

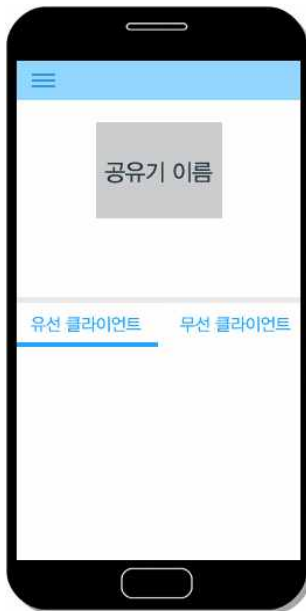
# APP 5월 넷째주 (예진)

---

## <이번 주 진행된 내용>

### 1. 어플 구상도 재작성

지난주에 회의했던 내용에 따라 어플 구상도를 다시 작성해봤다.



#### 메인화면

공유기 이름과 함께 그 아래에는 현재 공유기 상태를 표시하는 쪽으로 생각했다. 하단에는 유선 클라이언트와 무선 클라이언트 칸을 따로 나뉘었으며 fragment를 사용해 스와이프 할 수 있도록 한다.



#### 빠른 설치 화면

웹 팀과 동일한 형식으로 하기로 했다.



## 공유기 비밀번호 변경

해킹에 대비하기 위해 현재 비밀번호 입력이 일정 횟수 이상 오류가 났을 때, 더 이상 변경을 하지 못하게 차단하는 쪽을 생각해봤다. 차단을 푸는 방법은 회원가입을 했을 때 작성했던 이메일로 해제 코드를 보내는 방법을 선택했다. 시간 내로 코드를 입력하지 못했을 경우에는 액티비티가 종료된다.

## 2. 문자 일치 체크 기능(비밀번호 변경의 '비밀번호 확인' 부분)

간단하게 설명하면 문자를 입력할 수 있는 두 개의 EditText를 구성하고 두 번째 텍스트 뷰에 이벤트를 연결하면 완성된다.

### 1) 해당 화면이 뜰 레이아웃 xml 파일을 아래와 같이 수정한다.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:weightSum="100"
    android:background="#000000"
    android:orientation="vertical">
    <!--
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Touch Event!"
            android:textColor="#ffffff"
            android:textSize="20dp"
            android:background="#000000"
            android:id="@+id/touchEvent"/>

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
```

```

        android:text="Button Event"
        android:textSize="20dp"
        android:id="@+id/buttonEvent"/>-->
<!--<CheckBox
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="체크박스입니다."
    android:textSize="25dp"
    android:button="@null"
    android:drawableLeft="@drawable/check_color"
    android:textColor="#ffffff"
/>-->
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="30dp"
    android:background="#60000000"
    android:textColor="#ffffff"
    android:hint="editText"
    android:textColorHint="#80ffffff"
    android:id="@+id/firstText"/>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:weightSum="10"
    android:orientation="horizontal">

    <EditText
        android:layout_width="0dp"
        android:layout_weight="8.5"
        android:layout_height="wrap_content"
        android:textSize="30dp"
        android:background="#60000000"
        android:textColor="#ffffff"
        android:hint="editText2"
        android:textColorHint="#80ffffff"
        android:id="@+id/secondText"/>

    <ImageView
        android:layout_width="0dp"
        android:layout_weight="1.5"
        android:layout_height="match_parent"
        android:scaleType="fitStart"
        android:id="@+id/setImage"/>
</LinearLayout>
</LinearLayout>

```

2) 해당 Activity 자바 파일에 아래의 코드를 작성한다.

```

package com.example.a15u560.myapplication;

import android.app.Activity;
import android.os.Bundle;

```

```

import android.textEditable;
import android.text.TextWatcher;
import android.widget.EditText;
import android.widget.ImageView;

public class MainActivity extends Activity {

    EditText firstText, secondText;
    ImageView setImage;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        firstText = (EditText)findViewById(R.id.firstText);
        secondText = (EditText)findViewById(R.id.secondText);
        setImage = (ImageView)findViewById(R.id.setImage);

        secondText.addTextChangedListener(new TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {

            }

            @Override
            public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {
                if(firstText.getText().toString().equals(secondText.getText().toString())) {
                    setImage.setImageResource(R.drawable.sign_up_password_right);
                } else {

                    setImage.setImageResource(R.drawable.sign_up_password_current);
                }
            }

            @Override
            public void afterTextChanged(Editable editable) {

            }
        });
    }
}

```

### 3. 시간제한 있는 이메일 인증

1) 해당 화면을 띄울 레이아웃 xml 파일에 아래의 코드를 작성한다.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
        android:orientation="vertical"
        tools:context="com.example.gdtbg.blog.Auth">

        <EditText
            android:id="@+id/authEmail"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="인증 받을 이메일 계정을 입력하세요."
            android:gravity="center"/>

        <Button
            android:id="@+id/authBtn"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:text="이메일 인증하기" />

    </LinearLayout>

```

2) 레이아웃 xml 파일에 입력창과 버튼 하나를 만들어준다.

3) 이메일 인증 버튼을 클릭하면 톨 다이얼로그의 레이아웃 파일을 작성한다.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#CFD8DC"
    android:orientation="vertical"
    >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="이메일 인증하기"
        android:layout_marginTop="20dp"
        android:layout_gravity="center"
        android:textSize="30sp"
        />

    <EditText
        android:id="@+id/emailAuth_number"
        android:layout_width="300dp"
        android:layout_height="50dp"
        android:layout_marginTop="20dp"
        android:hint="인증 번호를 등록해 주세요"
        android:paddingLeft="10dp"
        android:layout_margin="20dp"
        android:background="#FFFFFF"
    >

```

```

        android:textSize="15sp"
        android:layout_gravity="center"
    />

    <TextView
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:text="이메일로 발송 된 인증 번호를 적어주세요."
        android:layout_margin="20dp"
        android:textColor="@android:color/black"
        android:layout_gravity="center"
    />

    <TextView
        android:id="@+id/emailAuth_time_counter"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="3:00"
        android:textSize="30sp"
        android:layout_marginTop="20dp"
        android:gravity="center"
    />

    <Button
        android:id="@+id/emailAuth_btn"
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="30dp"
        android:background="#FFFFFF"
        android:text="OK"
        android:textSize="22sp"
        android:layout_marginBottom="30dp"
        android:layout_gravity="center"/>

</LinearLayout>

```

#### 4) 버튼을 누르면 이메일 인증 다이얼로그가 뜨도록 자바 파일을 작성한다.

```

import android.app.Dialog;
import android.content.DialogInterface;
import android.os.Bundle;
import android.os.CountDownTimer;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class Auth extends AppCompatActivity implements View.OnClickListener, Dialog.OnCancelListener {

```

```

final int randomNum = 106254; // 테스트할 6자리 인증번호

EditText authEmail;
Button authBtn;

/*Dialog에 관련된 필드*/

LayoutInflater dialog; //LayoutInflater
View dialogLayout; //layout을 담을 View
Dialog authDialog; //dialog 객체

/*카운트 다운 타이머에 관련된 필드*/

TextView time_counter; //시간을 보여주는 TextView
EditText emailAuth_number; //인증 번호를 입력 하는 칸
Button emailAuth_btn; // 인증버튼
CountDownTimer countDownTimer;
final int MILLISINFUTURE = 300 * 1000; //총 시간 (300초 = 5분)
final int COUNT_DOWN_INTERVAL = 1000; //onTick 메소드를 호출할 간격 (1초)

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

    authEmail = (EditText) findViewById(R.id.authEmail);
    authBtn = (Button) findViewById(R.id.authBtn);
    authBtn.setOnClickListener(this);

}

public void countDownTimer() { //카운트 다운 메소드

    time_counter = (TextView) dialogLayout.findViewById(R.id.emailAuth_time_counter);
    //줄어드는 시간을 나타내는 TextView
    emailAuth_number = (EditText) dialogLayout.findViewById(R.id.emailAuth_number);
    //사용자 인증 번호 입력창
    emailAuth_btn = (Button) dialogLayout.findViewById(R.id.emailAuth_btn);
    //인증하기 버튼

    countDownTimer = new CountDownTimer(MILLISINFUTURE, COUNT_DOWN_INTERVAL) {
        @Override
        public void onTick(long millisUntilFinished) { //(300초에서 1초 마다 계속 줄어듦)

            long emailAuthCount = millisUntilFinished / 1000;
            Log.d("Alex", emailAuthCount + "");
        }
    };
}

```

```

        if ((emailAuthCount - ((emailAuthCount / 60) * 60)) >= 10) { //초가 10보다 크면 그냥 출력
            time_counter.setText((emailAuthCount / 60) + " : " + (emailAuthCount -
((emailAuthCount / 60) * 60)));
        } else { //초가 10보다 작으면 앞에 '0' 붙여서 같이 출력. ex) 02,03,04...
            time_counter.setText((emailAuthCount / 60) + " : 0" + (emailAuthCount -
((emailAuthCount / 60) * 60)));
        }

        //emailAuthCount은 종료까지 남은 시간임. 1분 = 60초 되므로,
        // 분을 나타내기 위해서는 종료까지 남은 총 시간에 60을 나눠주면 그 몫이 분이 된다.
        // 분을 제외하고 남은 초를 나타내기 위해서는, (총 남은 시간 - (분*60) = 남은 초) 로 하면 된다.

    }

    @Override
    public void onFinish() { //시간이 다 되면 다이얼로그 종료

        authDialog.cancel();

    }
}.start();

emailAuth_btn.setOnClickListener(this);
}

@Override
public void onClick(View v) {

    switch (v.getId()) {

        case R.id.authBtn:

            dialog = LayoutInflater.from(this);
            dialogLayout = dialog.inflate(R.layout.auth_dialog, null); // LayoutInflater를 통해 XML에 정의된
Resource들을 View의 형태로 반환 시켜 줌
            authDialog = new Dialog(this); //Dialog 객체 생성
            authDialog.setContentView(dialogLayout); //Dialog에 inflate한 View를 탑재 하여줌
            authDialog.setCanceledOnTouchOutside(false); //Dialog 바깥 부분을 선택해도 닫히지 않게 설정함.
            authDialog.setOnCancelListener(this); //다이얼로그를 닫을 때 일어날 일을 정의하기 위해
onCancelListener 설정
            authDialog.show(); //Dialog를 나타내어 준다.
            countdownTimer();
            break;

        case R.id.emailAuth_btn : //다이얼로그 내의 인증번호 인증 버튼을 눌렀을 시

            int user_answer = Integer.parseInt(emailAuth_number.getText().toString());
            if(user_answer==randomNum){
                Toast.makeText(this, "이메일 인증 성공", Toast.LENGTH_SHORT).show();
            }
        }
    }
}

```



```

        }else{

            Toast.makeText(this, "이메일 인증 실패", Toast.LENGTH_SHORT).show();
        }
        break;

    }

}

@Override
public void onCancel(DialogInterface dialog) {
    countdownTimer.cancel();
} //다이얼로그 닫을 때 카운트 다운 타이머의 cancel()메소드 호출
}

```

해당 코드는 이메일을 사용자가 직접 입력하는 방식이기 때문에 수정이 필요하다.

\*시간 카운트 기능을 구현할 때 `CountDownTimer` 라는 클래스를 사용하면 된다.

final void	cancel() 카운트를 종료한다.
abstract void	onFinish() 카운트가 종료 되었을 시 일어날 일을 정의한다.
abstract void	onTick(long millisUntilFinished) 생성자에서 지정해준 시간 간격이 지날 때마다 일어날 일을 정의한다. millisUntilFinished 라는 인자는 호출된 시점부터 종료될 때까지 남은 시간이다.
final CountDownTimer	start() 카운트를 시작한다.

Parameters	
millisInFuture	long start()메소드 호출 시점부터 onFinish()메소드가 호출되는 시점까지의 총 시간이다. 간단히 말해서 얼마동안 타이머를 실행시킬 건지 ms 단위로 넘겨주면 된다.
countDownInterval	long 시간 간격을 얼마로 설정해서 onTick 메소드를 호출시킬 건지 설정하는 인자다. 예를 들어 1초의 시간 간격으로 onTick 메소드를 호출하고 싶으면 1000을 넘겨주면 된다.

## 〈다음 주 진행할 내용〉

1. 비밀번호 변경 화면 구현하기
2. 일정 입력 횟수를 초과할 시 입력을 차단하는 것 구현하기