

# 7. 실습 보고서  
- Auto Generator 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	2020.05.03

- 목차 -

1. 서론	3p
1) 개요	
2) 요청사항 분석	
3) 초기 디자인	
2. 본론	
1) layout > activity_main.xml	4p
2) layout-land > activity_main.xml	5p
3) res > raw > snd1.mp3	6p
4) res > values > strings.xml	6p
5) MainActivity.java	
① 소스 코드 원본	6p
② 소스 코드 분석	10p
6) 결과 화면	13p
3. 결론	
1) 비밀번호 자동 생성기 앱 장점	15p
2) 비밀번호 자동 생성기 앱 단점	15p
3) 배운 점	16p

## 1. 서론

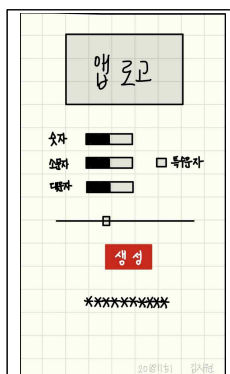
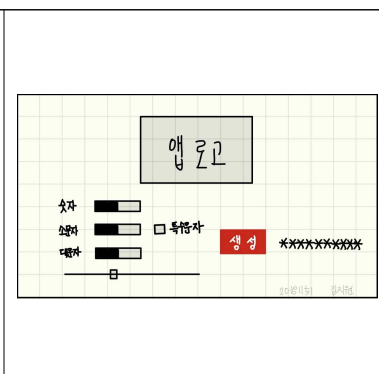
### 1) 개요

다양한 사이트에서 로그인을 요구하면 아이디와 비밀번호를 회원가입을 통해 만들어 입력 해야한다. 하지만 매번 비밀번호를 창의적으로 제작하는 것에 어려움을 느낀다. 해당 어플리케이션을 통해 랜덤으로 비밀번호를 생성할 수 있다. 숫자, 영어 소문자, 영어 대문자, 특수문자 그리고 비밀번호의 길이까지 옵션으로 선택하여 비밀번호를 자동 생성한다. 이는 일회용 비밀번호인 OTP(One Time Password) 번호로도 활용하여 보안을 강화시키는 용도로도 사용할 수 있다.

### 2) 요청사항 분석

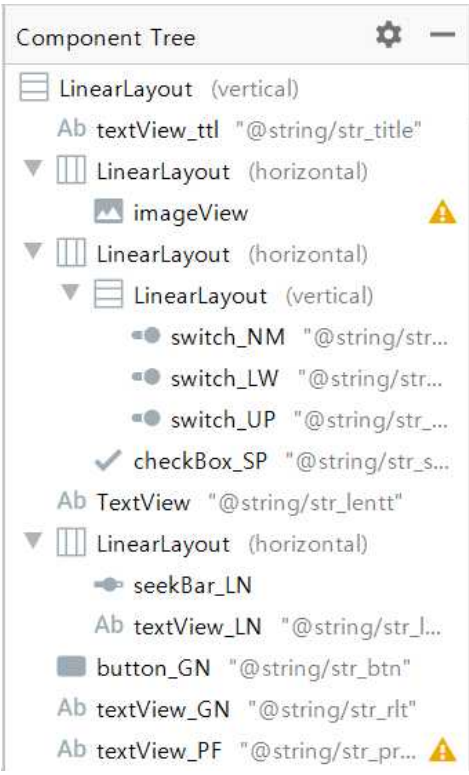
- ① Switch : 비밀번호의 옵션으로 숫자, 영어 소문자, 대문자를 Switch로 제작하여 옵션 선택함
- ② CheckBox : 비밀번호의 옵션으로 특수문자를 CheckBox로 제작하여 옵션 선택함
- ③ SeekBar : 비밀번호의 옵션으로 비밀번호의 길이를 SeekBar로 제작하여 최대 10자리의 비밀번호를 제작할 수 있음
- ④ Button 사운드 : Button을 클릭할 때 마다 극적인 효과를 내는 음향을 넣음
- ⑤ 앱 로고 : ImageView를 이용하여 직접 제작한 앱 로고를 넣음
- ⑥ 개발자 정보 : 앱의 하단 혹은 앱 로고 하단에 개발자 정보인 학번, 이름을 TextView를 이용하여 추가함
- ⑦ 가로 세로 디자인 : 디바이스를 회전시켰을 때, 화면에 맞는 Layout 배치를 다르게 디자인함
- ⑧ 비밀번호 출력 : 앱의 하단에 비밀번호를 생성하여 출력함

### 3) 초기 디자인

		<ul style="list-style-type: none"><li>- 상단에 어플리케이션 로고 이미지를 출력한다.</li><li>- 하단에 switch, checkbox, seekbar를 배치한다.</li><li>- 옵션의 하단 혹은 우측에 생성 Button을 배치한다.</li><li>- Button의 하단 혹은 우측에 자동 생성된 비밀번호를 출력한다.</li><li>- 화면의 가장 하단에 학번과 이름을 출력한다.</li></ul>
---	---	--


## 2. 본문

### 1) layout > activity\_main.xml



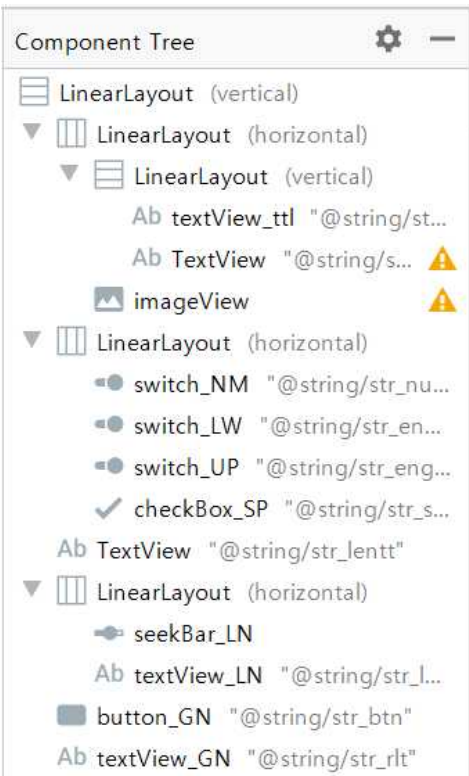
Component Tree

- LinearLayout (vertical)
  - Ab TextView\_ttl "@string/str\_title"
  - LinearLayout (horizontal)
    - imageView
  - LinearLayout (horizontal)
    - LinearLayout (vertical)
      - switch\_NM "@string/str..."
      - switch\_LW "@string/str..."
      - switch\_UP "@string/str..."
      - checkbox\_SP "@string/str\_s..."
    - Ab TextView "@string/str\_lentt"
    - LinearLayout (horizontal)
      - seekBar\_LN
      - Ab TextView\_LN "@string/str\_l..."
      - button\_GN "@string/str\_btn"
      - Ab TextView\_GN "@string/str\_rlt"
      - Ab TextView\_PF "@string/str\_pr..."




- vertical Linear Layout에 textView 4개와 Button 1개, 3개의 horizontal Linear Layout을 삽입한다.
- 첫 번째 textView에는 앱의 이름을 출력한다.
- 첫 번째 Linear Layout에는 imageView를 삽입하여 앱 로고 이미지를 출력한다.
- 두 번째 Linear Layout에는 vertical Layout과 checkbox를 삽입한다. vertical Layout에는 3개의 switch를 배치하여 옵션을 설정한다.
- 두 번째 TextView에는 “비밀번호 길이” 문자열을 출력한다.
- 세 번째 Linear Layout에는 seekBar와 TextView를 배치하여 seekBar의 값이 바뀌면 실시간으로 TextView에 값이 출력되도록 한다.
- 비밀번호를 생성하는 Button과 비밀번호를 출력하는 TextView, 학번이름을 출력하는 TextView를 배치한다.

## 2) layout-land > activity\_main.xml




Component Tree

- LinearLayout (vertical)
  - LinearLayout (horizontal)
    - LinearLayout (vertical)
      - Ab textView\_ttl "@string/st..."
      - Ab TextView "@string/s..."
      - imageView
    - LinearLayout (horizontal)
      - switch\_NM "@string/str\_nu..."
      - switch\_LW "@string/str\_en..."
      - switch\_UP "@string/str\_eng..."
      - checkbox\_SP "@string/str\_s..."
    - Ab TextView "@string/str\_lentt"
    - LinearLayout (horizontal)
      - seekBar\_LN
      - Ab textView\_LN "@string/str\_l..."
      - button\_GN "@string/str\_btn"
      - Ab textView\_GN "@string/str\_rlt"



- 세로 디자인과 약간 차이가 있다.
- vertical Linear Layout에 textView 2개와 Button 1개, 3개의 horizontal Linear Layout을 삽입한다.
- 첫 번째 Linear Layout에는 vertical Linear Layout과 ImageView를 삽입한다. Linear Layout에는 textView 2개를 삽입하여 앱 이름과 학번이름을 출력한다. *(학번이름 상단배치)*
- 두 번째 Linear Layout에는 switch 3개와 checkbox를 삽입하여 옵션 설정을 할 수 있도록 한다. *(옵션을 일렬로 배치)*
- TextView에는 “비밀번호 길이” 문자열을 출력한다.
- 세 번째 Linear Layout에는 seekBar와 TextView를 배치하여 seekBar의 값이 바뀌면 실시간으로 TextView에 값이 출력되도록 한다.
- 비밀번호를 생성하는 Button과 비밀번호를 출력하는 TextView를 배치한다.

### 3) res > raw > snd1.mp3

	<ul style="list-style-type: none"><li>- resource에 “raw” directory를 새로 생성하여 음원을 복사하여 삽입한다.</li></ul>
---	---

### 4) res > values > strings.xml

<pre>1 &lt;resources&gt; 2     &lt;string name="app_name"&gt;week7_assign_Autogenerator&lt;/string&gt; 3     &lt;string name="str_image"&gt;PW logo&lt;/string&gt; 4     &lt;string name="str_num"&gt;숫자&lt;/string&gt; 5     &lt;string name="str_engL"&gt;영어 소문자&lt;/string&gt; 6     &lt;string name="str_engU"&gt;영어 대문자&lt;/string&gt; 7     &lt;string name="str_spc"&gt;특수 문자&lt;/string&gt; 8     &lt;string name="str_lenttt"&gt;비밀번호 길이&lt;/string&gt; 9     &lt;string name="str_len"&gt;4자리&lt;/string&gt; 10    &lt;string name="str_btn"&gt;Generate 생성&lt;/string&gt; 11    &lt;string name="str_rlt"&gt;*****&lt;/string&gt; 12    &lt;string name="str_title"&gt;AUTO GENERATOR (자동 비밀번호 생성기)&lt;/string&gt; 13    &lt;string name="str_profile"&gt;20181151 김지원&lt;/string&gt; 14    &lt;string name="str_warn"&gt;조건을 선택하세요.&lt;/string&gt; 15 &lt;/resources&gt;</pre>	<ul style="list-style-type: none"><li>- resource의 value 중 string의 XML에 각종 문자열을 추가하여 코드에 이용한다.</li><li>- 하드코딩하는 것에 비해 프로그램의 유지보수가 좋아진다.</li></ul>
--	--

### 5) MainActivity.java

#### ① 소스 코드 원본

```
package com.example.week7_assign_autogenerator;

import androidx.appcompat.app.AppCompatActivity;

import android.media.AudioManager;
import android.media.SoundPool;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
```

```

import android.widget.CompoundButton;
import android.widget.SeekBar;
import android.widget.Switch;
import android.widget.TextView;
import android.widget.Toast;

import java.util.Random;

public class MainActivity extends AppCompatActivity {

    Switch sw_NM;
    Switch sw_LW;
    Switch sw_UP;
    CheckBox cb_SP;
    SeekBar sb_LN;
    TextView tv_LN;
    Button btn_GN;
    SoundPool sp_GN;
    int snd_id;
    TextView tv_GN;

    String[] num = {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9"};
    String[] eng_LW = {"a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z"};
    String[] eng_UP = {"A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z"};
    String[] special = {"!", "@", "#", "$", "%", "^", "&", "*", "(", ")", ""};
    String[] pass;

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

        sw_NM = findViewById(R.id.switch_NM);
        sw_LW = findViewById(R.id.switch_LW);
        sw_UP = findViewById(R.id.switch_UP);
        cb_SP = findViewById(R.id.checkBox_SP);

        sb_LN = findViewById(R.id.seekBar_LN);
        sb_LN.setMax(10);
        sb_LN.setProgress(4);
        sb_LN.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
            @Override
            public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
                tv_LN.setText(progress+"자리");
            }
        });

        @Override
        public void onStartTrackingTouch(SeekBar seekBar) {

        }

        @Override
        public void onStopTrackingTouch(SeekBar seekBar) {

        }
    });
    tv_LN = findViewById(R.id.textView_LN);

    sp_GN = new SoundPool(1, AudioManager.STREAM_MUSIC, 0);
    snd_id = sp_GN.load(this, R.raw.snd1, 1);
    btn_GN = findViewById(R.id.button_GN);
    btn_GN.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            sp_GN.play(snd_id, 1, 1, 0, 0, 1);
            Toast.makeText(getApplicationContext(), "CC0 1.0 Universal made by

```

```

DayDreamSound\n" +
"https://youtu.be/kfnh9QAfDgA",Toast.LENGTH_SHORT).show();

int len = sb_LN.getProgress();

if(sw_NM.isChecked()==false && sw_LW.isChecked()==false &&
sw_UP.isChecked()==false && cb_SP.isChecked()==false)
{
String str = getResources().getString(R.string.str_warn);
tv_GN.setText(str);
}
else
{
if(sw_NM.isChecked()==true && sw_LW.isChecked()==false &&
sw_UP.isChecked()==false && cb_SP.isChecked()==false)
{
pass = new String[num.length];
System.arraycopy(num, 0,pass,0,num.length);
}
else if(sw_NM.isChecked()==false && sw_LW.isChecked()==true &&
sw_UP.isChecked()==false && cb_SP.isChecked()==false)
{
pass = new String[eng_LW.length];
System.arraycopy(eng_LW, 0,pass,0,eng_LW.length);
}
else if(sw_NM.isChecked()==false && sw_LW.isChecked()==false &&
sw_UP.isChecked()==true && cb_SP.isChecked()==false)
{
pass = new String[eng_UP.length];
System.arraycopy(eng_UP, 0,pass,0,eng_UP.length);
}
else if(sw_NM.isChecked()==false && sw_LW.isChecked()==false &&
sw_UP.isChecked()==false && cb_SP.isChecked()==true)
{
pass = new String[special.length];
System.arraycopy(special, 0,pass,0,special.length);
}
else if(sw_NM.isChecked()==true && sw_LW.isChecked()==true &&
sw_UP.isChecked()==false && cb_SP.isChecked()==false)
{
pass = new String[num.length + eng_LW.length];
System.arraycopy(num, 0,pass,0,num.length);
System.arraycopy(eng_LW,0,pass,num.length,eng_LW.length);
}
else if(sw_NM.isChecked()==true && sw_LW.isChecked()==false &&
sw_UP.isChecked()==true && cb_SP.isChecked()==false)
{
pass = new String[num.length + eng UP.length];
System.arraycopy(num, 0,pass,0,num.length);
System.arraycopy(eng_UP,0,pass,num.length,eng UP.length);
}
else if(sw_NM.isChecked()==true && sw_LW.isChecked()==false &&
sw_UP.isChecked()==false && cb_SP.isChecked()==true)
{
pass = new String[num.length + special.length];
System.arraycopy(num, 0,pass,0,num.length);
System.arraycopy(special,0,pass,num.length,special.length);
}
else if(sw_NM.isChecked()==false && sw_LW.isChecked()==true &&
sw_UP.isChecked()==true && cb_SP.isChecked()==false)
{
pass = new String[eng_LW.length + eng UP.length];
System.arraycopy(eng_LW, 0,pass,0,eng_LW.length);
System.arraycopy(eng_UP,0,pass,eng_LW.length,eng UP.length);
}
else if(sw_NM.isChecked()==false && sw_LW.isChecked()==true &&
sw_UP.isChecked()==false && cb_SP.isChecked()==true)
{
pass = new String[eng_LW.length + special.length];
System.arraycopy(eng_LW, 0,pass,0,eng_LW.length);
System.arraycopy(special,0,pass,eng_LW.length,special.length);
}
}

```



```

else if(sw_NM.isChecked()==false && sw_LW.isChecked()==false &&
sw_UP.isChecked()==true && cb_SP.isChecked()==true)
{
pass = new String[eng_UP.length + special.length];
System.arraycopy(eng_UP, 0,pass,0,eng_UP.length);
System.arraycopy(special,0,pass,eng_UP.length,special.length);
}
else if(sw_NM.isChecked()==true && sw_LW.isChecked()==true &&
sw_UP.isChecked()==true && cb_SP.isChecked()==false)
{
pass = new String[num.length + eng_LW.length + eng_UP.length];
System.arraycopy(num, 0,pass,0,num.length);
System.arraycopy(eng_LW,0,pass,num.length,eng_LW.length);
System.arraycopy(eng_UP,0,pass,num.length+eng_LW.length,eng_UP.length);
}
else if(sw_NM.isChecked()==true && sw_LW.isChecked()==true &&
sw_UP.isChecked()==false && cb_SP.isChecked()==true)
{
pass = new String[num.length + eng_LW.length + special.length];
System.arraycopy(num, 0,pass,0,num.length);
System.arraycopy(eng_LW,0,pass,num.length,eng_LW.length);
System.arraycopy(special,0,pass,num.length+eng_LW.length,special.length);
}
else if(sw_NM.isChecked()==true && sw_LW.isChecked()==false &&
sw_UP.isChecked()==true && cb_SP.isChecked()==true)
{
pass = new String[num.length + eng_UP.length + special.length];
System.arraycopy(num, 0,pass,0,num.length);
System.arraycopy(eng_UP,0,pass,num.length,eng_UP.length);
System.arraycopy(special,0,pass,num.length+eng_UP.length,special.length);
}
else if(sw_NM.isChecked()==false && sw_LW.isChecked()==true &&
sw_UP.isChecked()==true && cb_SP.isChecked()==true)
{
pass = new String[eng_LW.length + eng_UP.length + special.length];
System.arraycopy(eng_LW, 0,pass,0,eng_LW.length);
System.arraycopy(eng_UP,0,pass,eng_LW.length,eng_UP.length);
System.arraycopy(special,0,pass,eng_LW.length+eng_UP.length,special.length);
}
else if(sw_NM.isChecked()==true && sw_LW.isChecked()==true &&
sw_UP.isChecked()==true && cb_SP.isChecked()==true)
{
pass = new String[num.length + eng_LW.length + eng_UP.length +
special.length];
System.arraycopy(num, 0,pass,0,num.length);
System.arraycopy(eng_LW,0,pass,num.length,eng_LW.length);
System.arraycopy(eng_UP,0,pass,num.length+eng_LW.length,eng_UP.length);
System.arraycopy(special,0,pass,num.length+eng_LW.length+eng_UP.length,speci
al.length);
}
Random random = new Random();

StringBuilder builder = new StringBuilder("");
for(int i=0; i<len;i++)
{
String str = pass[random.nextInt(pass.length)];
builder.append(str);
}
tv_GN.setText(builder.toString());
}
});
tv_GN = findViewById(R.id.textView_GN);
}
}

```

## ② 소스 코드 분석

```
package com.example.week7_assign_autogenerator;

import androidx.appcompat.app.AppCompatActivity;

import android.media.AudioManager;
import android.media.SoundPool;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.CompoundButton;
import android.widget.SeekBar;
import android.widget.Switch;
import android.widget.TextView;
import android.widget.Toast;

import java.util.Random;
```

### 1. Import

```
public class MainActivity extends AppCompatActivity {

    Switch sw_NM;
    Switch sw_LW;
    Switch sw_UP;
    CheckBox cb_SP;
    SeekBar sb_LN;
    TextView tv_LN;
    Button btn_GN;
    SoundPool sp_GN;
    int snd_id;
    TextView tv_GN;

    String[] num = {"0","1","2","3","4","5","6","7","8","9"};
    String[] eng_LW = {"a","b","c","d","e","f","g","h","i","j","k","l","m","n","o","p","q","r","s","t","u","v","w","x","y","z"};
    String[] eng_UP = {"A","B","C","D","E","F","G","H","I","J","K","L","M","N","O","P","Q","R","S","T","U","V","W","X","Y","Z"};
    String[] special = {"!", "@", "#", "$", "%", "^", "&", "*", "(", ")", " "};
    String[] pass;
```

### 2. Main Activity class

- 각 위젯을 선언한다.
- 음원을 사용할 것이기 때문에 Sound Pool도 선언한 후 파일을 load 할 변수인 "snd\_id"도 선언한다.
- 비밀번호 자동 생성 시 사용할 문자들의 집합을 array로 지정해준다.
- pass는 랜덤으로 문자 선택 시 최종으로 사용할 문자들의 array이다.

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

    sw_NM = findViewById(R.id.switch_NM);
    sw_LW = findViewById(R.id.switch_LW);
    sw_UP = findViewById(R.id.switch_UP);
    cb_SP = findViewById(R.id.checkBox_SP);
```

### 3. onCreate method & NM(숫자) & LW(영어 소문자) & UP(영어 대문자) & Sp(특수문자)

- 3개의 switch와 1개의 checkBox를 활성화 시킨다.

```
sb_LN = findViewById(R.id.seekBar_LN);
sb_LN.setMax(10);
sb_LN.setProgress(4);
sb_LN.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
    @Override
    public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
        tv_LN.setText(progress+"자리");
    }

    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {

    }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {

    }
});
tv_LN = findViewById(R.id.textView_LN);
```

### 4. LN(비밀번호 길이)

- seekBar를 활성화시키고, MAX는 10이고 Progress는 4로 이에 맞는 길이의 비밀번호를 생성한다.
- 리스너를 활용하여 seekBar의 값이 바뀔 때마다 실시간으로 TextView에 값이 출력된다.

```

sp_GN = new SoundPool( maxStreams: 1, AudioManager.STREAM_MUSIC, srcQuality: 0);
snd_id = sp_GN.load( context: this, R.raw.snd1, priority: 1);
btn_GN = findViewById(R.id.button_GN);
btn_GN.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        sp_GN.play(snd_id, leftVolume: 1, rightVolume: 1, priority: 0, loop: 0, rate: 1);
        Toast.makeText(getApplicationContext(), text: "CC0 1.0 Universal made by DayDreamSound\n" +
            "https://youtu.be/kfnh9QAfDgA", Toast.LENGTH_SHORT).show();

        int len = sb_LN.getProgress();
    }
});

```

## 5. 음원 재생

- SoundPool의 객체를 생성하여 sp\_GN으로 지정한 후 snd\_id에 raw에 저장한 음원 파일을 load한다.
- 버튼이 클릭할 때마다 해당 음원이 재생되고 Toast로 저작권 출처가 출력된다.
- 음원은 1번 반복(loop:0)되고, 속도는 음원 그대로의 속도(rate:1)인 1배속으로 출력된다.
- 동시에 아래의 코드가 실행된다.

```

if(sw_NM.isChecked()==false && sw_LW.isChecked()==false && sw_UP.isChecked()==false && cb_SP.isChecked()==false)
{
    String str = getResources().getString(R.string.str_warn);
    tv_GN.setText(str);
}
else
{
    if(sw_NM.isChecked()==true && sw_LW.isChecked()==false && sw_UP.isChecked()==false && cb_SP.isChecked()==false)
    {
        pass = new String[num.length];
        System.arraycopy(num, srcPos: 0, pass, destPos: 0, num.length);
    }
    else if(sw_NM.isChecked()==false && sw_LW.isChecked()==true && sw_UP.isChecked()==false && cb_SP.isChecked()==false)
    {
        pass = new String[eng_LW.length];
        System.arraycopy(eng_LW, srcPos: 0, pass, destPos: 0, eng_LW.length);
    }
    else if(sw_NM.isChecked()==false && sw_LW.isChecked()==false && sw_UP.isChecked()==true && cb_SP.isChecked()==false)
    {
        pass = new String[eng_UP.length];
        System.arraycopy(eng_UP, srcPos: 0, pass, destPos: 0, eng_UP.length);
    }
    else if(sw_NM.isChecked()==false && sw_LW.isChecked()==false && sw_UP.isChecked()==false && cb_SP.isChecked()==true)
    {
        pass = new String[special.length];
        System.arraycopy(special, srcPos: 0, pass, destPos: 0, special.length);
    }
    else if(sw_NM.isChecked()==true && sw_LW.isChecked()==true && sw_UP.isChecked()==false && cb_SP.isChecked()==false)
    {
        pass = new String[num.length + eng_LW.length];
        System.arraycopy(num, srcPos: 0, pass, destPos: 0, num.length);
        System.arraycopy(eng_LW, srcPos: 0, pass, num.length, eng_LW.length);
    }
    else if(sw_NM.isChecked()==true && sw_LW.isChecked()==false && sw_UP.isChecked()==true && cb_SP.isChecked()==false)
    {
        pass = new String[num.length + eng_UP.length];
        System.arraycopy(num, srcPos: 0, pass, destPos: 0, num.length);
        System.arraycopy(eng_UP, srcPos: 0, pass, num.length, eng_UP.length);
    }
    else if(sw_NM.isChecked()==true && sw_LW.isChecked()==false && sw_UP.isChecked()==false && cb_SP.isChecked()==true)
    {
        pass = new String[num.length + special.length];
        System.arraycopy(num, srcPos: 0, pass, destPos: 0, num.length);
        System.arraycopy(special, srcPos: 0, pass, num.length, special.length);
    }
    else if(sw_NM.isChecked()==false && sw_LW.isChecked()==true && sw_UP.isChecked()==true && cb_SP.isChecked()==false)
    {
        pass = new String[eng_LW.length + eng_UP.length];
        System.arraycopy(eng_LW, srcPos: 0, pass, destPos: 0, eng_LW.length);
        System.arraycopy(eng_UP, srcPos: 0, pass, eng_LW.length, eng_UP.length);
    }
    else if(sw_NM.isChecked()==false && sw_LW.isChecked()==true && sw_UP.isChecked()==false && cb_SP.isChecked()==true)
    {
        pass = new String[eng_LW.length + special.length];
        System.arraycopy(eng_LW, srcPos: 0, pass, destPos: 0, eng_LW.length);
        System.arraycopy(special, srcPos: 0, pass, eng_LW.length, special.length);
    }
}

```

```

else if(sw_NM.isChecked()==false && sw_LW.isChecked()==true && sw_UP.isChecked()==true && cb_SP.isChecked()==false)
{
    pass = new String[eng_UP.length + special.length];
    System.arraycopy(eng_UP, srcPos: 0,pass, destPos: 0,eng_UP.length);
    System.arraycopy(special, srcPos: 0,pass,eng_UP.length,special.length);
}
else if(sw_NM.isChecked()==true && sw_LW.isChecked()==true && sw_UP.isChecked()==true && cb_SP.isChecked()==false)
{
    pass = new String[num.length + eng_LW.length + eng_UP.length];
    System.arraycopy(num, srcPos: 0,pass, destPos: 0,num.length);
    System.arraycopy(eng_LW, srcPos: 0,pass,num.length,eng_LW.length);
    System.arraycopy(eng_UP, srcPos: 0,pass, destPos: num.length+eng_LW.length,eng_UP.length);
}

else if(sw_NM.isChecked()==true && sw_LW.isChecked()==true && sw_UP.isChecked()==false && cb_SP.isChecked()==true)
{
    pass = new String[num.length + eng_LW.length + special.length];
    System.arraycopy(num, srcPos: 0,pass, destPos: 0,num.length);
    System.arraycopy(eng_LW, srcPos: 0,pass,num.length,eng_LW.length);
    System.arraycopy(special, srcPos: 0,pass, destPos: num.length+eng_LW.length,special.length);
}
else if(sw_NM.isChecked()==true && sw_LW.isChecked()==false && sw_UP.isChecked()==true && cb_SP.isChecked()==true)
{
    pass = new String[num.length + eng_UP.length + special.length];
    System.arraycopy(num, srcPos: 0,pass, destPos: 0,num.length);
    System.arraycopy(eng_UP, srcPos: 0,pass,num.length,eng_UP.length);
    System.arraycopy(special, srcPos: 0,pass, destPos: num.length+eng_UP.length,special.length);
}
else if(sw_NM.isChecked()==false && sw_LW.isChecked()==true && sw_UP.isChecked()==true && cb_SP.isChecked()==true)
{
    pass = new String[eng_LW.length + eng_UP.length + special.length];
    System.arraycopy(eng_LW, srcPos: 0,pass, destPos: 0,eng_LW.length);
    System.arraycopy(eng_UP, srcPos: 0,pass,eng_LW.length,eng_UP.length);
    System.arraycopy(special, srcPos: 0,pass, destPos: eng_LW.length+eng_UP.length,special.length);
}

else if(sw_NM.isChecked()==true && sw_LW.isChecked()==true && sw_UP.isChecked()==true && cb_SP.isChecked()==true)
{
    pass = new String[num.length + eng_LW.length + eng_UP.length + special.length];
    System.arraycopy(num, srcPos: 0,pass, destPos: 0,num.length);
    System.arraycopy(eng_LW, srcPos: 0,pass,num.length,eng_LW.length);
    System.arraycopy(eng_UP, srcPos: 0,pass, destPos: num.length+eng_LW.length,eng_UP.length);
    System.arraycopy(special, srcPos: 0,pass, destPos: num.length+eng_LW.length+eng_UP.length,special.length);
}
}

```

## 6. GN(비밀번호 생성)의 pass 제작

- Button이 클릭될 때마다 switch와 checkBox의 상태 정보를 가져와 적합한 if문의 명령어를 실행한다.
- pass의 길이를 고려하여 새로운 객체를 생성해준 후 선택된 옵션의 배열을 복사하여 pass에 넣는다.

```
Random random = new Random();

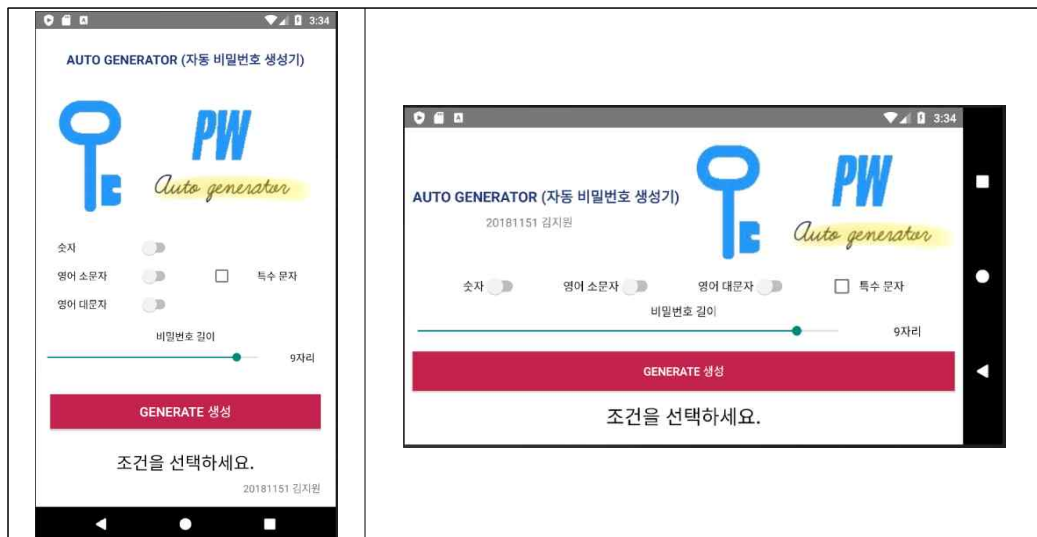
StringBuilder builder = new StringBuilder("");
for(int i=0; i<len;i++)
{
    String str = pass[random.nextInt(pass.length)];
    builder.append(str);
}
tv_GN.setText(builder.toString());
}
});
tv_GN = findViewById(R.id.textView_GN);
}
```

## 7. GN(비밀번호 생성)

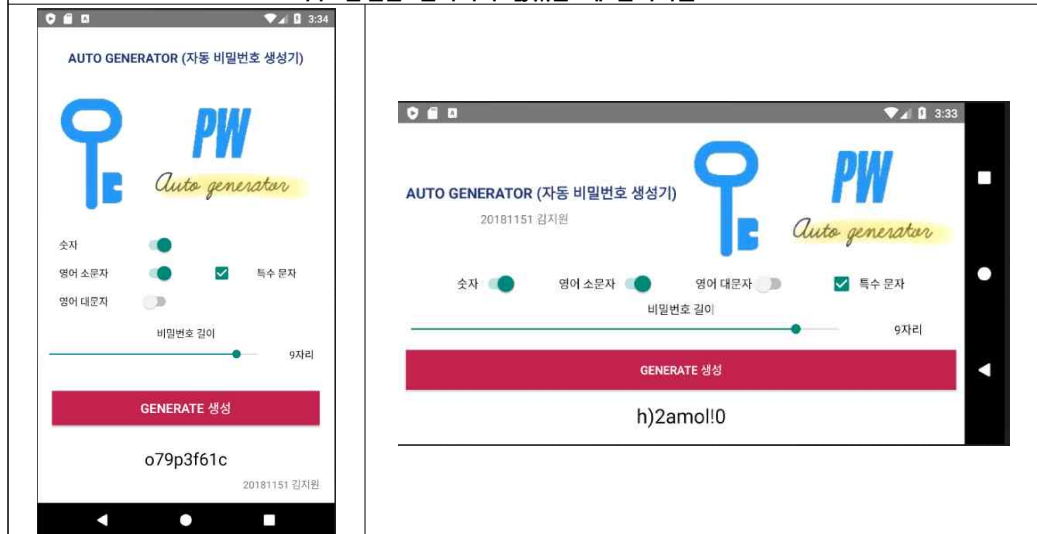
- 생성된 pass에서 seekBar의 값 만큼의 문자를 랜덤으로 선택하여 tv\_GN에 출력한다.
- 문자열을 일괄로 나열하기 위해 StringBuilder를 이용한다.

## 6) 결과 화면





아무 옵션을 선택하지 않았을 때 출력되는 Text



옵션 선택 후 자동 비밀번호 생성 화면

### 3. 결론

#### 1) 개인정보 확인 앱의 장점

##### ① 음원의 저작권을 공개하였다.

: 인터넷에 나와있는 음원을 함부로 사용하면 저작권 보호법에 걸리게 된다. 해당 어플리케이션을 제작할 때, 인터넷에서 무료 음원을 다운로드 받아서 사용하였다. 무료 사용이 가능하지만 수정 재사용 혹은 배포하기 위해서는 사이트에서 제시하는 출처를 필수로 밝히는 것을 규칙으로 하고 있다. 따라서 해당 음원을 사용할 때 어떤 방법으로 출처를 공개할지 고민하였다. Button을 클릭할 때마다 음원이 출력되기 때문에 Button 클릭 시 Toast로 출처를 공개하였다. 하지만 Toast 출력으로 비밀번호 생성기 어플리케이션을 사용함에 있어 불편함을 줄 수 있기 때문에 불편함을 최소화하기 위해 Toast의 출력 시간으로 짧게 수정하였다.

##### ② 상단의 파일이름을 제거했다.

: 어플리케이션 제작에 있어 모든 위젯을 스크롤 없이 한 화면에 들어가게 제작하기로 계획하였다. 하지만 삽입되는 내용이 많았기 때문에 화면의 공간이 부족했다. 현재 이 어플리케이션을 제작하면서 파일 이름은 불필요했기 때문에 이를 제거하고자 했다. 구글 검색을 통해 방법을 찾았고, app > manifests > AndroidManifest.xml 에서  
android : theme = “@style/Theme.AppCompat.Light.NoActionBar“ 을 입력하여 상단의 불필요한 부분을 제거하였다. 따라서 한 화면의 공간을 최대한 활용할 수 있게 되었다.

#### 2) 개인정보 확인 앱의 단점

##### ① 알고리즘의 반복이 많아 복잡해 가독성이 떨어진다.

: switch와 checkBox의 상태를 확인하여 Random의 후보로 사용할 문자들을 통합하여야 했다. 이 과정에서 알고리즘을 switch 3개의 상태, checkbox 1개의 상태를 모두 경우의 수를 계산하여  $2 \times 2 \times 2 \times 2 = 16$ 으로 16가지의 경우로 나누어 제작하였다. 따라서 코드의 길어도 불필요하게 길어지고, 복잡해보였다. 그리고 반복되는 코드가 많아 이를 다른 형태로 알고리즘을 제작한다면 훨씬 가독성이 높아질 것이라 생각한다. 하지만 방법을 찾지 못해 아쉬움이 남는다.

### 3) 배운 점

지금까지 했던 과제들은 큰 문제없이 비교적 쉽게 수행하였었다. 이번 과제는 수업 때 공지를 받고 많이 어렵지 않을 것이라 생각하였다. 하지만 가장 힘들었던 과제가 되었다. 특히 Sound Pool을 이용할 때 수업시간의 방법을 그대로 사용하였지만 계속 오류가 발생하였다. 음원 파일을 load 하는 과정에서 “R.raw.snd1”을 입력하면 raw에서 에러가 발생하였다. 해결방법으로 제시된 것은 snd1의 xml 파일을 제작하라고 제시되었다. 하지만 AVD에서 실행 했을 때는 음원이 정상 작동하였다. 구글 검색을 해본 결과 동일한 오류가 발생한 경우는 찾지 못하였다. 몇 시간을 고민하다가 다음 날로 미루었는데 다음날 다시 보니 이상하게 수정사항이 없는데 오류가 사라지고 정상 작동되었다. Sound Pool에서 정상적이지 않은 에러를 겪는 사람도 다수 있었고 간혹 파일을 삽입했음에도 불구하고 인식이 늦어지는 경우도 있다고 한다. 아마 이 부분에서 에러가 발생했던 것 같다. 처음에는 수업시간에는 wav 파일을 사용하였는데 과제 중에는 mp3 파일을 사용해서 문제가 발생하는 것인지도 고민해보고 개념적으로 부족한 부분이 있는지도 의심되어 수업을 반복해서 시청하기도 했다. 결과적으로는 허무하게 해결된 문제였지만, 고민하는 과정에서 개념적으로 다시 한번 재정립하는 기획이 되었다.