

4) 필수 입력 부분 입력 후 OK



New Key Store

Key store path: C:\Users\win10\Desktop\test1.jks

Password: Confirm:

Key

Alias: key0

Password: Confirm:

Validity (years): 25

Certificate

First and Last Name: **test1**

Organizational Unit:

Organization:

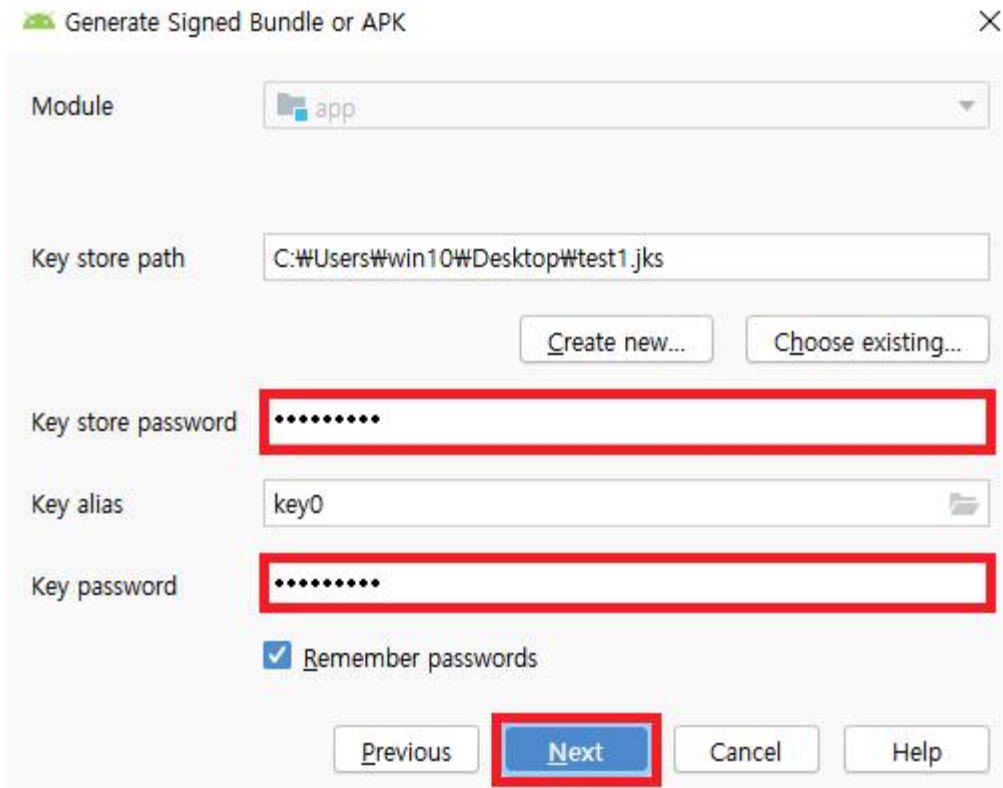
City or Locality:

State or Province:

Country Code (XX):

OK Cancel

5) 설정한 패스워드 입력 후 Next



Generate Signed Bundle or APK

Module: app

Key store path: C:\Users\win10\Desktop\test1.jks

Key store password: [Redacted]

Key alias: key0

Key password: [Redacted]

☒ Remember passwords

Previous Next Cancel Help

6) release , V2(Full APK Signature) 선택 후 Finish



Generate Signed Bundle or APK

Destination Folder: C:\Users\win10\Desktop

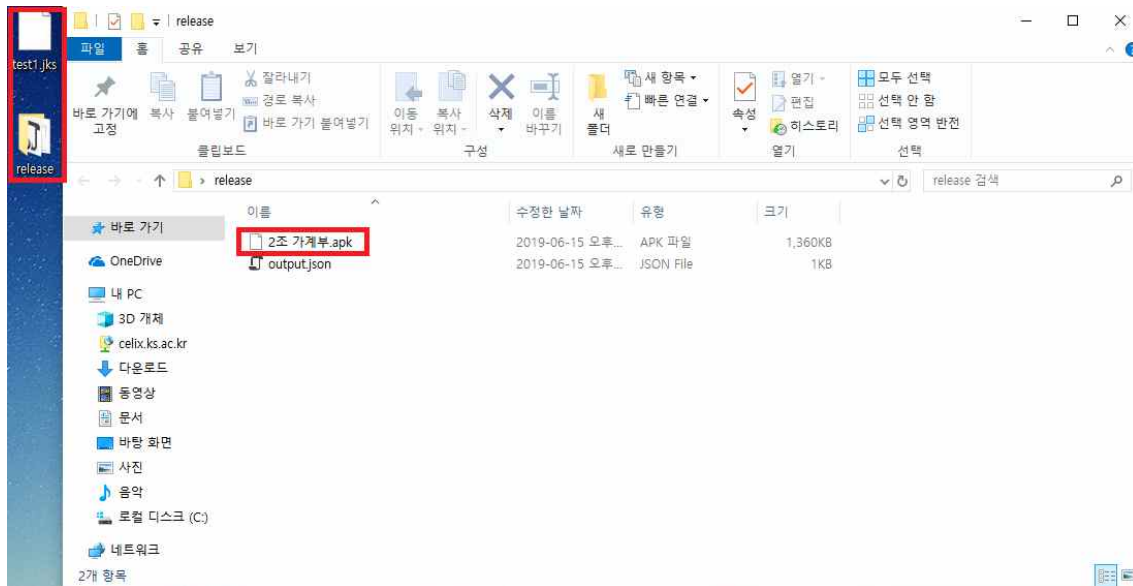
Build Variants:

- debug
- release

Signature Versions: ☐ V1 (Jar Signature) ☒ V2 (Full APK Signature) [Signature Help](#)

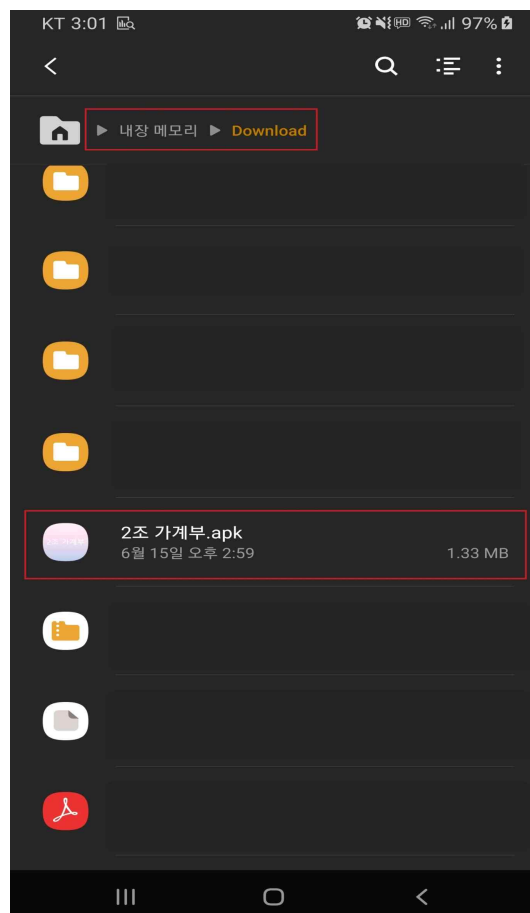
Previous Finish Cancel Help

7) 지정 경로에 jks파일, release(apk, json)파일이 생성된 것을 확인

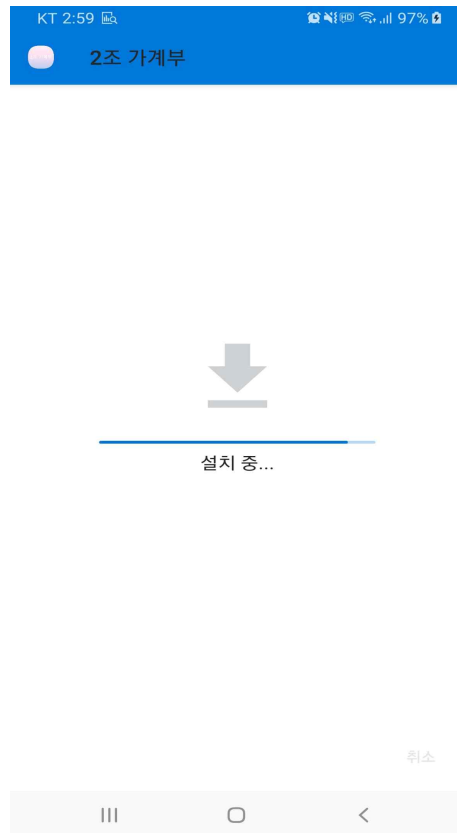


9-2. 스마트폰에 APK 파일 설치

1) 스마트폰의 지정 경로에 .apk 파일 다운로드



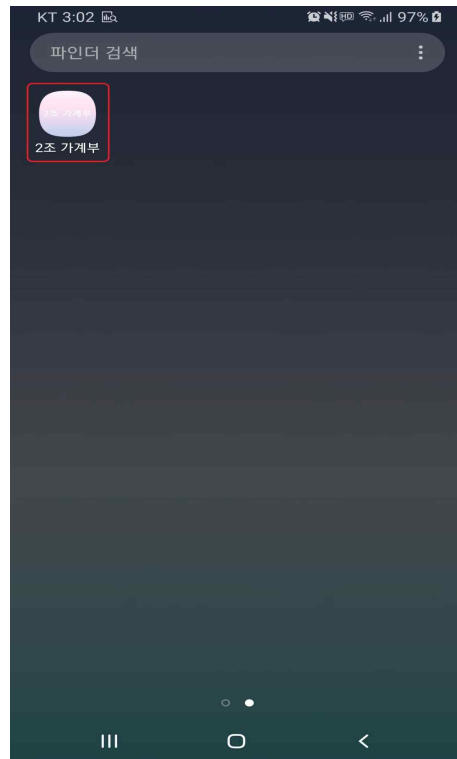
2) 다운로드 후 .apk 파일 설치



3) 설치 후 완료 또는 열기 클릭



4) 2조 가계부 앱 설치가 정상적으로 완료된 것을 확인할 수 있다.



9-3. 설치한 APK 파일 실행

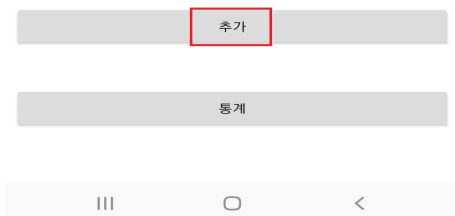
1) 메인 화면이 뜨기 전에 스플래시가 정상적으로 실행



2) 스플래쉬가 실행된 후 메인 화면 출력



3) 추가 버튼 클릭하면 내역 추가 페이지로 넘어간다.



- 4) 추가할 내역을 입력 후 확인 버튼을 누르면 입력한 내역이 리스트에 담겨서 메인 화면에 출력된다.

KT 5:46 97%

2조 가계부

내역 추가

종합설계프로젝트

10000

경성대학교

2019-06-15

확인

취소

III O <

→

KT 5:46 97%

2조 가계부

종합설계프로젝트

10000

경성대학교

2019-06-15

추가

통계

III O <

- 5) 리스트에 있는 내역을 짧게 한 번 누르면 수정 페이지로 이동한다.

KT 2:35 100%

2조 가계부

종합설계프로젝트

10000

경성대학교

2019-06-15

추가

통계

III O <

→

KT 2:35 100%

2조 가계부

내역 수정

종합설계프로젝트

10000

경성대학교

2019-06-15

확인

취소

III O <

6) 내역을 수정 후 확인 버튼을 누른다.

KT 2:35 100%

2조 가계부

내역 수정

종합설계프로젝트

10000

경성대학교

2019-06-15

KT 2:36 100%

2조 가계부

내역 수정

종합설계프로젝트

20000

경성대학교

2019-06-16

확인

취소

||| ○ <



확인

취소

||| ○ <

7) 내역을 길게 한 번 클릭하면 내역이 삭제된다.

KT 2:35 100%

2조 가계부

종합설계프로젝트

10000

경성대학교

2019-06-15

KT 5:47 97%

2조 가계부

추가

통계

||| ○ <

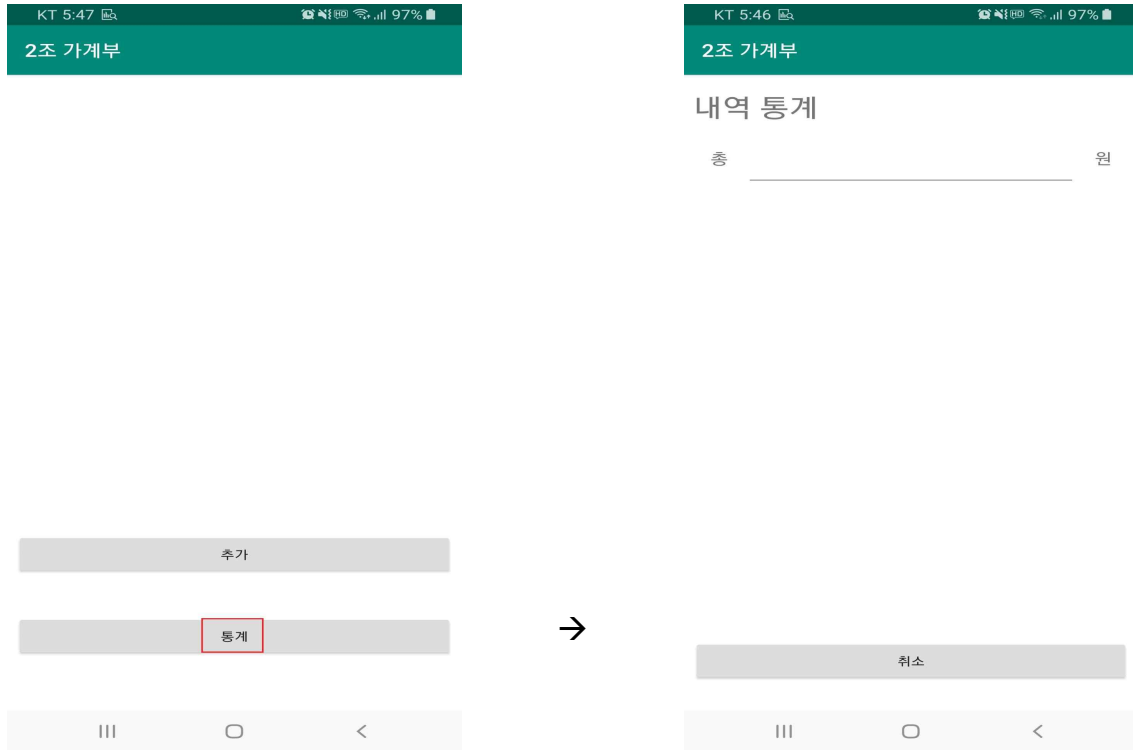


추가

통계

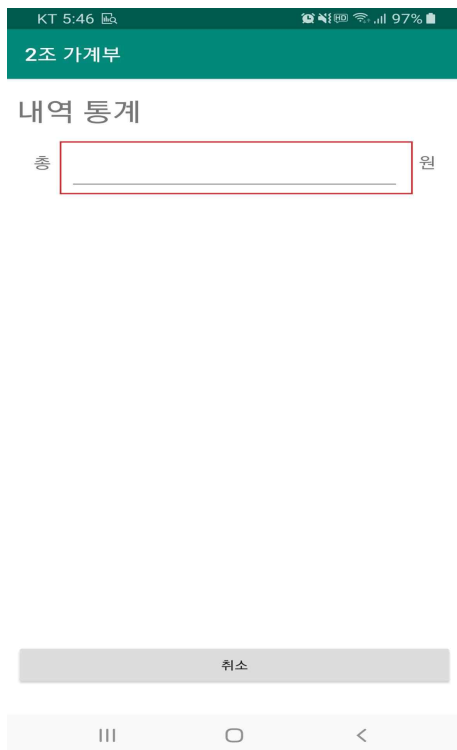
||| ○ <

8) 통계 버튼을 누르면 통계 페이지로 이동한다.



9) 통계 페이지로 이동하는 동시에 TextView에 누적합계가 출력된다.

* 누적합계 내는 코드를 구현 실패하여 값이 출력되지 않는다.



10. 결론

10-1. 프로젝트 결과의 문제점

- 1) 내역 추가 페이지에서 취소 버튼 클릭 시 스플래쉬가 호출되는 점
- 2) 내역 추가 및 내역 수정에서 지출과 수입을 분리하지 않아서
입력한 내역이 지출인지 수입인지 모르는 점
- 3) 통계 버튼 이벤트 구현을 실패한 점
- 4) 내역 추가 및 내역 수정에서 입력한 값을 저장할 데이터베이스
구현 실패한 점

10-2. 해결 및 보완책

- 1) 시간적으로 여유가 있을 때나 다음 프로젝트를 할 때 이러한
문제점들을 해결 및 보완하는 것을 목표로 잡겠습니다.

10-3. 프로젝트를 하면서 느낀점

- 1) 주제를 변경하고 나서부터 구체적인 세부계획을 잡지 못하여
프로젝트를 진행하는데 어려움이 있었습니다. 그렇지만 모르는
내용들은 인터넷으로 찾아보거나 안드로이드 프로그래밍
수업시간에 배운 내용들을 다시 한 번 차근차근 읽어보면서
프로젝트를 진행하였습니다. 비록 완벽하게 완성을 하지는
못하였지만 혼자서 앱을 만들었다는 것만으로도 굉장히 뿌듯하고
모르는 것을 찾아보면서 만들었기 때문에 공부가 많이
되었습니다.

11. 부록

11-1. 전체 소스코드

1) Manifests.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.win10.myapplication">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".Stats"></activity>
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".SplashActivity" />
        <activity android:name=".Sub1Activity" />
        <activity android:name=".Sub2Activity" />
        <activity android:name=".Ledger" />
        <activity android:name=".LedgerAdapter" />
    </application>

</manifest>
```


2) MainActivity.java

```
package com.example.win10.myapplication;

import android.content.Intent;
import android.database.sqlite.SQLiteDatabase;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.EditText;
import android.widget.ListView;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {
    //MyDBHelper mHelper;
    //SQLiteDatabase db;

    private static final int CALL_FOR_UPDATE = 0;
    private static final int CALL_FOR_CREATE = 1;
    ArrayList<Ledger> items;
    LedgerAdapter adapter;

    //final static String TABLE_NAME = "HouseHoldLedger";
    //final static String KEY_TITLE = "title";
    //final static String KEY_MONEY = "money";
    //final static String KEY_USAGE = "usage";
    //final static String KEY_DATE = "date";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```

//setContentView(R.layout.splashbackground);
setContentView(R.layout.activity_main);
Intent intent = new Intent(this, SplashActivity.class);
startActivity(intent);

//데이터베이스 생성
//mHelper = new MyDBHelper(this);
//db = mHelper.getWritableDatabase();

//1. 집합데이터를 생성하기 위한 공간확보 : ArrayList
items = new ArrayList<>();
//items.add(new Ledger("내역", "금액", "사용처", "날짜"));

//2. 객체를 저장하기 위한 Adapter 생성
//ArrayAdapter : 문자열만 저장함, ArrayAdapter는 사용하지 못한다.
adapter = new LedgerAdapter(this, R.layout.layout_item_sc,
items);

//3. AdapterView(ListView)에 Adapter 등록
ListView lv1 = findViewById(R.id.ListViewMain);
lv1.setAdapter(adapter);

lv1.setOnItemClickListener(new AdapterView.OnItemClickListener()
{
    @Override
    public void onItemClick(AdapterView<?> parent, View view,
int position, long id) {
        Intent intent = new Intent(MainActivity.this,
Sub2Activity.class);
        intent.putExtra("index", position);
        intent.putExtra("ledger", items.get(position));
        startActivityForResult(intent, CALL_FOR_UPDATE);
    }
});

```

```

        lv1.setOnItemLongClickListener(new
AdapterView.OnItemLongClickListener() {
    @Override
    public boolean onItemLongClick(AdapterView<?> parent,
View view, int position, long id) {
        items.remove(position); //현재 위치에서 삭제한다.
        //변경된 데이터를 공지해준다.
        adapter.notifyDataSetChanged();

        return false;
    }
});
}

public void onButtonMainClick(View view) {
    Intent intent = new Intent(this, Sub1Activity.class);
    startActivityForResult(intent, CALL_FOR_CREATE);
}

@Override
protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    switch (requestCode) {
        case CALL_FOR_CREATE :
            if (resultCode == RESULT_OK) {
                String title = data.getStringExtra("title");
                String money = data.getStringExtra("money");
                String usage = data.getStringExtra("usage");
                String date = data.getStringExtra("date");

                items.add(new Ledger(title, money, usage, date));
                adapter.notifyDataSetChanged();
            }
    }
}

```

```

    }
    break;

    case CALL_FOR_UPDATE :
        if (resultCode == RESULT_OK) {
            int index = data.getIntExtra("index", -1);
            if (index >= 0) {
                Ledger ledger =
(Ledger)data.getSerializableExtra("Ledger"); //(Ledger)로 강제형변환
                Ledger item = items.get(index);
                item.setTitle(ledger.getTitle());
                item.setMoney(ledger.getMoney());
                item.setUsage(ledger.getUsage());
                item.setDate(ledger.getDate());
                adapter.notifyDataSetChanged();
            }
        }
        break;
    }
}

public void onButtonStatsClick(View view) {
    Intent intent = new Intent(this, Stats.class);
    startActivityForResult(intent, CALL_FOR_CREATE);
}
}

```

3) 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"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
```

```
<ListView
    android:id="@+id/ListViewMain"
    android:layout_width="367dp"
    android:layout_height="373dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="58dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0" />
```

```
<Button
    android:id="@+id/button1"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="16dp"
    android:onClick="onButtonMainClick"
    android:text="추가"
```

```
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/ListViewMain" />
```

<Button

```
android:id="@+id/button2"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginStart="8dp"
android:layout_marginTop="16dp"
android:layout_marginEnd="8dp"
android:layout_marginBottom="32dp"
android:onClick="onButtonStatsClick"
android:text="통계"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/button1" />
```

</android.support.constraint.ConstraintLayout>

```

4) Ledger.java
package com.example.win10.myapplication;

import java.io.Serializable;

public class Ledger implements Serializable { //implements Serializable
= 직렬화 시켜준다.(Byte형태, String형태로 바꾸기 위해서)
    private String title;
    private String money;
    private String usage;
    private String date;

    //생성자 함수 : 마우스 우클릭 -> Generate -> Constructor

    public Ledger(String title, String money, String usage, String date) {
        this.title = title;
        this.money = money;
        this.usage = usage;
        this.date = date;
    }

    //접근자, 설정자 함수 -> Generate -> Getter and Setter

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getMoney() {
        return money;
    }
}

```

```
public void setMoney(String money) {  
    this.money = money;  
}  
  
public String getUsage() {  
    return usage;  
}  
  
public void setUsage(String usage) {  
    this.usage = usage;  
}  
  
public String getDate() {  
    return date;  
}  
  
public void setDate(String date) {  
    this.date = date;  
}  
}
```



```

5) LedgerAdapter.java
package com.example.win10.myapplication;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.TextView;

import java.util.ArrayList;

class LedgerAdapter extends BaseAdapter {
    ArrayList<Ledger> items;
    Context context;
    int itemlayout;

    public LedgerAdapter(Context context, int layout_item_sc,
ArrayList<Ledger> items) {
        this.items = items;
        this.itemlayout = layout_item_sc;
        this.context = context;
    }

    @Override
    public int getCount() {
        return items.size();
    }

    @Override
    public Object getItem(int position) {
        return items.get(position);
    }
}

```

```

@Override
//item간의 고유한 값을 넘겨준다.
public long getItemId(int position) {
    return position;
}

@Override
public View getView(int position, View convertView, ViewGroup
parent) {
    //convertView : 리스트를 담는 틀을 만드는 것
    if (convertView == null) {
        LayoutInflater inflater =
(LayoutInflater)context.getSystemService(context.LAYOUT_INFLATER_SERV
ICE);

        convertView = inflater.inflate(itemlayout, null);
    }
    TextView tv1 = convertView.findViewById(R.id.textViewitem1);
    tv1.setText(items.get(position).getTitle());

    TextView tv2 = convertView.findViewById(R.id.textViewitem2);
    tv2.setText(items.get(position).getMoney());

    TextView tv3 = convertView.findViewById(R.id.textViewitem3);
    tv3.setText(items.get(position).getUsage());

    TextView tv4 = convertView.findViewById(R.id.textViewitem4);
    tv4.setText(items.get(position).getDate());

    return convertView;
}
}

```

6) Sub1Activity.java

```
package com.example.win10.myapplication;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;

public class Sub1Activity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sub1);
        TextView tv = findViewById(R.id.textViewSub);
        tv.setText("내역 추가");
    }

    public void onButtonSubOK(View view) { //activity_sub1.xml의 버튼
        ID와 클래스명을 일치시킨다.
        EditText et1 = findViewById(R.id.editText1);
        String title = et1.getText().toString();

        EditText et2 = findViewById(R.id.editText2);
        String money = et2.getText().toString();

        EditText et3 = findViewById(R.id.editText3);
        String usage = et3.getText().toString();

        EditText et4 = findViewById(R.id.editText4);
        String date = et4.getText().toString();
    }
}
```

```
        Intent data = new Intent();
        data.putExtra("title", title);
        data.putExtra("money", money);
        data.putExtra("usage", usage);
        data.putExtra("date", date);

        setResult(RESULT_OK, data);
        finish();
    }

    public void buttonSubCancel(View view) {
        finish();
    } //activity_sub1.xml의 버튼 ID와 클래스명을 일치시킨다.
}
```

7) activity_sub1.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"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Sub1Activity">

    <TextView
        android:id="@+id/textViewSub"
        android:layout_width="0dp"
        android:layout_height="50dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="16dp"
        android:text="내역 수정"
        android:textSize="30sp"
        app:layout_constraintBottom_toTopOf="@+id/editText1"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/editText1"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="8dp"
        android:ems="10"
```

```
android:hint="내역"  
android:inputType="textPersonName"  
app:layout_constraintBottom_toTopOf="@+id/editText2"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toBottomOf="@+id/textViewSub" />
```

<EditText

```
android:id="@+id/editText2"  
android:layout_width="0dp"  
android:layout_height="wrap_content"  
android:layout_marginStart="8dp"  
android:layout_marginTop="8dp"  
android:layout_marginEnd="8dp"  
android:layout_marginBottom="8dp"  
android:ems="10"  
android:hint="금액"  
android:inputType="textPersonName"  
app:layout_constraintBottom_toTopOf="@+id/editText3"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toBottomOf="@+id/editText1" />
```

<EditText

```
android:id="@+id/editText3"  
android:layout_width="0dp"  
android:layout_height="wrap_content"  
android:layout_marginStart="8dp"  
android:layout_marginTop="8dp"  
android:layout_marginEnd="8dp"  
android:layout_marginBottom="8dp"  
android:ems="10"  
android:hint="사용처"  
android:inputType="textPersonName"
```

```
app:layout_constraintBottom_toTopOf="@+id/editText4"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toBottomOf="@+id/editText2" />
```

<EditText

```
android:id="@+id/editText4"  
android:layout_width="0dp"  
android:layout_height="wrap_content"  
android:layout_marginStart="8dp"  
android:layout_marginTop="8dp"  
android:layout_marginEnd="8dp"  
android:ems="10"  
android:hint="날짜"  
android:inputType="textPersonName"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toBottomOf="@+id/editText3" />
```

<Button

```
android:id="@+id/buttonSubOK"  
android:layout_width="0dp"  
android:layout_height="wrap_content"  
android:layout_marginStart="8dp"  
android:layout_marginEnd="8dp"  
android:layout_marginBottom="8dp"  
android:onClick="onButtonSubOK"  
android:text="확인"  
app:layout_constraintBottom_toTopOf="@+id/buttonSubCancel"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.0"  
app:layout_constraintStart_toStartOf="parent"  
tools:ignore="OnClick" />
```

```
<Button
    android:id="@+id/buttonSubCancel"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="16dp"
    android:onClick="onButtonSubCancel"
    android:text="취소"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    tools:ignore="OnClick" />
```

```
</android.support.constraint.ConstraintLayout>
```


8) Sub2Activity.java

```
package com.example.win10.myapplication;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.EditText;
import android.widget.TextView;

public class Sub2Activity extends AppCompatActivity {
    int index;
    Ledger ledger;
    EditText et1, et2, et3, et4;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sub1);
        TextView tv = findViewById(R.id.textViewSub);
        tv.setText("내역 수정");

        Intent intent = getIntent();
        index = intent.getIntExtra("index", -1); //자료형이 정수형이기
        때문에 getIntExtra("이름표", 초기값)로 사용한다.

        if (index != -1) {
            //(어떤 자료형인지 모르기 때문에 Ledger로 강제형변환)직렬화
            되어서 들어오는 객체는 순서대로 풀어준다.
            ledger = (Ledger)intent.getSerializableExtra("ledger");
        }
        et1 = findViewById(R.id.editText1);
        et1.setText(ledger.getTitle());

        et2 = findViewById(R.id.editText2);
```

```
        et2.setText(ledger.getMoney());

        et3 = findViewById(R.id.editText3);
        et3.setText(ledger.getUsage());

        et4 = findViewById(R.id.editText4);
        et4.setText(ledger.getDate());
    }
}
```

9) layout_item_sc.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"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/textViewItem1"
        android:layout_width="0dp"
        android:layout_height="50dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="8dp"
        android:text="TextView"
        app:layout_constraintBottom_toTopOf="@+id/textViewItem2"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textViewItem2"
        android:layout_width="0dp"
        android:layout_height="50dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="8dp"
        android:text="TextView"
        app:layout_constraintBottom_toTopOf="@+id/textViewItem3"
        app:layout_constraintEnd_toEndOf="parent"
```

```
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textViewItem1" />
```

```
<TextView
    android:id="@+id/textViewItem3"
    android:layout_width="0dp"
    android:layout_height="50dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:text="TextView"
    app:layout_constraintBottom_toTopOf="@+id/textViewItem4"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textViewItem2" />
```

```
<TextView
    android:id="@+id/textViewItem4"
    android:layout_width="0dp"
    android:layout_height="50dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:text="TextView"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textViewItem3" />
```

```
</android.support.constraint.ConstraintLayout>
```

10) SplashActivity.java

```
package com.example.win10.myapplication;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;

public class SplashActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.splashbackground);
        startLoading();
    }
    private void startLoading() {
        Handler handler = new Handler();
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                finish();
            }
        }, 2000);
    }
}
```

11) drawable/splashbackground.xml

```
<?xml version="1.0" encoding="utf-8"?>
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
    <item>
        <png
            android:src="@drawable/splash"/>
        </item>
</layer-list>
```

12) layout/splashbackground.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/splash">
</LinearLayout>
```

13) MyDBHelper.java

```
package com.example.win10.myapplication;
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import static com.example.win10.myapplication.Stats.KEY_DATE;
import static com.example.win10.myapplication.Stats.KEY_MONEY;
import static com.example.win10.myapplication.Stats.KEY_TITLE;
import static com.example.win10.myapplication.Stats.KEY_USAGE;
import static com.example.win10.myapplication.Stats.TABLE_NAME;
public class MyDBHelper extends SQLiteOpenHelper {
    public MyDBHelper(Context context) {
        super(context, "My_Account_Data.db", null, 4);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        //AUTOINCREMENT 속성 사용 시 PRIMARY KEY로 지정한다.
        String query = String.format("CREATE TABLE %s ("
            + "%s INTEGER PRIMARY KEY AUTOINCREMENT, "
            + "%s TEXT, "
            + "%s INTEGER, "
            + "%s TEXT);", TABLE_NAME, KEY_TITLE, KEY_MONEY,
            KEY_USAGE, KEY_DATE);
        db.execSQL(query);
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int
        newVersion) {
        String query = String.format("DROP TABLE IF EXISTS %s",
            TABLE_NAME);
        db.execSQL(query);
        onCreate(db);
    }
}
```


14) Stats.java

```
package com.example.win10.myapplication;

import android.database.sqlite.SQLiteDatabase;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;

public class Stats extends AppCompatActivity {
    EditText editTextStat1;
    MyDBHelper mHelper;
    SQLiteDatabase db;

    final static String TABLE_NAME = "HouseHoldLedger";
    final static String KEY_TITLE = "title";
    final static String KEY_MONEY = "money";
    final static String KEY_USAGE = "usage";
    final static String KEY_DATE = "date";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_stats1);
        TextView tv = findViewById(R.id.textViewStat);
        tv.setText("내역 통계");

        editTextStat1 = (EditText)findViewById(R.id.editTextStat1);

        //데이터베이스 생성
        mHelper = new MyDBHelper(this);
        db = mHelper.getWritableDatabase();
    }
}
```

```

        //통계 표시
        //String queryPriceSum = String.format( " SELECT SUM(price)
FROM %s", TABLE_NAME);
        //db.rawQuery( queryPriceSum, null );
        //int sum = 0;
        //Log.d(TAG, "sum : " + sum);
        //editTextStat1.setText(sum);
    }

    public void onButtonStatCancel(View view) {
        finish();
    }
}

```

15) activity_stats1.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"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Stats">
    <TextView
        android:id="@+id/textViewStat"
        android:layout_width="0dp"
        android:layout_height="50dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="8dp"
        android:text="내역 통계"
        android:textSize="30sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <TextView
        android:id="@+id/textViewStat2"
        android:layout_width="wrap_content"
        android:layout_height="40dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="16dp"
        android:text="총"
        android:textSize="18sp"
        app:layout_constraintEnd_toStartOf="@+id/editTextStat1"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textViewStat" />

    <TextView
```

```

        android:id="@+id/textViewStat3"
        android:layout_width="wrap_content"
        android:layout_height="40dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="8dp"
        android:text="원"
        android:textSize="18sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/editTextStat1"
        app:layout_constraintTop_toBottomOf="@+id/textViewStat" />
    <EditText
        android:id="@+id/editTextStat1"
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:ems="10"
        android:inputType="textPersonName"
        app:layout_constraintEnd_toStartOf="@+id/textViewStat3"
        app:layout_constraintStart_toEndOf="@+id/textViewStat2"
        app:layout_constraintTop_toBottomOf="@+id/textViewStat" />
    <Button
        android:id="@+id/buttonStat2"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="32dp"
        android:onClick="onButtonStatCancel"
        android:text="취소"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.6"
        app:layout_constraintStart_toStartOf="parent" />
</android.support.constraint.ConstraintLayout>

```

```

16) TextFileManager.java
package com.example.win10.myapplication;

import android.content.Context;
import android.os.Bundle;

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class TextFileManager extends MainActivity{
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    private static final String TF = "Memo.txt";
    //내역 내용을 저장할 파일 이름
    Context mContext = null;

    public TextFileManager(Context context) {
        mContext = context;
    }

    //파일에 메모를 저장하는 함수
    public void save(String strData) {
        if (strData == null || strData.equals("")) {
            return;
        }

        FileOutputStream fosMemo = null;

        try {
            //파일에 데이터를 쓰기 위해서 output 스트림 생성

```

```

        fosMemo = mContext.openFileOutput(TF,
Context.MODE_PRIVATE);
        //파일에 내용 적기
        fosMemo.write(strData.getBytes());
    }catch (Exception e) {
        e.printStackTrace();
    }
}
//저장된 내역을 불러오는 함수
public String load() {
    try {
        //파일에서 데이터를 읽기 위해서 input 스트림 생성
        FileInputStream fisMemo = mContext.openFileInput(TF);

        //데이터를 읽어온 뒤 String 타입 객체로 반환
        byte[] memoData = new byte[fisMemo.available()];
        while (fisMemo.read(memoData) != -1) {}

        return new String(memoData);
    }catch (IOException e) {
        return "";
    }
    //저장된 내용을 삭제하는 함수
    //public void delete() {
        // mContext.deleteFile(TF);
    //}
}
}

```

17) colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#008577</color>
    <color name="colorPrimaryDark">#00574B</color>
    <color name="colorAccent">#D81B60</color>
</resources>
```

18) string.xml

```
<resources>
    <string name="app_name">2조 가계부</string>
</resources>
```

19) drawable/ic_launcher_background.xml

```
<?xml version="1.0" encoding="utf-8"?>
<vector
    android:height="108dp"
    android:width="108dp"
    android:viewportHeight="108"
    android:viewportWidth="108"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <path android:fillColor="#008577"
        android:pathData="M0,0h108v108h-108z"/>
    <path android:fillColor="#00000000" android:pathData="M9,0L9,108"
        android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
    <path android:fillColor="#00000000"
        android:pathData="M19,0L19,108"
        android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
    <path android:fillColor="#00000000"
        android:pathData="M29,0L29,108"
        android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
    <path android:fillColor="#00000000"
        android:pathData="M39,0L39,108"
        android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
    <path android:fillColor="#00000000"
        android:pathData="M49,0L49,108"
        android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
    <path android:fillColor="#00000000"
        android:pathData="M59,0L59,108"
        android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
    <path android:fillColor="#00000000"
        android:pathData="M69,0L69,108"
        android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
    <path android:fillColor="#00000000"
        android:pathData="M79,0L79,108"
        android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
    <path android:fillColor="#00000000"
```



```

android:pathData="M89,0L89,108"
    android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
<path android:fillColor="#00000000"
android:pathData="M99,0L99,108"
    android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
<path android:fillColor="#00000000" android:pathData="M0,9L108,9"
    android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
<path android:fillColor="#00000000"
android:pathData="M0,19L108,19"
    android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
<path android:fillColor="#00000000"
android:pathData="M0,29L108,29"
    android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
<path android:fillColor="#00000000"
android:pathData="M0,39L108,39"
    android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
<path android:fillColor="#00000000"
android:pathData="M0,49L108,49"
    android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
<path android:fillColor="#00000000"
android:pathData="M0,59L108,59"
    android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
<path android:fillColor="#00000000"
android:pathData="M0,69L108,69"
    android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
<path android:fillColor="#00000000"
android:pathData="M0,79L108,79"
    android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
<path android:fillColor="#00000000"
android:pathData="M0,89L108,89"
    android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
<path android:fillColor="#00000000"
android:pathData="M0,99L108,99"
    android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>

```

```

        <path android:fillColor="#00000000"
android:pathData="M19,29L89,29"
        android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
        <path android:fillColor="#00000000"
android:pathData="M19,39L89,39"
        android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
        <path android:fillColor="#00000000"
android:pathData="M19,49L89,49"
        android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
        <path android:fillColor="#00000000"
android:pathData="M19,59L89,59"
        android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
        <path android:fillColor="#00000000"
android:pathData="M19,69L89,69"
        android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
        <path android:fillColor="#00000000"
android:pathData="M19,79L89,79"
        android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
        <path android:fillColor="#00000000"
android:pathData="M29,19L29,89"
        android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
        <path android:fillColor="#00000000"
android:pathData="M39,19L39,89"
        android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
        <path android:fillColor="#00000000"
android:pathData="M49,19L49,89"
        android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
        <path android:fillColor="#00000000"
android:pathData="M59,19L59,89"
        android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
        <path android:fillColor="#00000000"
android:pathData="M69,19L69,89"
        android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
        <path android:fillColor="#00000000"

```

```

android:pathData="M79,19L79,89"
    android:strokeColor="#33FFFFFF" android:strokeWidth="0.8"/>
</vector>

```

```

20) drawable/ic_launcher_background.xml(v24)
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:aapt="http://schemas.android.com/aapt"
    android:width="108dp"
    android:height="108dp"
    android:viewportWidth="108"
    android:viewportHeight="108">
    <path
        android:fillType="evenOdd"
        android:pathData="M32,64C32,64 38.39,52.99
44.13,50.95C51.37,48.37 70.14,49.57
70.14,49.57L108.26,87.69L108,109.01L75.97,107.97L32,64Z"
        android:strokeWidth="1"
        android:strokeColor="#00000000">
        <aapt:attr name="android:fillColor">
            <gradient
                android:endX="78.5885"
                android:endY="90.9159"
                android:startX="48.7653"
                android:startY="61.0927"

```

```

        android:type="linear">
        <item
            android:color="#44000000"
            android:offset="0.0" />
        <item
            android:color="#00000000"
            android:offset="1.0" />
    </gradient>
</aapt:attr>
</path>
<path
    android:fillColor="#FFFFFF"
    android:fillType="nonZero"
    android:pathData="M66.94,46.02L66.94,46.02C72.44,50.07
76,56.61 76,64L32,64C32,56.61 35.56,50.11
40.98,46.06L36.18,41.19C35.45,40.45 35.45,39.3
36.18,38.56C36.91,37.81 38.05,37.81
38.78,38.56L44.25,44.05C47.18,42.57 50.48,41.71
54,41.71C57.48,41.71 60.78,42.57
63.68,44.05L69.11,38.56C69.84,37.81 70.98,37.81
71.71,38.56C72.44,39.3 72.44,40.45
71.71,41.19L66.94,46.02ZM62.94,56.92C64.08,56.92 65,56.01
65,54.88C65,53.76 64.08,52.85 62.94,52.85C61.8,52.85 60.88,53.76
60.88,54.88C60.88,56.01 61.8,56.92
62.94,56.92ZM45.06,56.92C46.2,56.92 47.13,56.01
47.13,54.88C47.13,53.76 46.2,52.85 45.06,52.85C43.92,52.85
43,53.76 43,54.88C43,56.01 43.92,56.92 45.06,56.92Z"
    android:strokeWidth="1 "
    android:strokeColor="#00000000" />
</vector>

```

```
21) mipmap/ic_launcher.xml
<?xml version="1.0" encoding="utf-8"?>
<adaptive-icon
xmlns:android="http://schemas.android.com/apk/res/android">
    <background
android:drawable="@drawable/ic_launcher_background"/>
    <foreground android:drawable="@mipmap/ic_launcher_foreground"/>
</adaptive-icon>
```

```
22) mipmap/ic_launcher_round.xml
<?xml version="1.0" encoding="utf-8"?>
<adaptive-icon
xmlns:android="http://schemas.android.com/apk/res/android">
    <background
android:drawable="@drawable/ic_launcher_background"/>
    <foreground android:drawable="@mipmap/ic_launcher_foreground"/>
</adaptive-icon>
```

23) Gradle Scripts/build.gradle (Project: HouseholdLedger1)

// Top-level build file where you can add configuration options common to all sub-projects/modules.

```
buildscript {

    repositories {
        google()
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:3.4.0'

        // NOTE: Do not place your application dependencies here;
        // they belong
        // in the individual module build.gradle files
    }
}

allprojects {
    repositories {
        google()
        jcenter()
    }
}

task clean(type: Delete) {
    delete rootProject.buildDir
}
```

```

24) Gradle Scripts/build.gradle (Module: app)
apply plugin: 'com.android.application'

android {
    compileSdkVersion 28
    defaultConfig {
        applicationId "com.example.win10.myapplication"
        minSdkVersion 28
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner
            "android.support.test.runner.AndroidJUnitRunner"

        android {
            lintOptions {

                checkReleaseBuilds false

                // Or, if you prefer, you can continue to check for
                errors in release builds,

                // but continue the build even when errors are found:

                abortOnError false
            }
        }
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'),
                'proguard-rules.pro'
        }
    }
}

```



```

    }
  }
}

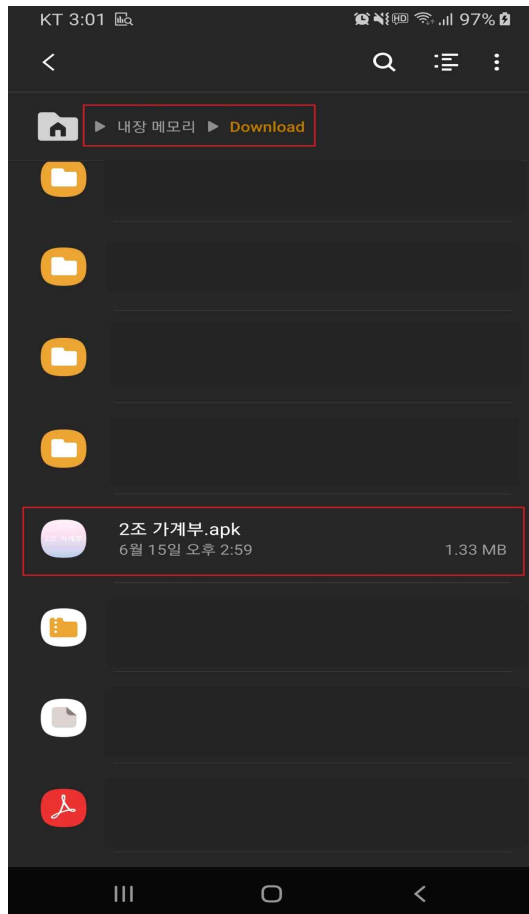
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation
'com.android.support.constraint:constraint-layout:1.1.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation
'com.android.support.test.espresso:espresso-core:3.0.2'
}

```

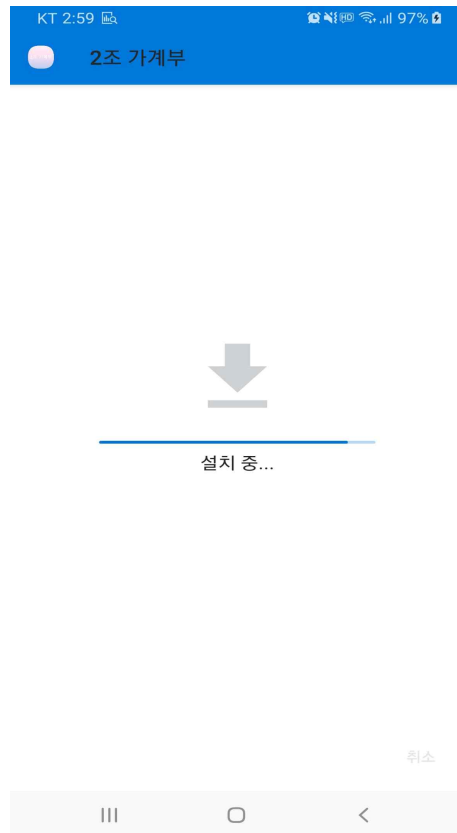
11-2. 2조 가계부 앱 사용설명서

ㄱ. 스마트폰에 APK 파일 설치

1) 스마트폰의 지정 경로에 .apk 파일 다운로드를 한다.



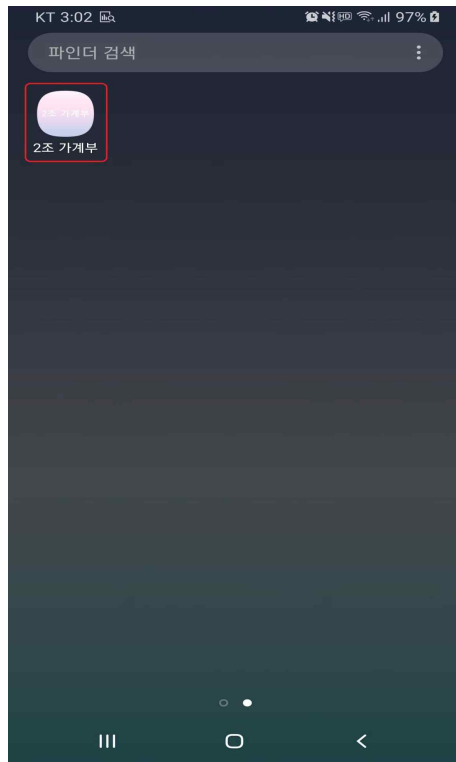
2) 다운로드 후 .apk 파일을 설치한다.



3) 설치 후 완료 또는 열기를 클릭한다.



- 4) 완료를 클릭한 후 앱 목록을 보면 2조 가게부 앱 설치가 정상적으로 완료된 것을 확인할 수 있다.



ㄴ. 설치한 APK 파일 실행

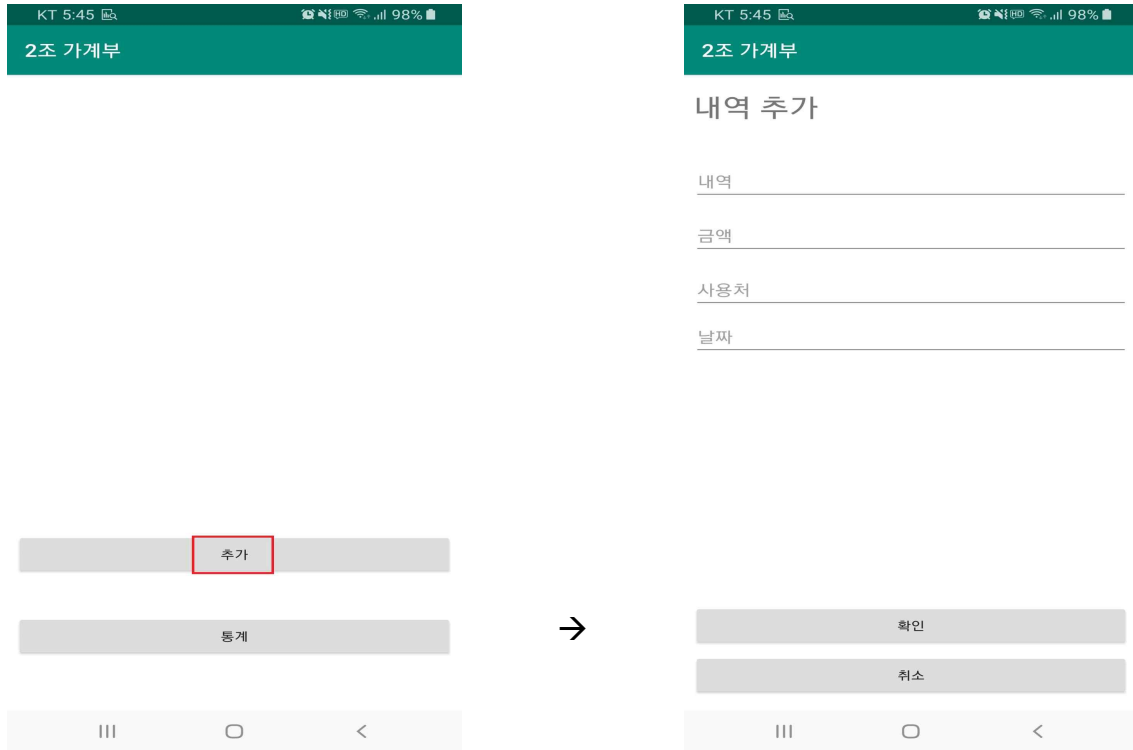
1) 실행을 하면 로딩 화면이 나오게 된다.



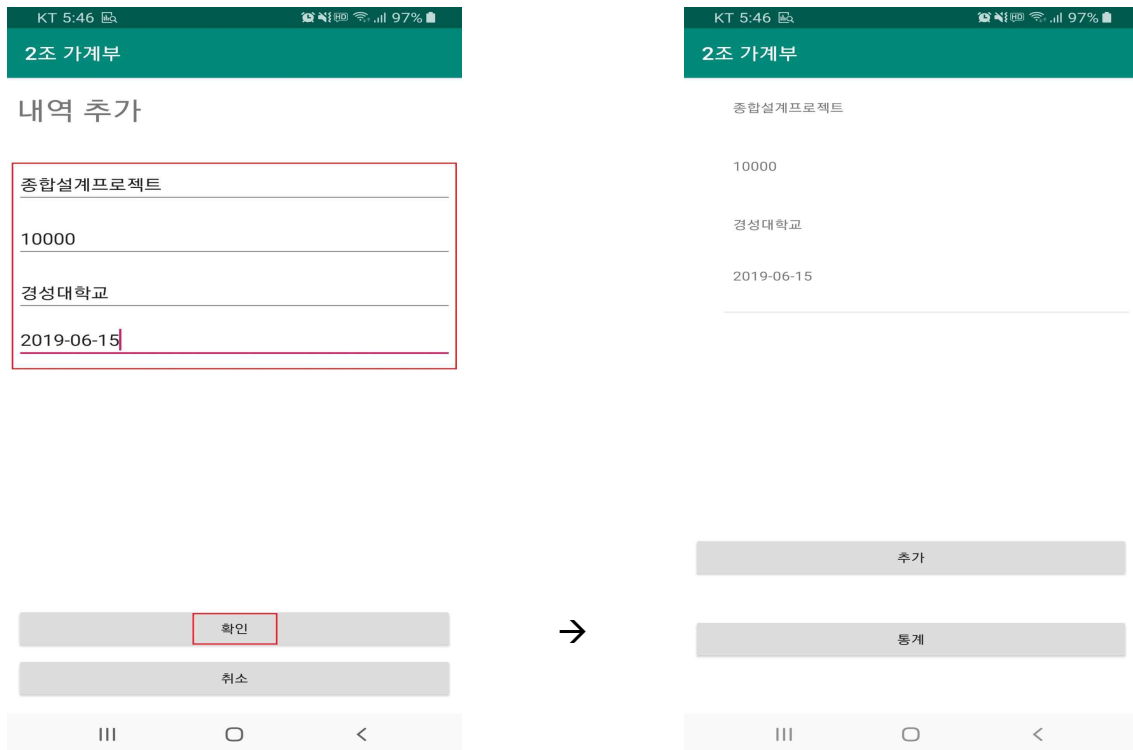
2) 추가할 내역이 있으면 추가 버튼을 클릭한다.



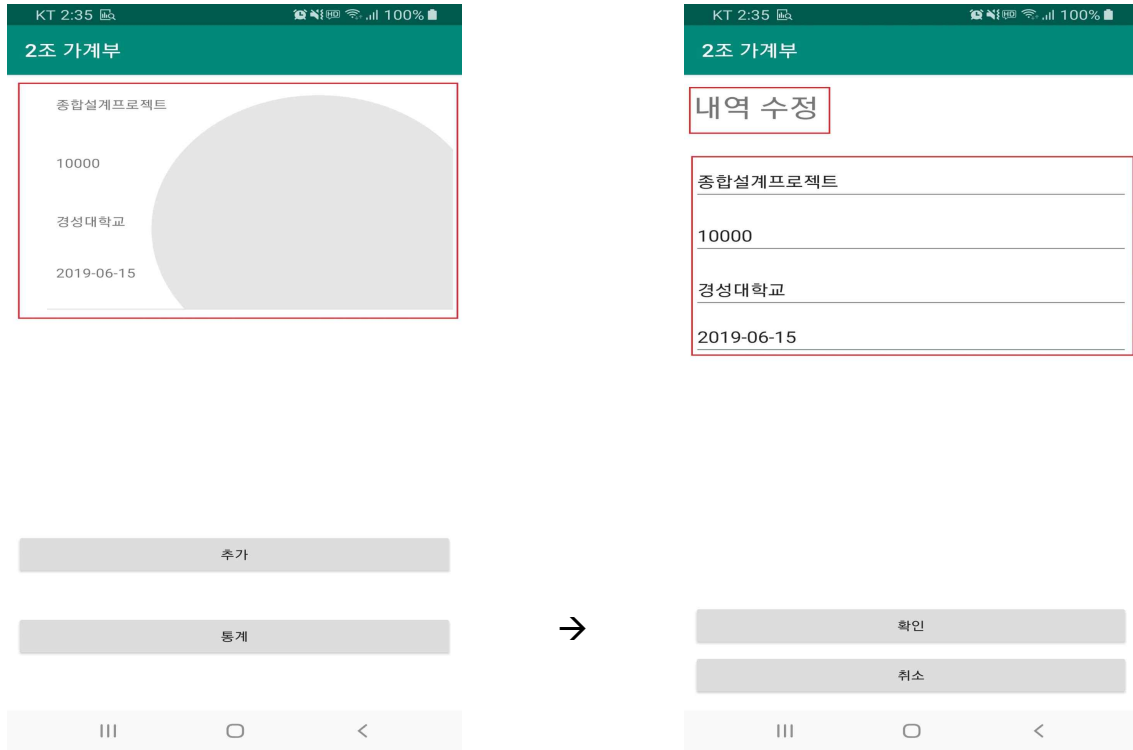
3) 추가 버튼 클릭하면 내역 추가 페이지로 넘어간다.



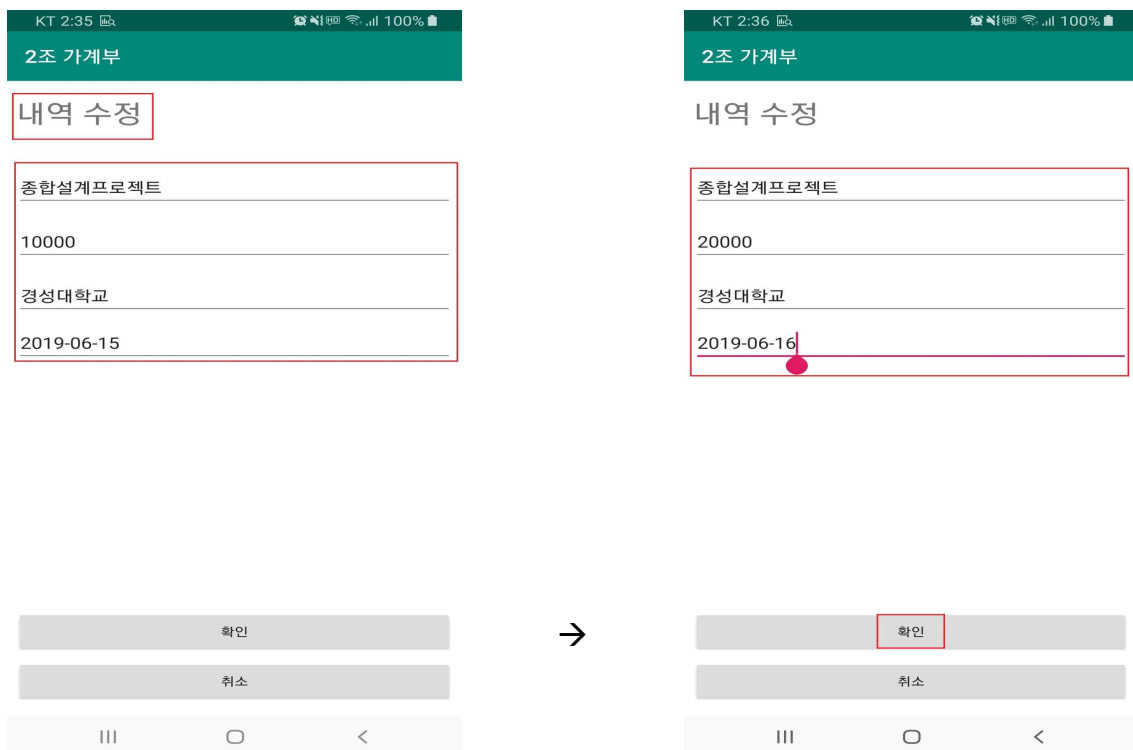
4) 추가할 내역을 입력 후 확인 버튼을 누르면 입력한 내역이 메인 화면에 출력된다.



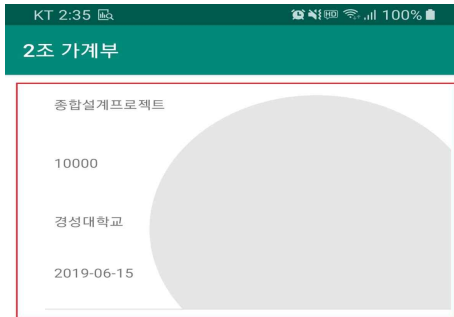
5) 입력된 내역을 짧게 한 번 누르면 수정 페이지로 이동한다.



6) 내역을 수정 후 확인 버튼을 누른다.



7) 내역을 길게 한 번 클릭하면 내역이 삭제된다.

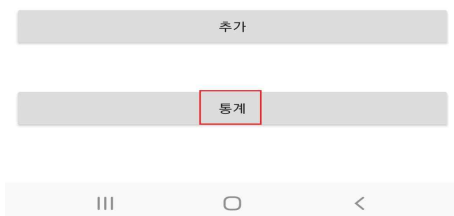


8) 통계 버튼을 누르면 통계 페이지로 이동한다.



내역 통계

총 _____ 원



9) 통계 페이지로 이동하는 동시에 빈 칸에 누적합계가 출력된다.

KT 5:46 97%

2조 가계부

내역 통계

총 원

취소

III O <

11-3. 프로젝트 설치 환경

1) 사용한 기기

- 노트북

2) 운영체제

- Windows10

3) 사용한 프로그램 종류

- Android Studio 4.3
- Android Studio 가상 에뮬레이터(Nexus 4 API 28)

11-4. 프로젝트 결과 시연 환경

1) 사용한 기기

- 갤럭시노트9

2) 운영체제

- Android PIE (Android 9)

11-5. 참고 자료

1) 네이버 가계부

- <https://moneybook.naver.com>

2) 블로그

- <https://yongtech.tistory.com/100>
- <https://yoo-hyeok.tistory.com/31>
- <https://googry.tistory.com/24>
- <https://coding-factory.tistory.com/50>
- <https://apphappy.tistory.com/88>
- <https://1hk-star.tistory.com/11>

3) 안드로이드 프로그래밍 수업 자료