

**# 6. 실습 보고서**  
**- 에디트텍스트와 컴파운드버튼 실습 -**

보 고 자 성 명	20181151 김 지 원
보 고 자 학 과	산 업 경 영 공 학 과
담 당 교 수 성 명	정 진 만 교 수 님
과 목 명	모 바 일 프 로 그 래 밍 01분 반
제 출 날 짜	2020.04.26

- 목차 -

1. 서론	3p
1) 개요	
2) 요청사항 분석	
3) 추가사항	
4) 초기 디자인	
2. 본론	
1) activity_main.xml	4p
2) MainActivity.java	
① 소스 코드 원본	5p
② 소스 코드 분석	10p
3) 결과 화면	16p
3. 결론	
1) 개인정보 확인 앱 장점	18p
2) 개인정보 확인 앱 단점	19p
3) 배운 점	19p

## 1. 서론

### 1) 개요

단순히 주민등록번호를 확인하는 앱이 아닌 주제를 가지고 앱을 제작하였다. 최근 코로나 바이러스로 인해 학교 건물에 출입 시 자신의 개인정보를 적어야 하는 경우가 간혹 있다. 이를 수기로 적어 제출하는 것이 아닌 앱을 이용하여 개인정보를 제출하는 형식의 Entry Log 프로젝트를 제작하였다.

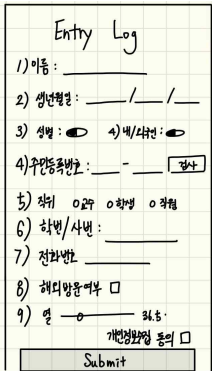
### 2) 요청사항 분석

- ① Radio Button : 출입하는 본인이 교수, 학생, 직원 중 어느 부류에 속하는지 선택하는 Radio Button 제작함
- ② Switch : 출입하는 본인의 성별과 내국인 혹은 외국인을 선택하는 항목을 Switch로 제작함
- ③ 데이트피커 : 출입하는 본인의 생년월일을 데이트피커로 입력할 수 있게 제작함
- ④ 주민등록번호 유효성 검증 : 주민등록번호 입력 후 유효성 검사 Button을 클릭하면 Toast로 유효성 여부를 출력함
- ⑤ 개발자 정보 출력 : 어플리케이션 시작과 동시에 Toast로 학번과 이름을 출력함

### 3) 추가사항

- ① Check Box : 해외 방문 여부와 개인정보 수집 동의에 대한 정보를 Check Box로 받음
- ② Seek Bar : 열을 측정하여 해당 정보를 Seek Bar로 입력함

### 4) 초기 디자인

	<ul style="list-style-type: none"><li>- 상단에 어플리케이션의 이름은 출력한다.</li><li>- 이름, 생년월일, 성별, 내/외국인, 주민등록번호, 학위, 학번, 전화번호, 해외 방문 여부, 열, 개인정보 수집 동의 순서로 위젯을 배치한다.</li><li>- 어플리케이션의 가장 하단에는 제출 Button을 배치한다.</li></ul>
---	---

## 2. 본문

### 1) activity\_main.xml

Component Tree

- LinearLayout (vertical)
  - Ab textViewTP "Entry Log"
  - ▼ LinearLayoutNM (horizontal)
    - Ab textView2 "1) 이름(Name) : "
    - Ab editTextNM
  - ▼ LinearLayoutDB (horizontal)
    - Ab textView4 "2) 생년월일(Date..."
    - Ab textViewDB "Click!"
  - ▼ LinearLayoutSX (horizontal)
    - Ab textView5 "3) 성별(Sex) : "
    - Ab switchSX
    - Ab textViewSX "Male"
  - ▼ LinearLayoutLF (horizontal)
    - Ab textView6 "4) 내/외국인(Loc..."
    - Ab switchLF
    - Ab textViewLF "내국인(Local)"
  - ▼ LinearLayoutRR (horizontal)
    - Ab textView7 "5) 주민등록번호(..."
  - ▼ LinearLayoutRR2 (horizontal)
    - Ab textViewRR1
    - Ab editTextRR2 (Number)
    - Ab buttonRR "유효성 검사(Che..."
  - ▼ LinearLayoutPS (horizontal)
    - Ab textView9 "6) 구분(Position)..."
    - ▼ RadioGroupPS (horizontal)
      - radioButton "교수(Prof)"
      - radioButton2 "학생(Std)"
      - radioButton3 "직원(Staff)"
  - ▼ LinearLayoutID (horizontal)
    - Ab textView10 "7) 사번/학번(ID..."
    - Ab editTextID (Number)
  - ▼ LinearLayoutPN (horizontal)
    - Ab textView14 "8) 전화번호(Ph..."
    - Ab editTextPN (Phone)
  - ▼ LinearLayoutAB (horizontal)
    - Ab textView15 "9) 해외 방문 여..."
    - Ab checkBoxAB
  - ▼ LinearLayoutFV (horizontal)
    - Ab textView16 "10) 열(Fever) : "
    - Ab seekBarFV
    - Ab textViewFV "36.5°C"
  - ▼ LinearLayoutAG (horizontal)
    - Ab textView18 "개인 정보 수집 ..."
    - Ab checkBoxAG
  - ▼ LinearLayoutSB (horizontal)
    - Ab buttonSB "제출(Submit)"

- vertical Linear Layout에 1개의 text View와 13개의 horizontal Linear Layout을 삽입한다.
- 각 Layout에 에디트 텍스트와 컴파운드 버튼을 삽입한다.
- 모든 위젯은 구분이 용이하도록 padding을 5dp 부여하여 위젯 사이의 간격을 조정한다.

## 2) MainActivity.java

### ① 소스 코드 원본

```
package com.example.week6_entrylog;

import androidx.appcompat.app.AppCompatActivity;

import android.app.DatePickerDialog;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.CompoundButton;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.RadioGroup;
import android.widget.SeekBar;
import android.widget.Switch;
import android.widget.TextView;
import android.widget.Toast;

import java.util.Calendar;
import java.util.TimeZone;

public class MainActivity extends AppCompatActivity {

    int year_check;
    int sex_check;
    int lf_check;
    int rr_check;

    EditText et_NM;

    TextView tv_DB;
    DatePickerDialog DP;

    Switch sw_SX;
    TextView tv_SX;

    Switch sw_LF;
    TextView tv_LF;

    TextView tv_RR1;
    EditText et_RR2;
    Button btn_RR;

    RadioGroup rg_PS;
    EditText et_ID;
    EditText et_PN;
    CheckBox cb_AB;
    SeekBar sb_FV;
    TextView tv_FV;
    CheckBox cb_AG;
    Button btn_SB;

    public void onBtnCK(View v)
    {
        String input1 = tv_RR1.getText().toString();
        String input2 = et_RR2.getText().toString();

        int first = Integer.parseInt(input1.substring(0,1));
        int second = Integer.parseInt(input1.substring(1,2));
        int third = Integer.parseInt(input1.substring(2,3));
        int fourth = Integer.parseInt(input1.substring(3,4));
        int fifth = Integer.parseInt(input1.substring(4,5));
        int sixth = Integer.parseInt(input1.substring(5,6));
        int seventh = Integer.parseInt(input1.substring(27,28));
        int eighth = Integer.parseInt(input2.substring(0,1));
        int ninth = Integer.parseInt(input2.substring(1,2));
        int tenth = Integer.parseInt(input2.substring(2,3));
    }
}
```

```

int eleventh = Integer.parseInt(input2.substring(3,4));
int twelfth = Integer.parseInt(input2.substring(4,5));
int Thirteenth = Integer.parseInt(input2.substring(5,6));

int sum = first*2 + second*3 + third*4 + fourth*5 + fifth*6 + sixth*7 +
seventh*8 + eighth*9 + ninth*2 + tenth*3 + eleventh*4 + twelfth*5;
int check = (11 - (sum % 11)) % 10;

if(check==Thirteenth)
{
rr_check = 1;
Toast.makeText(getApplicationContext(),"주민등록번호가 유효합니다.(Valid)",Toast.LENGTH_LONG).show();
}
else if(check!=Thirteenth)
{
rr_check = 0;
Toast.makeText(getApplicationContext(),"주민등록번호가 유효하지 않습니다.(Invalid)",Toast.LENGTH_LONG).show();
}
}

public void onBtnSB(View v)
{
if(rr_check == 1 && cb_AG.isChecked() == true )
{
Toast.makeText(getApplicationContext(),"제출되었습니다.",Toast.LENGTH_LONG).show();
}
else if(cb_AG.isChecked() == false )
{
Toast.makeText(getApplicationContext(),"개인정보 수집동의 항목은\n"+"필수 체크 항목입니다.",Toast.LENGTH_LONG).show();
}
else if(rr_check == 0 )
{
Toast.makeText(getApplicationContext(),"주민등록번호가 유효하지 않습니다.\n"+"재입력해주세요.",Toast.LENGTH_LONG).show();
}
}

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

Toast.makeText(getApplicationContext(),"20181151
김지원",Toast.LENGTH_LONG).show();

et_NM = findViewById(R.id.editTextNM);

tv_DB = findViewById(R.id.textViewDB);
final Calendar c = Calendar.getInstance();
c.setTimeZone(TimeZone.getTimeZone("Asia/Seoul"));
final int d = c.get(Calendar.DAY_OF_MONTH);
final int m = c.get(Calendar.MONTH);
final int y = c.get(Calendar.YEAR);
tv_DB.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
Calendar c = Calendar.getInstance();
c.setTimeZone(TimeZone.getTimeZone("Asia/Seoul"));
int d = c.get(Calendar.DAY_OF_MONTH);
int m = c.get(Calendar.MONTH);
int y = c.get(Calendar.YEAR);
DP = new DatePickerDialog(MainActivity.this, new
DatePickerDialog.OnDateSetListener() {
@Override
public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {
tv_DB.setText(""+year+"/"+(month+1)+"/"+dayOfMonth);
if(year<2000 && (month+1)<10 && dayOfMonth<10)
{
tv_RR1.setText(""+(year-1900)+"0"+(month+1)+"0"+dayOfMonth+"
-
");
}
}
else if(year<2000 && (month+1)<10 && dayOfMonth>=10)
{
}
}
}

```

```

tv_RR1.setText(""+(year-1900)+"0"+(month+1)+dayOfMonth+"
");
}
else if(year<2000 && (month+1)>=10 && dayOfMonth<10)
{
tv_RR1.setText(""+(year-1900)+(month+1)+"0"+dayOfMonth+"
");
}
else if(year<2000 && (month+1)>=10 && dayOfMonth>=10)
{
tv_RR1.setText(""+(year-1900)+(month+1)+dayOfMonth+"
");
}

else if(year>=2000 && (month+1)<10 && dayOfMonth<10)
{
tv_RR1.setText(""+(year-2000)+"0"+(month+1)+"0"+dayOfMonth+"
");
}
else if(year>=2000 && (month+1)<10 && dayOfMonth>=10)
{
tv_RR1.setText(""+(year-2000)+"0"+(month+1)+dayOfMonth+"
");
}
else if(year>=2000 && (month+1)>=10 && dayOfMonth<10)
{
tv_RR1.setText(""+(year-2000)+(month+1)+"0"+dayOfMonth+"
");
}
else if(year>=2000 && (month+1)>=10 && dayOfMonth>=10)
{
tv_RR1.setText(""+(year-2000)+(month+1)+dayOfMonth+"
");
}

if(year<2000)
{
year_check = 0;
}
else if(year>=2000)
{
year_check = 1;
}
}, y, m, d);
DP.show();
});

sw_SX = findViewById(R.id.switchSX);
tv_SX = findViewById(R.id.textViewSX);
sw_SX.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
@Override
public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
boolean SX = sw_SX.isChecked();
if(SX == true)
{
tv_SX.setText(""+"Female");
sex_check = 1;
}
else
{
tv_SX.setText(""+"Male");
sex_check = 0;
}
}
});

sw_LF = findViewById(R.id.switchLF);
tv_LF = findViewById(R.id.textViewLF);
sw_LF.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
@Override

```

```

public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
    boolean LF = sw_LF.isChecked();
    if(LF == true)
    {
        tv_LF.setText(""+외국인(Foreigner));
        lf_check = 1;
    }
    else
    {
        tv_LF.setText(""+내국인(Local));
        lf_check = 0;
    }
}
});

tv_RR1 = findViewById(R.id.textViewRR1);
et_RR2 = findViewById(R.id.editTextRR2);
btn_RR = findViewById(R.id.buttonRR);
tv_RR1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(tv_RR1.length() == 28)
        {
            String str = tv_RR1.getText().toString();
            tv_RR1.setText(str.substring(0,27));
        }
        if(lf_check == 1)
        {
            if(year_check==0 && sex_check==0)
            {
                tv_RR1.append(""+5);
            }
            else if(year_check==0 && sex_check==1)
            {
                tv_RR1.append(""+6);
            }
            else if(year_check==1 && sex_check==0)
            {
                tv_RR1.append(""+7);
            }
            else if(year_check==1 && sex_check==1)
            {
                tv_RR1.append(""+8);
            }
        }
        else if(lf_check == 0)
        {
            if(year_check==0 && sex_check==0)
            {
                tv_RR1.append(""+1);
            }
            else if(year_check==0 && sex_check==1)
            {
                tv_RR1.append(""+2);
            }
            else if(year_check==1 && sex_check==0)
            {
                tv_RR1.append(""+3);
            }
            else if(year_check==1 && sex_check==1)
            {
                tv_RR1.append(""+4);
            }
        }
    }
});

rg_PS = findViewById(R.id.RadioGroupPS);
et_ID = findViewById(R.id.editTextID);

```



```

et_PN = findViewById(R.id.editTextPN);
cb_AB = findViewById(R.id.checkBoxAB);

sb_FV = findViewById(R.id.seekBarFV);
tv_FV = findViewById(R.id.textViewFV);
sb_FV.setMax(50);
sb_FV.setProgress(15);
sb_FV.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
@Override
public void onProgressChanged(SeekBar seekBar, int progress, boolean
fromUser) {
tv_FV.setText(""+((progress)*0.1+35)+"°C");
}

@Override
public void onStartTrackingTouch(SeekBar seekBar) {
}

@Override
public void onStopTrackingTouch(SeekBar seekBar) {
}
});

cb_AG = findViewById(R.id.checkBoxAG);
btn_SB = findViewById(R.id.buttonSB);

}
}

```

## ② 소스 코드 분석

```
package com.example.week6_entrylog;

import androidx.appcompat.app.AppCompatActivity;

import android.app.DatePickerDialog;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.CompoundButton;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.RadioGroup;
import android.widget.SeekBar;
import android.widget.Switch;
import android.widget.TextView;
import android.widget.Toast;

import java.util.Calendar;
import java.util.TimeZone;

public class MainActivity extends AppCompatActivity {

    int year_check;
    int sex_check;
    int lf_check;
    int rr_check;

    EditText et_NM;
    TextView tv_DB;
    DatePickerDialog DP;
    Switch sw_SX;
    TextView tv_SX;
    Switch sw_LF;
    TextView tv_LF;
    TextView tv_RR1;
    EditText et_RR2;
    Button btn_RR;
    RadioGroup rg_PS;
    EditText et_ID;
    EditText et_PN;
    CheckBox cb_AB;
    SeekBar sb_FV;
    TextView tv_FV;
    CheckBox cb_AG;
    Button btn_SB;
```

### 1. Import & Main Activity class

- year\_check, sex\_check, lf\_check 즉, 출생 연도, 성별, 내외국인을 선언하여 주민등록번호 자동입력 시 상태를 확인한다.
- rr\_check으로 주민등록번호 유효성 검사에 통과하였는지의 상태를 확인하여 마지막에 제출 가능한지를 판단한다.
- 어플리케이션에 삽입되는 다양한 위젯들을 선언한다.

```

public void onBtnCK(View v)
{
    String input1 = tv_RR1.getText().toString();
    String input2 = et_RR2.getText().toString();

    int first = Integer.parseInt(input1.substring(0,1));
    int second = Integer.parseInt(input1.substring(1,2));
    int third = Integer.parseInt(input1.substring(2,3));
    int fourth = Integer.parseInt(input1.substring(3,4));
    int fifth = Integer.parseInt(input1.substring(4,5));
    int sixth = Integer.parseInt(input1.substring(5,6));
    int seventh = Integer.parseInt(input1.substring(7,8));
    int eighth = Integer.parseInt(input2.substring(0,1));
    int ninth = Integer.parseInt(input2.substring(1,2));
    int tenth = Integer.parseInt(input2.substring(2,3));
    int eleventh = Integer.parseInt(input2.substring(3,4));
    int twelfth = Integer.parseInt(input2.substring(4,5));
    int Thirteenth = Integer.parseInt(input2.substring(5,6));

    int sum = first*2 + second*3 + third*4 + fourth*5 + fifth*6 + sixth*7 + seventh*8 + eighth*9 + ninth*2 + tenth*3 + eleventh*4 + twelfth*5;
    int check = (11 - (sum % 11)) % 10;

    if(check==Thirteenth)
    {
        rr_check = 1;
        Toast.makeText(getApplicationContext(), text: "주민등록번호가 유효합니다.(Valid)", Toast.LENGTH_LONG).show();
    }
    else if(check!=Thirteenth)
    {
        rr_check = 0;
        Toast.makeText(getApplicationContext(), text: "주민등록번호가 유효하지 않습니다.(Invalid)", Toast.LENGTH_LONG).show();
    }
}

```

## 2. onBtnCK method

- 해당 Button이 눌리면 주민등록번호의 유효성 검사를 실시한다.
- 주민등록번호의 각 숫자에 2, 3, 4, 5, 6, 7, 8, 9, 2, 3, 4, 5를 곱한 후 11로 나눈 나머지를 11에서 뺀다. 그 결과 값의 일의 자리숫자와 주민등록번호의 마지막 숫자와 일치하면 유효한 주민등록번호이다.
- 주민등록번호가 유효하면 rr\_check를 1로 전환하고, 주민등록번호가 유효하다는 Toast Message를 4초간 출력한다.
- 주민등록번호가 유효하지 않다면 rr\_check를 0으로 전환하고, 주민등록번호가 유효하지 않다는 Toast Message를 출력한다.

```

public void onBtnSB(View v)
{
    if(rr_check == 1 && cb_AG.isChecked() == true )
    {
        Toast.makeText(getApplicationContext(), text: "제출되었습니다.", Toast.LENGTH_LONG).show();
    }
    else if(cb_AG.isChecked() == false )
    {
        Toast.makeText(getApplicationContext(), text: "개인정보 수집동의 항목은\n"+"필수 체크 항목입니다.", Toast.LENGTH_LONG).show();
    }
    else if(rr_check == 0 )
    {
        Toast.makeText(getApplicationContext(), text: "주민등록번호가 유효하지 않습니다.\n"+"재입력해주세요.", Toast.LENGTH_LONG).show();
    }
}

```

## 3. onBtnSB

- 해당 Button이 눌리면 유저가 작성한 정보가 제출된다.
- 주민등록번호의 유효성 검사를 통과하고 개인 정보 수집에 동의하면 Toast Message로 “제출되었습니다”가 출력된다.
- 개인 정보 수집에 동의하지 않는다면 Toast Message로 “개인정보 수집동의 항목은 필수 체크 항목입니다.”가 출력된다.
- 주민등록번호의 유효성 검사를 통과하지 못한 상태로 제출하면 Toast Message로 “주민등록번호가 유효하지 않습니다. 재입력해주세요.”가 출력된다.

```

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

    Toast.makeText(getApplicationContext(), text: "20181151 김지원", Toast.LENGTH_LONG).show();
}

```

## 4. Oncreate method

- 어플리케이션에서 가장 먼저 실행되는 method로 하위에 나오는 모든 위젯들이 작동된다.
- 어플리케이션을 시작하자마자 Toast Message로 학번과 이름을 4초 동안 출력한다.

```
et_NM = findViewById(R.id.editTextNM);
```

## 5. NM ( Name )

- et\_NM으로 이름을 쓸 수 있는 editText를 활성화시킨다.

```
tv_DB = findViewById(R.id.textViewDB);
final Calendar c = Calendar.getInstance();
c.setTimeZone(TimeZone.getTimeZone("Asia/Seoul"));
final int d = c.get(Calendar.DAY_OF_MONTH);
final int m = c.get(Calendar.MONTH);
final int y = c.get(Calendar.YEAR);
tv_DB.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Calendar c = Calendar.getInstance();
        c.setTimeZone(TimeZone.getTimeZone("Asia/Seoul"));
        int d = c.get(Calendar.DAY_OF_MONTH);
        int m = c.get(Calendar.MONTH);
        int y = c.get(Calendar.YEAR);

        DP = new DatePickerDialog( context: MainActivity.this, new DatePickerDialog.OnDateSetListener() {
            @Override
            public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {
                tv_DB.setText(""+year+"/"+(month+1)+"/"+dayOfMonth);
                if(year<2000 && (month+1)<10 && dayOfMonth<10)
                {
                    tv_RR1.setText(""+(year-1900)+"0"+(month+1)+"0"+dayOfMonth+" - ");
                }
                else if(year<2000 && (month+1)<10 && dayOfMonth>=10)
                {
                    tv_RR1.setText(""+(year-1900)+"0"+(month+1)+dayOfMonth+" - ");
                }
                else if(year<2000 && (month+1)>=10 && dayOfMonth<10)
                {
                    tv_RR1.setText(""+(year-1900)+(month+1)+"0"+dayOfMonth+" - ");
                }
                else if(year<2000 && (month+1)>=10 && dayOfMonth>=10)
                {
                    tv_RR1.setText(""+(year-1900)+(month+1)+dayOfMonth+" - ");
                }

                else if(year>=2000 && (month+1)<10 && dayOfMonth<10)
                {
                    tv_RR1.setText(""+(year-2000)+"0"+(month+1)+"0"+dayOfMonth+" - ");
                }
                else if(year>=2000 && (month+1)<10 && dayOfMonth>=10)
                {
                    tv_RR1.setText(""+(year-2000)+"0"+(month+1)+dayOfMonth+" - ");
                }
                else if(year>=2000 && (month+1)>=10 && dayOfMonth<10)
                {
                    tv_RR1.setText(""+(year-2000)+(month+1)+"0"+dayOfMonth+" - ");
                }
                else if(year>=2000 && (month+1)>=10 && dayOfMonth>=10)
                {
                    tv_RR1.setText(""+(year-2000)+(month+1)+dayOfMonth+" - ");
                }

                if(year<2000)
                {
                    year_check = 0;
                }
                else if(year>=2000)
                {
                    year_check = 1;
                }
            }
        }, y, m, d);
        DP.show();
    }
});
```

## 6. DB ( 생년월일 )

- tv\_DB으로 생년월일을 출력할 수 있는 textView를 활성화 시킨다.
- 생년월일은 데이트피커를 활용하여 작성한다.
- 날짜를 선택한 후 해당 textView에 year/month/day 형태로 출력된다.
- 생년월일이 선택되면 주민등록번호의 앞의 6자리가 자동으로 입력된다. ( ex)1998년 10월 3일 생 → 981003)
- 주민등록번호 뒤의 첫 번째 숫자를 자동입력하기 위해 출생년도 확인하는 year\_check를 2000년을 기준으로 0과 1로 지정한다.

```

sw_SX = findViewById(R.id.switchSX);
tv_SX = findViewById(R.id.textViewSX);
sw_SX.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        boolean SX = sw_SX.isChecked();
        if(SX == true)
        {
            tv_SX.setText("+"+"Female");
            sex_check = 1;
        }
        else
        {
            tv_SX.setText("+"+"Male");
            sex_check = 0;
        }
    }
});

```

## 7. SX ( 성별 )

- sw\_SX로 성별을 입력하는 switch와 tv\_SX로 성별을 출력할 수 있는 textView를 활성화시킨다.
- switch가 OFF상태이면 textView에 Male이 출력되고, ON 상태이면 textView에 Female이 출력된다.
- 주민등록번호 뒤의 첫 번째 숫자를 자동입력하기 위해 성별을 확인하는 sex\_check를 남녀를 기준으로 0과 1로 지정한다.

```

sw_LF = findViewById(R.id.switchLF);
tv_LF = findViewById(R.id.textViewLF);
sw_LF.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        boolean LF = sw_LF.isChecked();
        if(LF == true)
        {
            tv_LF.setText("+"+"외국인(Foreigner)");
            lf_check = 1;
        }
        else
        {
            tv_LF.setText("+"+"내국인(Local)");
            lf_check = 0;
        }
    }
});

```

## 8. LF ( 내외국인 )

- sw\_LF로 내외국인을 입력하는 switch와 tv\_LF로 내외국인을 출력할 수 있는 textView를 활성화시킨다.
- switch가 OFF상태이면 textView에 내국인(Local)이 출력되고, ON 상태이면 textView에 외국인(Foreigner)이 출력된다.
- 주민등록번호 뒤의 첫 번째 숫자를 자동입력하기 위해 내외국인지를 확인하는 lf\_check를 내국인인지 외국인지를 기준으로 0과 1로 지정한다.

```

tv_RR1 = findViewById(R.id.textViewRR1);
et_RR2 = findViewById(R.id.editTextRR2);
btn_RR = findViewById(R.id.buttonRR);
tv_RR1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(tv_RR1.length() == 28)
        {
            String str = tv_RR1.getText().toString();
            tv_RR1.setText(str.substring(0,27));
        }
        if(lf_check == 1)
        {
            if(year_check==0 && sex_check==0)
            {
                tv_RR1.append(""+5);
            }
            else if(year_check==0 && sex_check==1)
            {
                tv_RR1.append(""+6);
            }
            else if(year_check==1 && sex_check==0)
            {
                tv_RR1.append(""+7);
            }
            else if(year_check==1 && sex_check==1)
            {
                tv_RR1.append(""+8);
            }
        }
        else if(lf_check == 0)
        {
            if(year_check==0 && sex_check==0)
            {
                tv_RR1.append(""+1);
            }
            else if(year_check==0 && sex_check==1)
            {
                tv_RR1.append(""+2);
            }
            else if(year_check==1 && sex_check==0)
            {
                tv_RR1.append(""+3);
            }
            else if(year_check==1 && sex_check==1)
            {
                tv_RR1.append(""+4);
            }
        }
    }
});

```

#### 9. RR( 주민등록번호 )

- tv\_RR1으로 주민등록번호 중 생년월일과 성별 구별을 하는 7번째 자리까지 자동 입력되는 부분을 textView로 나타낼 수 있게 하고, tv\_RR2로 주민등록번호 중 뒤에 6자리는 직접 유저가 입력할 수 있도록 하는 부분인 editText로 활성화시킨다.
- btn\_RR로 주민등록번호의 유효성 검사를 실시할 수 있는 Button을 활성화시켜 onBtnCK의 항목이 실행된다.
- tv\_RR1을 클릭하면 주민 번호의 7자리의 숫자가 앞서 작성한 생년월일, 성별, 내외국인 정보를 비교하여 자동 입력된다.
- 1900년대 태어난 내국인 남자 : 1 / 1900년대 태어난 내국인 여자 : 2 / 2000년대에 태어난 내국인 남자 : 3 / 2000년대에 태어난 내국인 여자 : 4 / 1900년대에 태어난 외국인 남자 : 5 / 1900년대에 태어난 외국인 여자 : 6 / 2000년대에 태어난 외국인 남자 : 7 / 2000년대에 태어난 외국인 여자 : 8



	<code>rg_PS = findViewById(R.id.RadioGroupPS);</code>
<b>10. PS ( 구분 )</b> - rg_PS로 교수, 학생, 직원 중 자신의 구분을 입력하는 Radio Button을 활성화시킨다.	
	<code>et_ID = findViewById(R.id.editTextID);</code>
<b>11. ID ( 학번 / 사번 )</b> - et_ID로 유저의 학번과 사번을 입력하는 editText를 활성화시킨다.	
	<code>et_PN = findViewById(R.id.editTextPN);</code>
<b>12. PN ( 전화번호 )</b> - et_PN으로 유저의 전화번호를 입력하는 editText를 활성화시킨다.	
	<code>cb_AB = findViewById(R.id.checkBoxAB);</code>
<b>13. AB ( 해외 방문 여부 )</b> - cb_AB로 최근에 해외를 방문한 적이 있는지를 나타내는 항목을 checkBox호 활성화시킨다.	
	<pre> sb_FV = findViewById(R.id.seekBarFV); tv_FV = findViewById(R.id.textViewFV); sb_FV.setMax(50); sb_FV.setProgress(15); sb_FV.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {     @Override     public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {         tv_FV.setText(""+((progress)*0.1+35)+"°C");     }      @Override     public void onStartTrackingTouch(SeekBar seekBar) {     }      @Override     public void onStopTrackingTouch(SeekBar seekBar) {     } }); </pre>
<b>14. FV ( 열 )</b> - sb_FV로 열을 입력하는 seekBar를 활성화시키고, tv_FV로 seekBar로 입력된 값을 textView에 출력되도록 활성화시킨다. - seekBar는 최대 50까지 현재의 값은 15로 설정한다. - seekBar의 값에 0.1을 곱하고 35를 더하여 값이 출력된다. - seekBar를 통해 출력될 수 있는 값은 35°C부터 40°C까지이다.	
	<code>cb_AG = findViewById(R.id.checkBoxAG);</code>
<b>15. AG ( 개인 정보 수집 동의 )</b> - cb_AG로 개인정보 수집 동의를 얻는 checkBox를 활성화시킨다.	
	<pre> btn_SB = findViewById(R.id.buttonSB); </pre>
<b>16. SB ( 제출 )</b> - btn_SB로 개인정보 수집 동의에 대한 Button을 활성화시키고, onBtn	



### 3) 결과 화면



<p><b>Entry Log</b></p> <p>1) 이름(Name) : _____</p> <p>2) 생년월일(Date of Birth) : Click!</p> <p>3) 성별(Sex) : <input type="radio"/> Male</p> <p>4) 내/외국인(Local/Foreigner) : <input type="radio"/> 내국인(Local)</p> <p>5) 주민등록번호(Resident Registration Number) : _____ 유효성 검사(CHECK)</p> <p>6) 구분(Position) : <input type="radio"/> 교수(Prof) <input type="radio"/> 학생(Std) <input type="radio"/> 직원(Staff)</p> <p>7) 사번/학번(ID) : _____</p> <p>8) 전화번호(Phone No.) : _____</p> <p>9) 해외 방문 여부(Whether to visit overseas) <input type="checkbox"/></p> <p>10) 열(Fever) : _____ 36.5°C</p> <p>개인 정보 수집 동의(Consent to Collection and Use of Personal Information) <input type="checkbox"/></p> <p><b>제출(SUBMIT)</b></p>	<p><b>Entry Log</b></p> <p>1) 이름(Name) : Kim Ji Won</p> <p>2) 생년월일(Date of Birth) : 2020 4월 26일 (일)</p> <p>3) 성별(Sex) : _____</p> <p>4) 내/외국인(Local/Foreigner) : _____</p> <p>5) 주민등록번호(Resident Registration Number) : _____ 유효성 검사(CHECK)</p> <p>6) 구분(Position) : _____</p> <p>7) 사번/학번(ID) : _____</p> <p>8) 전화번호(Phone No.) : _____</p> <p>9) 해외 방문 여부(Whether to visit overseas) <input type="checkbox"/></p> <p>10) 열(Fever) : _____ 36.5°C</p> <p>개인 정보 수집 동의(Consent to Collection and Use of Personal Information) <input type="checkbox"/></p> <p><b>제출(SUBMIT)</b></p>
<p>- 앱 실행 시 초기 화면 (학번, 이름 출력)</p>	<p>- 생년월일 입력시 나타나는 데이트피커 화면</p>
<p><b>Entry Log</b></p> <p>1) 이름(Name) : Kim Ji Won</p> <p>2) 생년월일(Date of Birth) : 1998/10/3</p> <p>3) 성별(Sex) : <input checked="" type="radio"/> Female</p> <p>4) 내/외국인(Local/Foreigner) : <input type="radio"/> 내국인(Local)</p> <p>5) 주민등록번호(Resident Registration Number) : 981003 - 2 유효성 검사(CHECK)</p> <p>6) 구분(Position) : <input type="radio"/> 교수(Prof) <input checked="" type="radio"/> 학생(Std) <input type="radio"/> 직원(Staff)</p> <p>7) 사번/학번(ID) : _____</p> <p>8) 전화번호(Phone No.) : _____</p> <p>9) 해외 방문 여부(Whether to visit overseas) <input type="checkbox"/></p> <p>10) 열(Fever) : _____ 36.5°C</p> <p>개인 정보 수집 동의(Consent to Collection and Use of Personal Information) <input type="checkbox"/></p> <p><b>제출(SUBMIT)</b></p>	<p><b>Entry Log</b></p> <p>1) 이름(Name) : Kim Ji Won</p> <p>2) 생년월일(Date of Birth) : 1998/10/3</p> <p>3) 성별(Sex) : <input checked="" type="radio"/> Female</p> <p>4) 내/외국인(Local/Foreigner) : <input type="radio"/> 내국인(Local)</p> <p>5) 주민등록번호(Resident Registration Number) : 981003 - 2 유효성 검사(CHECK)</p> <p>6) 구분(Position) : <input type="radio"/> 교수(Prof) <input checked="" type="radio"/> 학생(Std) <input type="radio"/> 직원(Staff)</p> <p>7) 사번/학번(ID) : _____</p> <p>8) 전화번호(Phone No.) : _____</p> <p>9) 해외 방문 여부(Whether to visit overseas) <input type="checkbox"/></p> <p>10) 열(Fever) : _____ 36.5°C</p> <p>개인 정보 수집 동의(Consent to Collection and Use of Personal Information) <input type="checkbox"/></p> <p><b>제출(SUBMIT)</b></p>
<p>- 생년월일, 성별, 내외국인 입력 정보로 주민등록번호가 일부 자동 입력된 화면</p>	<p>- 주민등록번호 유효성 검사 통과 시 화면</p>



 <p><b>Entry Log</b></p> <p>1) 이름(Name) : Kim Ji Won</p> <p>2) 생년월일(Date of Birth) : 1998/10/3</p> <p>3) 성별(Sex) : <input checked="" type="radio"/> Female</p> <p>4) 내/외국인(Local/Foreigner) : <input checked="" type="radio"/> 내국인(Local)</p> <p>5) 주민등록번호(Resident Registration Number) : 981003 - 2 [Invalid] 유효성 검사(CHECK)</p> <p>6) 구분(Position) : <input type="radio"/> 교수(Prof) <input type="radio"/> 학생(Student) <input type="radio"/> 직원(Staff)</p> <p>7) 사번/학번(ID) : </p> <p>8) 전화번호(Phone No.) : </p> <p>9) 해외 방문 여부(Whether to visit overseas) <input type="checkbox"/></p> <p>10) 열(Fever) : <input type="range"/> 36.5°C</p> <p>주민등록번호가 유효하지 않습니다. (Invalid)</p> <p><b>제출(SUBMIT)</b></p>	 <p><b>Entry Log</b></p> <p>1) 이름(Name) : Kim Ji Won</p> <p>2) 생년월일(Date of Birth) : 1998/10/3</p> <p>3) 성별(Sex) : <input checked="" type="radio"/> Female</p> <p>4) 내/외국인(Local/Foreigner) : <input checked="" type="radio"/> 내국인(Local)</p> <p>5) 주민등록번호(Resident Registration Number) : 981003 - 2 [Valid] 유효성 검사(CHECK)</p> <p>6) 구분(Position) : <input type="radio"/> 교수(Prof) <input checked="" type="radio"/> 학생(Student) <input type="radio"/> 직원(Staff)</p> <p>7) 사번/학번(ID) : 20181151</p> <p>8) 전화번호(Phone No.) : 010-2167-0337</p> <p>9) 해외 방문 여부(Whether to visit overseas) <input type="checkbox"/></p> <p>10) 열(Fever) : <input type="range"/> 36.7°C</p> <p>개인 정보 수집 동의 (Personal Information) <input checked="" type="checkbox"/></p> <p>제출되었습니다.</p> <p><b>제출(SUBMIT)</b></p>
--	---

- 주민등록번호 유효성 검사 미 통과 시 화면

- 주민등록번호 유효성 통과와 개인정보 수집 동의 후 제출 버튼 클릭 시 화면

 <p><b>Entry Log</b></p> <p>1) 이름(Name) : Kim Ji Won</p> <p>2) 생년월일(Date of Birth) : 1998/10/3</p> <p>3) 성별(Sex) : <input checked="" type="radio"/> Female</p> <p>4) 내/외국인(Local/Foreigner) : <input checked="" type="radio"/> 내국인(Local)</p> <p>5) 주민등록번호(Resident Registration Number) : 981003 - 2 [Valid] 유효성 검사(CHECK)</p> <p>6) 구분(Position) : <input type="radio"/> 교수(Prof) <input checked="" type="radio"/> 학생(Student) <input type="radio"/> 직원(Staff)</p> <p>7) 사번/학번(ID) : 20181151</p> <p>8) 전화번호(Phone No.) : 010-2167-0337</p> <p>9) 해외 방문 여부(Whether to visit overseas) <input type="checkbox"/></p> <p>10) 열(Fever) : <input type="range"/> 36.7°C</p> <p>개인 정보 수집 동의의 항목은 필수 체크 항목입니다.</p> <p><b>제출(SUBMIT)</b></p>	 <p><b>Entry Log</b></p> <p>1) 이름(Name) : Kim Ji Won</p> <p>2) 생년월일(Date of Birth) : 1998/10/3</p> <p>3) 성별(Sex) : <input checked="" type="radio"/> Female</p> <p>4) 내/외국인(Local/Foreigner) : <input checked="" type="radio"/> 내국인(Local)</p> <p>5) 주민등록번호(Resident Registration Number) : 981003 - 2 [Invalid] 유효성 검사(CHECK)</p> <p>6) 구분(Position) : <input type="radio"/> 교수(Prof) <input checked="" type="radio"/> 학생(Student) <input type="radio"/> 직원(Staff)</p> <p>7) 사번/학번(ID) : 20181151</p> <p>8) 전화번호(Phone No.) : 010-2167-0337</p> <p>9) 해외 방문 여부(Whether to visit overseas) <input type="checkbox"/></p> <p>10) 열(Fever) : <input type="range"/> 36.7°C</p> <p>주민등록번호가 유효하지 않습니다. 재입력해주시요.</p> <p><b>제출(SUBMIT)</b></p>
--	---

- 개인정보 수집 동의하지 않은 상태로 제출 버튼 클릭 시 화면

- 주민등록번호 유효성 검사를 통과하지 않은 상태로 제출 버튼 클릭 시 화면

### 3. 결론

#### 1) 개인정보 확인 앱의 장점

① 건물 출입자 명단을 확인하는 주제를 가지고 어플리케이션을 개발하였다.

: 과제의 가장 핵심인 주민등록번호의 유효성 검사를 포함한 건물 출입자 명단을 확인하는 어플리케이션을 만들었다. 최근 코로나 바이러스로 인해 건물 출입자의 정보를 기록, 관리하는 것이 의무화되어 있는 경우가 많이 있다. 이러한 상황에서 수기로 기록하는 것이 아니라 어플리케이션을 통해 작성하면 편리할 것 같아 아이디어를 내고 다양한 컴파운드 버튼을 추가하여 제작하였다.

#### ② 자동 주민등록번호 입력

: 일반적으로 모든 사람의 주민등록번호에 규칙이 있다. 그중 생년월일과 성별을 나타내는 7번째 숫자까지는 예상할 수 있다. 앞서 입력한 생년월일과 성별, 내외국인 정보를 확인하여 자동으로 7번째 자리까지 입력되어 유저의 입장에서 편리할 수 있게 어플리케이션을 제작하였다.

#### ③ 주민등록번호 최대 자릿수 설정

: 주민등록번호는 13자리 숫자이다. 유저 입장에서 13자리 이상의 숫자를 입력하면 더 이상 입력되지 않도록 문자열의 최대 길이를 설정해 놓았다. 만약 성별과 내외국인 정보가 수정된다면 뒤에 숫자가 추가되는 것이 아니라 성별 정보에 해당하는 숫자를 삭제하고 새로운 숫자가 입력된다.

#### ④ editText의 목적별로 다른 InputType을 설정

: 이름을 입력하는 editText에는 InputType을 text로, 학번/사번을 입력하는 editText에는 InputType을 number로, 전화번호를 입력하는 editText에는 InputType을 phone으로 하여 유저가 사용할 때 적합한 키보드가 나타날 수 있도록 제작하였다.

#### ⑤ 개인정보 수집 동의를 필수 항목으로 설정

: 개인의 개인정보를 함부로 얻어내거나 수집하는 것은 불법 행위이다. 따라서 개인정보 수집 동의 항목에 체크하는 것은 필수이다. 만약 개인정보 수집 동의 항목에 체크되지 않은 상태로 제출하게 되면 제출되지 않고 개인정보 수집 동의 항목이 필수 체크 항목임을 유저에게 알린다.

## 2) 개인정보 확인 앱의 단점

### ① 한글 자판이 출력되지 않는다.

: editText의 각 InputType을 달리하여 편리함을 추구하였지만 가장 첫번째 항목인 이름을 입력하는 editText의 항목에서 원하지 않는 결과가 나왔다. InputType을 text로 설정한 후 나타나는 키보드를 보면 영어 자판만 출력된다. 한글 자판이 나오게끔 제작하고 싶어서 다양한 방법을 탐색해 보았지만 해결 방법을 찾지 못하였다.

### ② seekbar 자체의 기능으로 Min과 step을 설정하지 못했다.

: seekBar로 유저의 열을 입력할 때 seekBar의 기능 자체로 step과 Min 값을 설정하고 싶었으나 해당 기능을 어떻게 구현하는지 찾지 못하여 수식 통해 소수점 자리를 구현하였다

## 3) 배운 점

이번 프로젝트를 제작하면서 지금까지 배운 내용을 최대한 활용하고 싶다는 욕심을 바탕으로 여러 가지 기능을 포함시켜 어플리케이션을 제작하였다. 하나씩 순서대로 제작하여 큰 어려움 없이 프로젝트를 완성할 수 있었다. 가장 어려웠던 점은 seekBar를 구현하는 것이었다. 이 위젯을 사용할 때 초기값과 최대값을 설정해주게 되어있다. 초반에는 최대값과 초기값을 설정해주는 방식처럼 최소값과 step을 설정하는 것이 가능하다고 생각하였다. 하지만 설정 시 오류가 발생했고, 구글링을 해보아도 이를 해결할 수 있는 해결책을 찾지 못하였다. 고민 끝에 seekBar 자체를 step으로 생각하여 0.1을 곱한 후 35를 더해 출력하면 소수점 첫째자리까지 표현할 수 있다는 점을 깨달았다. 이를 해결한 후 새로운 방법을 찾아낸 것 같아 뿌듯하기도 했지만 한편으로는 seekBar 자체의 기능으로 구현하고 싶은 마음도 남아있었다. 이번 프로젝트를 통해 다양한 위젯을 사용하는 방법을 배우고 구현해보았다. 아직 초입단계의 학습이기 때문에 코드를 작성하는데 오래 걸리고 깔끔하지 못한 코드를 작성하기도 하고, 코드를 작성하는 것이 미숙하여 수업자료와 구글을 많이 참고하고 있다. 하지만 이렇게 모르는 것을 찾아가면서 하는 것이 더 오래 기억에 남는 것 같다. 어렵지만 이렇게 프로젝트를 제작하는 것이 실력 향상에 큰 도움이 되고 있다.