

QR코드를 이용한 능동적 출석체크

팀	명 :	응애세력
지도	교수 :	이병천 교수님
팀	장 :	강인제
팀	원 :	박순현
		박종명
		진효준

2020. 11.

중부대학교 정보보호학과

목 차

1. 서론

1.1 연구 목적	3
1.2 연구 필요성	3
1.3 연구 목적 및 주제 선정	3

2. 관련 연구

2.1 FIDO	3
2.2 출석체크 알고리즘	5
2.2.1 패시브 방식	5
2.2.2 액티브 방식	5
2.3 WIFI 탐지 알고리즘	5

3. 본론

3.1 서비스 설계	6
3.2 구현 환경 보안 기술	7
3.3 주요 기능 데모	7

4. 결론

4.1 결론	8
4.2 기대 효과 아니면 향후 계획	8

5. 별첨

5.1 소스 코드	9
5.2 발표 자료	85

1. 서론

1.1 연구 배경

기존 출석과 신원확인 서비스에서의 블루투스 기술은 상당수의 단점과 취약점을 내포하고 있다. 한국인터넷진흥원(KISA)에서는 모바일과 IOT 환경에서의 블루투스 취약점과 관련한 연구를 지속적으로 발표하며 해당 대응 조치로 보안 업데이트 권고와 업데이트를 하지 않은 경우 기능을 사용하지 않을 것을 권고하고 있다. 또한 블루투스 서비스는 전파의 거리와 장애물에 따른 전파 수신 여부가 데이터 수신율에 영향을 주게 되고, 서비스 시 지속적인 연결 상태를 요구할 경우 전력 소비 또한 증가하게 되며 한정적인 자원을 가진 모바일 환경에서의 사용자 경험은 더욱 줄어들게 된다.

1.2 연구 필요성

QR코드의 경우 사용자가 직접 태깅하는 행위에 의미 부여가 가능하며 이러한 태깅의 인식률 또한 높아 사용자 경험에 해가 되지 않는다. QR코드는 사용자가 타인에게 전송하여 태깅하는 행위의 정당성을 훼손할 수 있다, 그렇기에 WIFI와 같은 신호기술과 결합하여 신뢰성을 높일 수 있다.

1.3 연구 목적 및 주제 선정

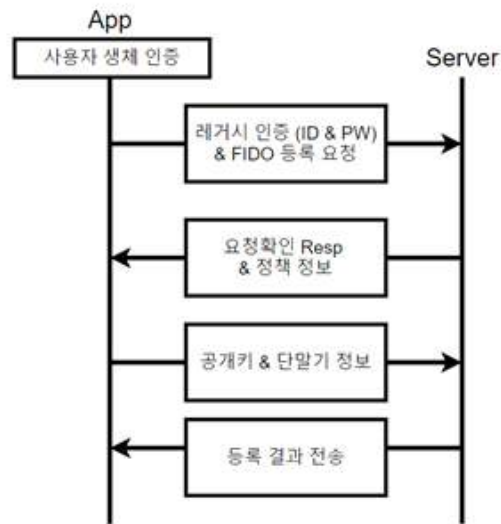
이러한 사용자 경험을 개선함과 보안에 있어 취약점을 극복하기 위해 QR 코드를 사용하여 사용자가 직접 자신에게 생성된 QR코드를 태깅하거나(이하 패시브 방식이라 한다) 공간에 설치된 디스플레이에 표기된 QR코드를 찍는 방식(이하 액티브 방식이라 한다)을 선택 사용하여 인증하게 되고 해당 인증 과정에서는 사용자 단말기와 FIDO((Fast IDentity Online)를 구성하여 생체정보를 통한 사용자 등록 후 전자 서명 방식의 멀티채널 인증을 하여 사용자 확인절차를 밟게 된다.

2. 관련 연구

2.1 FIDO 알고리즘

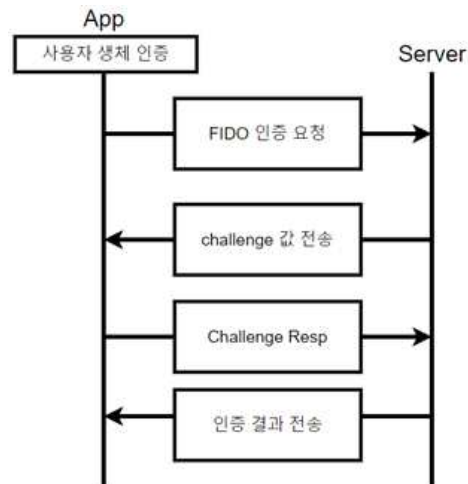
먼저 사용자 단말의 인증과 간편한 로그인 구성을 위해 FIDO를 구성하였다. FIDO 과정은 사용자 등록과 인증으로 나뉘어 진다.

사용자 등록일 경우 사용자가 생체정보 등록 시 비밀키와 공개키를 생성하고, 안드로이드 키스토어에 저장한다. 이후 서버에게 등록 요청을 보내게 되며 이때 인증은 기존의 ID와 Password 방식의 레거시 인증으로 등록한다. 등록 요청에 대한 서버의 리스폰스가 온다면 생성해 두었던 공개키와 단말기 정보를 서버에 전송하게 된다.



(그림 1) FIDO 등록 과정

등록 과정 이후 인증 과정 시 사용자는 생체 인증 후 서버에게 인증요청을 하게 되고 서버에서 보내오는 Challenge 값을 저장되어 있던 비밀키로 암호화하여 서버에게 보낸다. 서버는 해당 암호화 값을 검증하여 사용자에게 인증 결과를 전송한다.



(그림 2) FIDO 인증 과정

사용자의 비밀키는 안드로이드 OS에서 제공하는 키스토어에 보관되며 이는 사용자의 생체인식으로만 접근할 수 있고 사용자의 생체정보 변화, 애플리케이션의 재설치 등의 비정상적인 변화 시에 키 스토어의 비밀키는 말소된다. 이후 사용자는 재등록과정을 거치게 되며 재등록과정시 단말기의 변경 여부를 통해 비인가 단말의 인증이 거부되며 관리자의 승인이 필요해진다. 이는 사용자가 비정상적인 접근을 통해 QR코드를 인증하는 방법을 차단하기 위해 고안되었다. 안드로이드 키 스토어에 보관되는 비밀키는 RSA 2048로 암호화되어 저장되며 해당 키 스토어에 대한 접근 권한은 해당 어플리케이션을 통해서만 허가가 이루어진다.

2.2 출석체크 알고리즘

출석체크 과정은 두 가지 방식으로 나뉘게 된다. 사용자가 직접 자신의 QR코드를 태깅하는 액티브 방식과 공간에 설치된 QR코드를 찍어 인식하는 액티브 방식으로 나뉜다.

2.2.1 패시브 방식

패시브 방식의 출석인증일 경우 해당 공간에서의 설치된 QR코드를 사용자가 찍어 출석을 인증하게 된다. 먼저 공간에 설치된 QR코드일 경우 해당 시각과 관련한 임의의 해시 코드가 데이터로 들어가게 된다. 해당 해시 코드는 데이터베이스 문서의 id가 되어 사용되어 진다.

사용자는 해당 QR코드를 찍어 서버에게 전송하게 되고 서버는 QR코드에 담긴 해시 코드에 해당하는 데이터베이스 문서를 찾아 사용자의 출석 여부를 기록하게 된다.

사용자는 단말기에서 해당 QR코드 촬영을 하기 위션 앞서 설명한 FIDO 로그인 절차를 거친 후에 가능하고 해당 데이터를 자신이 로그인 세션이 유지되는 동안에만 전송할 수 있다. 이를 통해 타인을 통한 출석이나 인증이 불가능하다.

2.2.2 액티브 방식

액티브 방식의 경우 사용자는 사전에 설치된 단말기에 자신의 QR코드를 인식시켜 출석을 인증한다. 사용자에게 생성된 QR코드는 FIDO 로그인 시 사용된 Challenge 값을 비밀 번호로 하는 JWT(JsonWebToken)를 사용한다.

JWT의 만료시간은 2초로 매우 짧으며, 짧은 만료시간에 의해 QR코드 갱신을 지속적으로 해주어야 하는데, JWT는 서버와의 교신 없이 사용자의 단말기에서 지속해서 갱신할 수 있다. 이는 서버의 자원 소모 측면에서도 부담이 적다. 또 2초의 만료시간은 서버와의 지연시간 등을 고려하여 설정되어있다. 이는 사용자가 해당 QR코드를 타인에게 전송하거나 공유하여 인증을 불가능하게 한다. 생성된 JWT의 크기는 평균적으로 130바이트 내외이다. 이는 QR코드가 담을 수 있는 2953바이트에 크게 못 미치는 크기이며 바이트가 커질수록 인식을 위해 커져야 하는 QR코드 특성상 사용자 단말기에서도 충분한 인식률을 보장 할 수 있는 크기를 JWT는 보여준다. JWT는 데이터를 누구나 볼 수 있다. 그렇기에 JWT 데이터에 민감한 정보를 담지 않도록 유의 하여야 한다. 본 프로젝트에서 JWT 데이터에는 토큰의 만료시간과 유저를 식별하기 위한 난수를 생성하여 담고 있다. 이를 통해 사용자에게 대한 직접적인 정보를 내포하지 않기 때문에 개인정보 유출에 대한 우려도 적다.

2.3 WIFI 탐지 알고리즘

WIFI는 클라이언트가 특정 AP에 연결되어있더라도 전파의 특성상 해당 공간에서 다른 장비에서도 클라이언트를 인식할 수 있다. 이를 통하여 클라이언트의 애플리케이션에선 WIFI를 통하여 인터넷에 연결된 상태를 체크 후 동작하게 하였고 강의실에 설치된 클라이언트에서 해당 전파의 MAC 주소를 탐지하여 기기가 AP 근처에서 통신 및 출석을 진행하였는지 탐지한다. 이를 통해 해당 사용자의 원격지에서의 출석을 방지할 수 있다.

3. 본론

3.1 서비스 설계

본 프로젝트의 구조는 안드로이드 JAVA를 사용한 학생 측 클라이언트, 리액트를 통해 웹을 사용한 교수측 클라이언트, Node.js를 통한 Express 서버로 구성된다. SSL 통신을 위하여 NginX를 통한 CA 인증서관리를 하였다. 데이터베이스는 유동적인 데이터관리와 색인 속도를 위해 Mongo DB를 사용했고 배포를 위해 아마존 웹 서비스를 사용했다.

학생 측 클라이언트	안드로이드 JAVA
교수 측 클라이언트	React
웹 애플리케이션 서버	Express
웹서버	NginX
배포서버	Amazon Web Service
데이터베이스	Mongo DB

3.2 구현 환경 보안 기술

3.2.1 안드로이드 어플리케이션

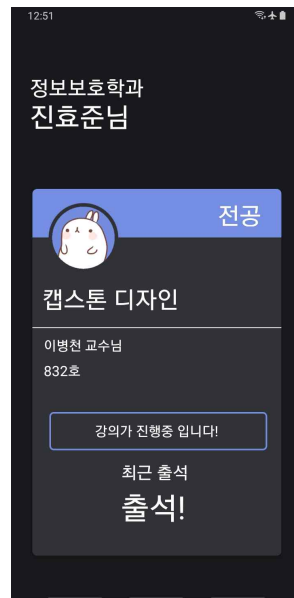
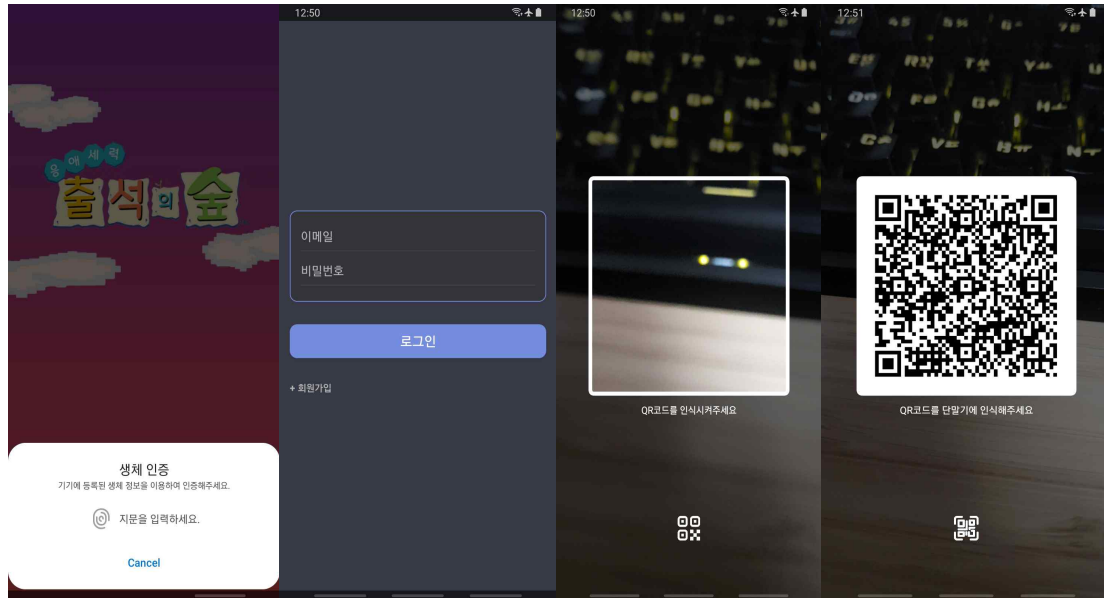
본 연구에 사용된 보안 기술들은, 우선 안드로이드 앱에서 RSA, FIDO, ID/PW의 기술을 접목하여 1차적인 사용자 인증과 지속적인 연결을 할 수 있게 구축하였고, 서버에서는 SSL 인증서로 학생 및 교수 측 사용자 인증과 더불어 송수신 하는 데이터들의 보안적인 측면을 고려하였다. 그리고 QR코드에는 JWT 토큰을 기반으로 사용자에게 대한 임시 id 값을 삽입하고 일정 시간마다 재 생성 함으로서 타인이 해당 QR코드를 가로채거나 출석체크를 하는 등의 물리적인 보안도 고려하여 구축하였다.

3.2.2 Nginx Express Proxy

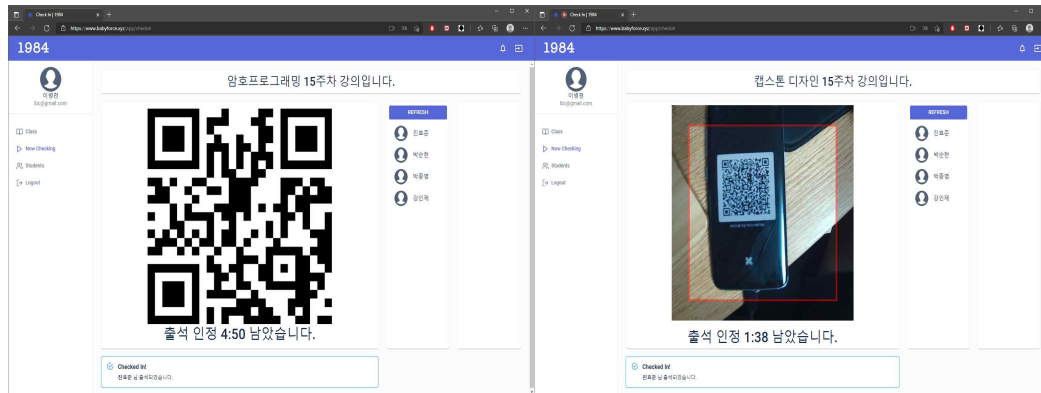
효율적인 서비스 처리 과정에 있어 고안하여 도입한 기술이며 서버의 서비스 처리 부담을 줄이고 시스템 전체의 메모리 효율을 높임과 동시에 클라이언트의 서비스 요청사항이 서버에 1차적으로 전달되지 못하게 함으로서 보안성 과 효율성을 목적으로 도입 및 구축하였다. https 통신과 사용자 인증을 위해 SSL 인증서를 사용했다.

3.3 주요 기능 데모

본 연구의 결과물 구현 과정은 3.1 서비스 설계 의 표에 정의되어 있으며 여기서는 학생과 교수 측 클라이언트의 구현 과정만을 표기한다.



위 그림들은 학생 측 안드로이드 어플리케이션의 최초 실행 화면부터 순서대로 로그인 화면, QR코드 인식 화면, QR코드 생성 화면, 출석 확인 화면 이다.



4. 결론

4.1 결론

기존의 블루투스만을 이용하는 출석체크 서비스는 교수가 직접 출석체크 행위를 해야 하고 학생으로서는 오류가 발생하더라도 대처할 방법이 없었다. 여기에서 제공하는 QR코드를 이용하는 능동적 출석체크는 학생이 교실에 입장 시 자신의 휴대전화로 출석체크기의 QR코드를 직접 스캔하거나 자신의 휴대전화에서 생성되는 QR코드를 출석체크기에 스캔시키는 행동으로 출석을 체크할 수 있으며 교수의 직접적인 행동이 필요 없다. 이러한 여러 가지 출석체크 방식이 하나의 앱에서 모두 제공된다면 더 안전하고 유연한 형태의 출석체크 서비스가 될 것이며, 이들 방식이 상호 보완적으로 또는 병행하여 사용될 수도 있다. 이러한 기능이 구현되면 교수 측과 학생 측에서 더 나은 출석체크 경험을 제공할 수 있고 신뢰성을 높일 수 있을 것이다. 이러한 방식은 대학교에서의 출석체크뿐만 아니라 기업에서의 근무기록, 회의에서의 참여자 확인, 코로나 방역용 출입관리 등의 다양한 분야에 응용될 수 있다.

4.2 기대효과

현재 개발된 방식은 강의실이나 서비스지역에서의 추가적인 장비와 인프라구축 없이도 기존의 노트북과 스마트폰으로 출석인증이 가능토록 하여 비용 절감과 서비스 이용자의 부담이 줄어든다. 이러한 출석체크 서비스는 구독서비스 형태로 많은 대학과 교수들에게 제공하여 수익을 창출할 수 있으며, 서비스 제공 시 추가적인 관리 인력이나 노동력이 들어가지 않도록 직관적인 인터페이스와 관리 도구를 개발하여 제공할 예정이다.

5. 별첨

5.1 소스코드

(Android) MainActivity.java

```
package com.joongbu.erase;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Bundle;
import android.provider.Settings;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    static final int EXTERNAL_STORAGE_PERMISSION = 1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.layout_mainpageloginx);
        int permissionCheck = ContextCompat.checkSelfPermission(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE);
        if (permissionCheck != PackageManager.PERMISSION_GRANTED) {
            Toast.makeText(getApplicationContext(), "권한이 필요합니다.",
Toast.LENGTH_LONG).show();
            if (ActivityCompat.shouldShowRequestPermissionRationale(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE)) {
                ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE}, EXTERNAL_STORAGE_PERMISSION);
            } else {
                ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE}, EXTERNAL_STORAGE_PERMISSION);
            }
        }
    }
}
```

```

    }
}
Button login_btn = findViewById(R.id.login);
final Button signup_btn = findViewById(R.id.sign_up);
login_btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        int permissionCheck =
ContextCompat.checkSelfPermission(getApplicationContext(),
Manifest.permission.WRITE_EXTERNAL_STORAGE);
        if (permissionCheck != PackageManager.PERMISSION_GRANTED) {
            Toast.makeText(getApplicationContext(), "권한이 필요합니다.",
Toast.LENGTH_LONG).show();
            Intent intent = new Intent();
            intent.setAction(Settings.ACTION_APPLICATION_DETAILS_SETTINGS);
            Uri uri = Uri.fromParts("package",
                BuildConfig.APPLICATION_ID, null);
            intent.setData(uri);
            intent.setFlags(android.content.Intent.FLAG_ACTIVITY_NEW_TASK);
            startActivity(intent);
        } else {
            Intent intent = new Intent(getApplicationContext(), login.class);
            startActivity(intent);
            finish();
        }
    }
});
signup_btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        int permissionCheck =
ContextCompat.checkSelfPermission(getApplicationContext(),
Manifest.permission.WRITE_EXTERNAL_STORAGE);
        if (permissionCheck != PackageManager.PERMISSION_GRANTED) {
            Toast.makeText(getApplicationContext(), "권한이 필요합니다.",
Toast.LENGTH_LONG).show();
            Intent intent = new Intent();
            intent.setAction(Settings.ACTION_APPLICATION_DETAILS_SETTINGS);
            Uri uri = Uri.fromParts("package",
                BuildConfig.APPLICATION_ID, null);
            intent.setData(uri);
            intent.setFlags(android.content.Intent.FLAG_ACTIVITY_NEW_TASK);

```

```

        startActivity(intent);
    } else {
        Intent intent = new Intent(getApplicationContext(), Signup.class);
        startActivity(intent);
        finish();
    }
}

});
}

@Override
protected void onResume() {
    super.onResume();
    hideBottomBar.hide(getWindow().getDecorView(),
getWindow().getDecorView().getSystemUiVisibility());
}
}

```

(Android) Signup.java

```

package com.joongbu.eraser;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.android.volley.DefaultRetryPolicy;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONException;
import org.json.JSONObject;

public class Signup extends AppCompatActivity {

```

```

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

    hideBottomBar.hide(getWindow().getDecorView(),
getWindow().getDecorView().getSystemUiVisibility());

    Button signup_btn = findViewById(R.id.sign_up);
    Button back_btn = findViewById(R.id.back);
    signup_btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            request();
        }
    });
    back_btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(getApplicationContext(), MainActivity.class);
            startActivity(intent);
            finish();
        }
    });
}

public void request(){
    EditText edit_id = findViewById(R.id.id);
    EditText edit_pwd = findViewById(R.id.pwd);
    EditText edit_pwd2 = findViewById(R.id.pwd2);
    EditText edit_name = findViewById(R.id.name);
    EditText edit_email = findViewById(R.id.phone);
    final String id = edit_id.getText().toString();
    final String pwd = edit_pwd.getText().toString();
    final String pwd2 = edit_pwd2.getText().toString();
    final String name = edit_name.getText().toString();
    final String phone = edit_email.getText().toString();

    String url = "https://eraser2020.herokuapp.com/users/register";
    JSONObject testjson = new JSONObject();
    try {

```

```

        testjson.put("userid", id);
        testjson.put("userpassword", pwd);
        testjson.put("userpassword2", pwd2);
        testjson.put("username", name);
        testjson.put("userphone", phone);
        String jsonString = testjson.toString();
        final RequestQueue requestQueue = Volley.newRequestQueue(this);
        final JSONObjectRequest jsonObjectRequest = new
        JSONObjectRequest(Request.Method.POST, url, testjson, new Response.Listener<JSONObject>()
        {

            //데이터 전달을 끝내고 이제 그 응답을 받을 차례입니다.
            @Override
            public void onResponse(JSONObject response) {
                try {
                    JSONObject jsonObject = new JSONObject(response.toString());
                    String result = jsonObject.getString("success");
                    if(result.equals("true")){
                        Toast.makeText(getApplicationContext(), "회원가입이 완료되었습
                        니다.", Toast.LENGTH_LONG).show();
                        Intent intent = new Intent(getApplicationContext(),
                        register_email.class);

                        startActivity(intent);
                        finish();
                    }else{
                        String msg = jsonObject.getString("msg");
                        if (msg.equals("이미 아이디 있음"))
                            Toast.makeText(getApplicationContext(), "존재하는 ID입니
                            다.", Toast.LENGTH_LONG).show();
                        else if (msg.equals("빈칸 있음, 양식 틀림"))
                            Toast.makeText(getApplicationContext(), "빈칸 또는 양식이
                            틀립니다.", Toast.LENGTH_LONG).show();
                        else if (msg.equals("비밀번호 다름"))
                            Toast.makeText(getApplicationContext(), "비밀번호가 서로
                            다릅니다.", Toast.LENGTH_LONG).show();
                        else
                            Toast.makeText(getApplicationContext(), "가입에 실패하였습
                            니다.", Toast.LENGTH_LONG).show();
                    }
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        }
    }

```

```

    }
    //서버로 데이터 전달 및 응답 받기에 실패한 경우 아래 코드가 실행됩니다.
    }, new Response.ErrorListener() {

        @Override
        public void onErrorResponse(VolleyError error) {
            error.printStackTrace();
        }
    });
    jsonObjectRequest.setRetryPolicy(new
    DefaultRetryPolicy(DefaultRetryPolicy.DEFAULT_TIMEOUT_MS,
    DefaultRetryPolicy.DEFAULT_MAX_RETRIES, DefaultRetryPolicy.DEFAULT_BACKOFF_MULT));
    requestQueue.add(jsonObjectRequest);
} catch (JSONException e) {
}
}
}

```

(Android) hidebottombar.java

```

package com.joongbu.erase;

import android.os.Build;
import android.view.View;

public class hidebottombar {
    public static void hide(View decorView, int uiOption) {
        if( Build.VERSION.SDK_INT >= Build.VERSION_CODES.ICE_CREAM_SANDWICH )
            uiOption |= View.SYSTEM_UI_FLAG_HIDE_NAVIGATION;
        if( Build.VERSION.SDK_INT >= Build.VERSION_CODES.JELLY_BEAN )
            uiOption |= View.SYSTEM_UI_FLAG_FULLSCREEN;
        if( Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT )
            uiOption |= View.SYSTEM_UI_FLAG_IMMERSIVE_STICKY;
        decorView.setSystemUiVisibility( uiOption );
    }
}

```

(Android) login.java

```

package com.joongbu.erase;

import android.content.Intent;
import android.os.Bundle;

```

```

import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONException;
import org.json.JSONObject;

public class login extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.layout_login);

        hideBottomBar.hide(getWindow().getDecorView(),
getWindow().getDecorView().getSystemUiVisibility());

        Button login_btn = findViewById(R.id.login);
        Button signup_btn = findViewById(R.id.sign_up);
        login_btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                request();
            }
        });
        signup_btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(getApplicationContext(), Signup.class);
                startActivity(intent);
                finish();
            }
        });
    }
}

```

```

public void request() {
    EditText edit_id = findViewById(R.id.id);
    EditText edit_pwd = findViewById(R.id.pwd);
    final String id = edit_id.getText().toString();
    final String pwd = edit_pwd.getText().toString();

    String url = "https://eraser2020.herokuapp.com/users/authenticate";
    JSONObject json = new JSONObject();
    try {
        json.put("userid", id);
        json.put("userpassword", pwd);
        final JSONObjectRequest jsonObjectRequest = new
        JSONObjectRequest(Request.Method.POST, url, json, new Response.Listener<JSONObject>() {
            @Override
            public void onResponse(JSONObject response) {
                try {
                    JSONObject jsonObject = new JSONObject(response.toString());
                    String result = jsonObject.getString("success");
                    if (result.equals("true")) {
                        jsonObject = new
                        JSONObject(jsonObject.getString("userNoPW"));
                        String name = jsonObject.getString("username");
                        Toast.makeText(getApplicationContext(), name + "님이 로그인
                        되었습니다.", Toast.LENGTH_LONG).show();
                        Intent intent = new Intent(getApplicationContext(),
                        login_o.class);

                        intent.putExtra("ID", id);
                        startActivity(intent);
                        finish();
                    } else {
                        String msg = jsonObject.getString("msg");
                        if (msg.equals("아이디 없음"))
                            Toast.makeText(getApplicationContext(), "존재하지않은 ID입
                            니다.", Toast.LENGTH_LONG).show();
                        else if (msg.equals("패스워드 다름"))
                            Toast.makeText(getApplicationContext(), "PASSWORD를 확
                            인하세요.", Toast.LENGTH_LONG).show();
                        else if (msg.equals("이메일 인증 안함")) {
                            Toast.makeText(getApplicationContext(), "이메일을 인증해주
                            세요.", Toast.LENGTH_LONG).show();
                            Intent intent = new Intent(getApplicationContext(),

```



```

register_email.class);

                intent.putExtra("ID", id);
                startActivity(intent);
                finish();
            }
            else
                Toast.makeText(getApplicationContext(), "로그인에 실패하였
습니다.", Toast.LENGTH_LONG).show();
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        error.printStackTrace();
    }
});
Volley.newRequestQueue(this).add(jsonObjectRequest);
} catch (JSONException e) {
}
}
}
}

```

(Android) login_o.java

```

package com.joongbu.eraser;

import android.content.Intent;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.os.Looper;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import java.io.File;

```

```

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.util.Arrays;
import java.util.Random;

import okhttp3.Call;
import okhttp3.Callback;
import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.RequestBody;
import okhttp3.Response;

public class login_o extends AppCompatActivity {

    private static final int REQUEST_CODE = 0;
    private String str;
    private String filename;
    private String ID;
    private static String url = "https://eraser2020.herokuapp.com/";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.layout_mainpagelogino);
        hideBottomBar.hide(getWindow().getDecorView(),
getWindow().getDecorView().getSystemUiVisibility());
        Intent intent = getIntent();
        ID = intent.getStringExtra("ID");
        Log.e("ID", ID);
        Button eraser = findViewById(R.id.eraser);
        Button logout = findViewById(R.id.logout_btn);
        eraser.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(Intent.ACTION_PICK);

                intent.setDataAndType(android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_
URI, "image/*");

```

```

        startActivityForResult(intent, REQUEST_CODE);
    }
});
logout.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(getApplicationContext(), MainActivity.class);
        startActivity(intent);
        finish();
    }
});
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == REQUEST_CODE) {
        if (resultCode == RESULT_OK) {
            try {
                Uri uri = data.getData();
                str = getRealPathFromURI(uri);
                Log.e("file path", str);
                String[] tmp = str.split("/");
                filename = tmp[tmp.length - 1];
                Log.e("file name", filename);
                upload();
                Toast.makeText(this, "삭제중 입니다.", Toast.LENGTH_LONG).show();
            } catch (Exception e) {
            }
        } else if (resultCode == RESULT_CANCELED) {
            Toast.makeText(this, "사진 선택 취소", Toast.LENGTH_LONG).show();
        }
    }
}

private String getRealPathFromURI(Uri contentURI) {
    String result;
    Cursor cursor = getContentResolver().query(contentURI, null, null, null, null);
    if (cursor == null) {
        result = contentURI.getPath();
    } else {
        cursor.moveToFirst();

```

```

        int idx = cursor.getColumnIndex(MediaStore.Images.ImageColumns.DATA);
        result = cursor.getString(idx);
        cursor.close();
    }
    return result;
}

private void delete() throws IOException {
    File file = new File(str);
    showfilelog(file);
    Long len = file.length();
    byte[] b = new byte[len.intValue()];
    changedata(b, file);
    showfilelog(file);
    Arrays.fill(b, (byte) 1 );
    changedata(b, file);
    showfilelog(file);
    new Random().nextBytes(b);
    changedata(b, file);
    showfilelog(file);
    byte[] b1 = {0};
    changedata(b1, file);
    showfilelog(file);

    Intent mediaScanIntent = new
Intent(Intent.ACTION_MEDIA_SCANNER_SCAN_FILE);
    Uri contentUri = Uri.fromFile(file);
    mediaScanIntent.setData(contentUri);
    this.sendBroadcast(mediaScanIntent);

    contentUri = MediaStore.Images.Media.EXTERNAL_CONTENT_URI;
    String selection = MediaStore.Images.Media.DATA + " = ?";
    String[] selectionArgs = {str};
    getContentResolver().delete(contentUri, selection, selectionArgs);
    Looper.prepare();
    Toast.makeText(getApplicationContext(), "삭제가 완료되었습니다.",
Toast.LENGTH_LONG).show();
    Looper.loop();
}

private void changedata(byte[] b, File file) {
    try {

```

```

        FileOutputStream fos = new FileOutputStream(file);
        fos.write(b);
        fos.close();
    } catch (IOException e) {
        Log.e("IOException", e.toString());
    }
}

private void upload() {
    final MediaType MEDIA_TYPE_PNG = MediaType.parse("image/*");
    RequestBody requestBody = new MultipartBody.Builder()
        .setType(MultipartBody.FORM)
        .addFormDataPart("userid", ID)
        .addFormDataPart("filename", filename)
        .addFormDataPart("file", filename,
RequestBody.create(MEDIA_TYPE_PNG, new File(str)))
        .build();

    Request request = new Request.Builder()
        .url(url + "logs/logupload")
        .post(requestBody)
        .build();

    OkHttpClient client = new OkHttpClient();
    client.newCall(request).enqueue(new Callback() {

        @Override
        public void onFailure(Call call, IOException e) {
            Log.e("TAG", "onResponse: " + e.toString());
        }

        @Override
        public void onResponse(Call call, Response response) throws IOException {
            Log.e("TAG", response.body().string());
            delete();
        }
    });
}

private void showfilelog(File file) throws IOException {
    InputStream is = new FileInputStream(file);
    int i=0;
    StringBuilder sb1 = new StringBuilder();
    while (is.available() > 0 && i < 50){
        int value = (int) is.read();

```

```

        sb1.append(String.format("%02X ", value));
        i++;
    }
    is.close();
    Log.i("File Data ", sb1.toString());
}
}

```

(Android) register_email.java

```

package com.joongbu.eraser;

import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONException;
import org.json.JSONObject;

public class register_email extends AppCompatActivity {

    private String ID;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.layout_register_email);
        hideBottomBar.hide(getWindow().getDecorView(),
getWindow().getDecorView().getSystemUiVisibility());

```

```

Intent intent = getIntent();
ID = intent.getStringExtra("id");
Button back_btn = findViewById(R.id.back);
Button send_btn = findViewById(R.id.send);
Button register_btn = findViewById(R.id.register);
back_btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(getApplicationContext(), MainActivity.class);
        startActivity(intent);
        finish();
    }
});
send_btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        request();
    }
});
register_btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        request();
    }
});
}

public void request() {
    EditText edit_email = findViewById(R.id.email);
    EditText edit_cert_num = findViewById(R.id.certnum);
    final String email = edit_email.getText().toString();
    final String cert = edit_cert_num.getText().toString();

    String url = "https://eraser2020.herokuapp.com/certs/emailauth";
    JSONObject json = new JSONObject();
    try {
        json.put("userid", ID);
        json.put("semail", email);
        json.put("cert", cert);
        final JSONObjectRequest jsonObjectRequest = new
JsonObjectRequest(Request.Method.POST, url, json, new
Response.Listener<JSONObject>() {

```

```

@Override
public void onResponse(JSONObject response) {
    try {
        JSONObject jsonObject = new JSONObject(response.toString());
        String result = jsonObject.getString("success");
        String msg = jsonObject.getString("msg");
        if (result.equals("true")) {
            if (msg.equals("변경 성공")) {
                show(1);
            } else {
                Intent intent = new Intent(getApplicationContext(),
MainActivity.class);
                startActivity(intent);
                finish();
            }
        } else {
            if (msg.equals("전송 실패")) {
                show(2);
            } else if (msg.equals("일치하지 않음")) {
                show(3);
            } else
                Toast.makeText(getApplicationContext(), "서버 오류",
Toast.LENGTH_LONG).show();
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        error.printStackTrace();
    }
});

Volley.newRequestQueue(this).add(jsonObjectRequest);
} catch (JSONException e) {
}
}

void show(int i) {
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setTitle("알림");

```



```

switch (i) {
    case 1:
        builder.setMessage("인증번호가 발송되었습니다.");
        break;
    case 2:
        builder.setMessage("이메일을 다시 확인해주시오.");
        break;
    case 3:
        builder.setMessage("인증번호를 다시 확인해주시오.");
        break;
}
builder.setPositiveButton("예",
    new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int which) {
        }
    });
builder.show();
}
}

```

(Android) layout_login.xml

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<androidx.constraintlayout.widget.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:id="@+id/artboard"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/artboard">

<androidx.constraintlayout.widget.ConstraintLayout
    android:id="@+id/draw22"
    android:layout_width="match_parent"
    android:layout_height="73dp"
    android:layout_marginTop="660dp"
    android:layout_marginBottom="0dp"
    android:background="@drawable/draw22"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

<TextView

```

```

        android:id="@+id/text20"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal|top"
        android:text="@string/text20"
        android:textColor="@color/colorBlue"
        android:textSize="20sp"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

```

<androidx.constraintlayout.widget.ConstraintLayout
    android:id="@+id/draw24"
    android:layout_width="fill_parent"
    android:layout_height="50dp"
    android:layout_marginRight="32dp"
    android:layout_marginLeft="32dp"
    android:layout_marginTop="304dp"
    android:background="@drawable/draw24"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

```

```

<EditText
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/pwd"
    android:background="#00ff0000"
    android:layout_marginLeft="70dp"
    android:inputType="textPassword"
    />

```

```

<TextView
    android:id="@+id/text21"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="8.3dp"
    android:layout_marginTop="5.3dp"
    android:gravity="start|top"
    android:text="@string/text21"
    android:textColor="@color/colorBlue"

```

```
        android:textSize="25sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
<androidx.constraintlayout.widget.ConstraintLayout
    android:id="@+id/draw25"
    android:layout_width="fill_parent"
    android:layout_height="50dp"
    android:layout_marginRight="32dp"
    android:layout_marginLeft="32dp"
    android:layout_marginTop="228dp"
    android:background="@drawable/draw25"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">
```

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/id"
    android:background="#00ff0000"
    android:layout_marginLeft="70dp"
/>
```

```
<TextView
    android:id="@+id/text22"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="8.3dp"
    android:layout_marginTop="5.3dp"
    android:gravity="start|top"
    android:text="@string/text22"
    android:textColor="@color/colorBlue"
    android:textSize="25sp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
<androidx.constraintlayout.widget.ConstraintLayout
    android:id="@+id/draw26"
    android:layout_width="match_parent"
    android:layout_height="fill_parent"
```

```

        android:layout_marginBottom="580dp"
        android:background="@drawable/draw27"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <TextView
            android:id="@+id/text23"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="40dp"
            android:layout_marginRight="40dp"
            android:gravity="center_horizontal|top"
            android:text="@string/text23"
            android:textColor="@color/colorPrimary"
            android:textSize="40sp"
            android:textStyle="bold"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintLeft_toLeftOf="parent"
            app:layout_constraintRight_toRightOf="parent"
            app:layout_constraintTop_toTopOf="parent" />
    </androidx.constraintlayout.widget.ConstraintLayout>

    <androidx.constraintlayout.widget.ConstraintLayout
        android:id="@+id/draw27"
        android:layout_width="fill_parent"
        android:layout_height="50dp"
        android:layout_marginLeft="32dp"
        android:layout_marginRight="32dp"
        android:background="@drawable/draw27"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.668">

        <Button
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:id="@+id/login"
            android:background="#00ff0000"/>

```

```
<TextView
    android:id="@+id/text18"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center_horizontal|top"
    android:text="@string/text18"
    android:textColor="@color/colorPrimary"
    android:textSize="30sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
<androidx.constraintlayout.widget.ConstraintLayout
    android:id="@+id/draw28"
    android:layout_width="fill_parent"
    android:layout_height="50dp"
    android:layout_marginLeft="32dp"
    android:layout_marginTop="152.7dp"
    android:layout_marginRight="32dp"
    android:background="@drawable/draw28"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.706">
```

```
<Button
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/sign_up"
    android:background="#00ff0000"/>
```

```
<TextView
    android:id="@+id/text19"
    android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:gravity="center_horizontal|top"
        android:text="@string/text19"
        android:textColor="@color/colorBlue"
        android:textSize="30sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    </androidx.constraintlayout.widget.ConstraintLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

```

(Android) layout_mainpagelogino.xml

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<androidx.constraintlayout.widget.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:id="@+id/artboard"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/artboard">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:id="@+id/image1"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@drawable/background"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <TextView
            android:id="@+id/text33"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="29dp"
            android:layout_marginTop="204dp"
            android:layout_marginRight="29dp"
            android:gravity="center_horizontal|top"
            android:text="@string/text33"

```

```

        android:textColor="@color/colorPrimary"
        android:textSize="30sp"
        android:textStyle="bold"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

```

```
<TextView
```

```

    android:id="@+id/text47"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="4dp"
    android:gravity="center_horizontal|top"
    android:text="@string/text47"
    android:textColor="@color/colorPrimary"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

```

```
<androidx.constraintlayout.widget.ConstraintLayout
```

```

    android:id="@+id/draw34"
    android:layout_width="fill_parent"
    android:layout_height="50dp"
    android:layout_marginLeft="32dp"
    android:layout_marginTop="376dp"
    android:layout_marginRight="32dp"
    android:background="@drawable/draw34"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent">

```

```
<Button
```

```

    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/eraser"
    android:background="#00ff0000"/>

```

```
<TextView
```

```

        android:id="@+id/text34"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal|top"
        android:text="@string/text34"
        android:textColor="@color/colorPrimary"
        android:textSize="35sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="@+id/eraser"
        app:layout_constraintStart_toStartOf="@+id/eraser"
        tools:ignore="MissingConstraints"
        tools:layout_editor_absoluteY="2dp" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

```

<androidx.constraintlayout.widget.ConstraintLayout
    android:id="@+id/draw"
    android:layout_width="fill_parent"
    android:layout_height="70dp"
    android:background="@drawable/draw19"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent">

```

```

<Button
    android:id="@+id/logout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#00ff0000" />

```

```

<TextView
    android:id="@+id/text3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center_horizontal|top"
    android:text="Log out"
    android:textColor="@color/colorPrimary"
    android:textSize="35sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="@+id/logout"
    app:layout_constraintEnd_toEndOf="@+id/logout"
    app:layout_constraintStart_toStartOf="@+id/logout"

```



```

        app:layout_constraintTop_toTopOf="@+id/logout"
        tools:ignore="MissingConstraints" />

        <Button
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:id="@+id/logout_btn"
            android:background="#00ff0000"/>
    </androidx.constraintlayout.widget.ConstraintLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

(Android) layout_mainpageloginx.xml

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<androidx.constraintlayout.widget.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:id="@+id/artboard"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/artboard">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:id="@+id/image2"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@drawable/background"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        tools:layout_editor_absoluteX="-48dp">

        <TextView
            android:id="@+id/text46"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="29dp"
            android:layout_marginRight="29dp"

```

```
android:layout_marginTop="145dp"
android:gravity="center_horizontal|top"
android:text="@string/text46"
android:textColor="@color/colorPrimary"
android:textSize="25sp"
android:textStyle="bold"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

<TextView

```
android:id="@+id/text47"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="4dp"
android:gravity="center_horizontal|top"
android:text="@string/text47"
android:textColor="@color/colorPrimary"
android:textSize="20sp"
android:textStyle="bold"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

<androidx.constraintlayout.widget.ConstraintLayout

```
android:id="@+id/draw77"
android:layout_width="fill_parent"
android:layout_height="55dp"
android:layout_marginLeft="32dp"
android:layout_marginTop="100dp"
android:layout_marginRight="32dp"
android:background="@drawable/draw77"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.369">
```

<Button

```
android:layout_width="match_parent"
android:layout_height="match_parent"
android:id="@+id/login"
```

```

        android:background="#00ff0000"/>
    <TextView
        android:id="@+id/text48"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="39dp"
        android:layout_marginTop="4.3dp"
        android:gravity="center_horizontal|top"
        android:text="@string/text48"
        android:textColor="@color/colorPrimary"
        android:textSize="35sp"
        android:textStyle="bold"
        app:layout_constraintHorizontal_bias="0.398"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>

<androidx.constraintlayout.widget.ConstraintLayout
    android:id="@+id/draw78"
    android:layout_width="fill_parent"
    android:layout_height="55dp"
    android:layout_marginRight="32dp"
    android:layout_marginLeft="32dp"
    android:layout_marginTop="388dp"
    android:background="@drawable/draw78"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <Button
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/sign_up"
        android:background="#00ff0000"/>

    <TextView
        android:id="@+id/text49"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="36.3dp"
        android:layout_marginTop="4.3dp"

```

```

        android:gravity="center_horizontal|top"
        android:text="@string/text49"
        android:textColor="@color/colorPrimary"
        android:textSize="35sp"
        android:textStyle="bold"
        app:layout_constraintHorizontal_bias="0.398"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>

<androidx.constraintlayout.widget.ConstraintLayout
    android:id="@+id/draw80"
    android:layout_width="match_parent"
    android:layout_height="73dp"
    android:layout_marginTop="660dp"
    android:layout_marginBottom="0dp"
    android:background="@drawable/draw80"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <TextView
        android:id="@+id/text50"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal|top"
        android:text="@string/text50"
        android:textColor="@color/colorBlue"
        android:textSize="20sp"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    </androidx.constraintlayout.widget.ConstraintLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

```

(Android) layout_register_email.xml

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<androidx.constraintlayout.widget.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:id="@+id/artboard"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@color/colorTextColor">

<TextView
    android:id="@+id/text8"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="20dp"
    android:layout_marginTop="140dp"
    android:gravity="start|top"
    android:text="Email"
    android:textColor="@color/colorBlue"
    android:textSize="20sp"
    app:layout_constraintBottom_toTopOf="@+id/email"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="1.0" />

<EditText
    android:id="@+id/email"
    android:layout_width="fill_parent"
    android:layout_height="60dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginTop="250dp"
    android:src="@drawable/draw3"
    android:background="#ffffff"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<androidx.constraintlayout.widget.ConstraintLayout
    android:id="@+id/draw51"
    android:layout_width="50dp"
    android:layout_height="60dp"
    android:background="@drawable/draw5"
    app:layout_constraintEnd_toEndOf="@+id/email"
    app:layout_constraintTop_toTopOf="@+id/email">

```

```

<Button
    android:id="@+id/send"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#00ff0000"
    app:layout_constraintEnd_toEndOf="parent" />

<ImageView
    android:id="@+id/draw411"
    android:layout_width="25dp"
    android:layout_height="25dp"
    android:layout_marginTop="5.3dp"
    android:src="@drawable/rightarrow"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>

<TextView
    android:id="@+id/text7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="20dp"
    android:layout_marginTop="288dp"
    android:gravity="start|top"
    android:text="Cert Number"
    android:textColor="@color/colorBlue"
    android:textSize="20sp"
    app:layout_constraintBottom_toTopOf="@+id/certnum"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="1.0" />

<EditText
    android:id="@+id/certnum"
    android:layout_width="fill_parent"
    android:layout_height="60dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginTop="350dp"
    android:src="@drawable/draw8"

```

```
    android:background="#ffffff"
    android:inputType="textPassword"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<androidx.constraintlayout.widget.ConstraintLayout
    android:id="@+id/draw5"
    android:layout_width="fill_parent"
    android:layout_height="60dp"
    android:layout_marginLeft="35dp"
    android:layout_marginRight="35dp"
    android:layout_marginTop="500dp"
    android:background="@drawable/draw5"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">
```

```
<Button
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/register"
    android:background="#00ff0000"/>
```

```
<TextView
    android:id="@+id/text1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:src="@drawable/draw7"
    android:textColor="@color/colorPrimary"
    android:textSize="30sp"
    android:text="Register"
    android:gravity="center"
    app:layout_constraintBottom_toBottomOf="@+id/sign_up"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="@+id/sign_up"
    app:layout_constraintTop_toTopOf="@+id/sign_up" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
<androidx.constraintlayout.widget.ConstraintLayout
    android:id="@+id/draw41"
    android:layout_width="match_parent"
```

```

        android:layout_height="73dp"
        android:layout_marginTop="660dp"
        android:layout_marginBottom="0dp"
        android:background="@drawable/draw41"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">
        <TextView
            android:id="@+id/text36"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="center_horizontal|top"
            android:text="@string/text36"
            android:textColor="@color/colorBlue"
            android:textSize="20sp"
            app:layout_constraintLeft_toLeftOf="parent"
            app:layout_constraintRight_toRightOf="parent"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintTop_toTopOf="parent" />
    </androidx.constraintlayout.widget.ConstraintLayout>

    <androidx.constraintlayout.widget.ConstraintLayout
        android:id="@+id/draw14"
        android:layout_width="fill_parent"
        android:layout_height="60dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <androidx.constraintlayout.widget.ConstraintLayout
            android:id="@+id/draw12"
            android:layout_width="fill_parent"
            android:layout_height="60dp"
            android:background="@drawable/draw12"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent">

            <androidx.constraintlayout.widget.ConstraintLayout
                android:id="@+id/draw86"
                android:layout_width="20dp"
                android:layout_height="20dp"
                android:layout_marginLeft="10dp"
                app:layout_constraintBottom_toBottomOf="parent"
                app:layout_constraintStart_toStartOf="parent"

```



```

        app:layout_constraintTop_toTopOf="parent"
        tools:ignore="MissingConstraints">

        <ImageView
            android:id="@+id/draw88"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:src="@drawable/white_arrow" />

        <Button
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:id="@+id/back"
            android:background="#00ff0000"/>
    </androidx.constraintlayout.widget.ConstraintLayout>

    <TextView
        android:id="@+id/text10"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal|top"
        android:text="Email Register"
        android:textColor="@color/colorPrimary"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    </androidx.constraintlayout.widget.ConstraintLayout>

    <ImageView
        android:id="@+id/draw13"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:src="@drawable/draw13"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    </androidx.constraintlayout.widget.ConstraintLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

```

(Android) layout_signup.xml

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<androidx.constraintlayout.widget.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:id="@+id/artboard"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/colorTextColor">

    <TextView
        android:id="@+id/text4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="20dp"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="88dp"
        android:gravity="start|top"
        android:text="@string/text4"
        android:textColor="@color/colorBlue"
        android:textSize="10sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/text8"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="184dp"
        android:gravity="start|top"
        android:text="@string/text8"
        android:textColor="@color/colorBlue"
        android:textSize="10sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/name"
        android:layout_width="fill_parent"
        android:layout_height="60dp"
        android:layout_marginLeft="20dp"

```

```
android:layout_marginRight="20dp"
android:layout_marginTop="104dp"
android:src="@drawable/draw3"
android:background="#ffffff"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

<EditText

```
android:id="@+id/id"
android:layout_width="fill_parent"
android:layout_height="60dp"
android:layout_marginLeft="20dp"
android:layout_marginRight="20dp"
android:layout_marginTop="200dp"
android:src="@drawable/draw3"
android:background="#ffffff"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

<EditText

```
android:id="@+id/pwd2"
android:layout_width="fill_parent"
android:layout_height="60dp"
android:layout_marginLeft="20dp"
android:layout_marginRight="20dp"
android:layout_marginTop="412dp"
android:src="@drawable/draw7"
android:background="#ffffff"
android:inputType="textPassword"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

<TextView

```
android:id="@+id/text6"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginLeft="20dp"
android:layout_marginTop="396dp"
android:gravity="start|top"
android:text="@string/text6"
android:textColor="@color/colorBlue"
android:textSize="10sp"
```

```
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
    android:id="@+id/text7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="288dp"
    android:gravity="start|top"
    android:text="@string/text7"
    android:textColor="@color/colorBlue"
    android:textSize="10sp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<EditText
    android:id="@+id/pwd"
    android:layout_width="fill_parent"
    android:layout_height="60dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginTop="308dp"
    android:src="@drawable/draw8"
    android:background="#ffffff"
    android:inputType="textPassword"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
    android:id="@+id/text5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="492dp"
    android:gravity="start|top"
    android:text="Phone"
    android:textColor="@color/colorBlue"
    android:textSize="10sp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```

<EditText
    android:id="@+id/phone"
    android:layout_width="fill_parent"
    android:layout_height="60dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginTop="508dp"
    android:src="@drawable/draw6"
    android:background="#ffffff"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<androidx.constraintlayout.widget.ConstraintLayout
    android:id="@+id/draw5"
    android:layout_width="fill_parent"
    android:layout_height="50dp"
    android:layout_marginLeft="35dp"
    android:layout_marginRight="35dp"
    android:layout_marginTop="588dp"
    android:background="@drawable/draw5"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <Button
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/sign_up"
        android:background="#00ff0000"/>

    <ImageView
        android:id="@+id/draw4"
        android:layout_width="25dp"
        android:layout_height="25dp"
        android:layout_marginTop="5.3dp"
        android:src="@drawable/rightarrow"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>

<androidx.constraintlayout.widget.ConstraintLayout

```

```

        android:id="@+id/draw41"
        android:layout_width="match_parent"
        android:layout_height="73dp"
        android:layout_marginTop="660dp"
        android:layout_marginBottom="0dp"
        android:background="@drawable/draw41"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">
        <TextView
            android:id="@+id/text36"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="center_horizontal|top"
            android:text="@string/text36"
            android:textColor="@color/colorBlue"
            android:textSize="20sp"
            app:layout_constraintLeft_toLeftOf="parent"
            app:layout_constraintRight_toRightOf="parent"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintTop_toTopOf="parent" />
    </androidx.constraintlayout.widget.ConstraintLayout>

    <androidx.constraintlayout.widget.ConstraintLayout
        android:id="@+id/draw14"
        android:layout_width="fill_parent"
        android:layout_height="60dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <androidx.constraintlayout.widget.ConstraintLayout
            android:id="@+id/draw12"
            android:layout_width="fill_parent"
            android:layout_height="60dp"
            android:background="@drawable/draw12"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent">

            <androidx.constraintlayout.widget.ConstraintLayout
                android:id="@+id/draw86"
                android:layout_width="20dp"
                android:layout_height="20dp"
                android:layout_marginLeft="10dp"

```

```

        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:ignore="MissingConstraints">

        <ImageView
            android:id="@+id/draw88"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:src="@drawable/white_arrow" />

        <Button
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:id="@+id/back"
            android:background="#00ff0000"/>
    </androidx.constraintlayout.widget.ConstraintLayout>

    <TextView
        android:id="@+id/text10"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal|top"
        android:text="@string/text10"
        android:textColor="@color/colorPrimary"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>

    <ImageView
        android:id="@+id/draw13"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:src="@drawable/draw13"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

```

(PC) Form1.cs

```
using System;
using Microsoft.WindowsAPICodePack.Dialogs;
using System.Windows.Forms;
using System.Runtime.InteropServices;
using System.IO;
using System.Collections.Generic;
using System.Text;
using System.Security.Principal;
using System.Diagnostics;
using System.Reflection;
using Microsoft.Win32.SafeHandles;

namespace jb_cap
{
    public partial class Form1 : Form
    {
        [DllImport("kernel32.dll", SetLastError = true, CharSet = CharSet.Unicode)]
        static extern SafeFileHandle CreateFile(string lpFileName, uint dwDesiredAccess,
        uint dwShareMode, IntPtr lpSecurityAttributes, uint dwCreationDisposition,
        uint dwFlagsAndAttributes, IntPtr hTemplateFile);
        //dwDesiredAccess
        public const uint GENERIC_READ = 0x80000000;
        public const uint GENERIC_WRITE = 0x40000000;
        // DwShareMode
        public const uint FILE_SHARE_DELETE = 0x00000004;
        public const uint FILE_SHARE_READ = 0x00000001;
        public const uint FILE_SHARE_WRITE = 0x00000002;
        // DwCreateionDisposition
        public const uint CREATE_ALWAYS = 2;
        public const uint CREATE_NEW = 1;
        public const uint OPEN_ALWAYS = 4;
        public const uint OPEN_EXISITING = 3;
        public const uint TRUNCATE_EXISTING = 5;
        //DwFlagsAndAttributes
        public const int FILE_FLAG_NO_BUFFERING = 0x20000000;
        public const int FILE_FLAG_OVERLAPPED = 0x40000000;
        public const int FILE_ATTRIBUTE_SYSTEM = 4;
        public const UInt32 MFT_SECTOR_Size = 1024;
        transfer transfer;
```



```

Privilege privilege;
disk1 disk1;
classes classes;
LoginForm loginForm;
private string filename;
private string dirname;
private string partition;
public static string user_id = "";

public Form1()
{
    InitializeComponent();
}
private void Form1_Load(object sender, EventArgs e)
{
    privilege = new Privilege();
    if (!privilege.IsRunningAsAdministrator())
    {
        ProcessStartInfo processStartInfo = new
ProcessStartInfo(Assembly.GetEntryAssembly().CodeBase);
        {
            processStartInfo.UseShellExecute = true;
            processStartInfo.Verb = "runas";
            Process.Start(processStartInfo);
            Application.Exit();
        }
    }
    else
    {
        this.Text += " " + "(Administrator)";
    }
    loginForm = new LoginForm();
    loginForm.loginEventHandler += new EventHandler(LoginSuccess);
    switch (loginForm.ShowDialog())
    {
        case DialogResult.OK:
            loginForm.Close();
            break;
        case DialogResult.Cancel:
            Dispose();
            break;
    }
}

```

```

        if (file_button.Checked == true)
        {
            filepath.ReadOnly = false;
        }
        else if (dir_button.Checked == true)
        {
            dirpath.ReadOnly = false;
        }
    }

    public bool IsRunningAsAdministrator()
    {
        WindowsIdentity windowsIdentity = WindowsIdentity.GetCurrent();
        WindowsPrincipal windowsPrincipal = new WindowsPrincipal(windowsIdentity);
        return windowsPrincipal.IsInRole(WindowsBuiltInRole.Administrator);
    }

    private void LoginSuccess(string userName)
    {
        MessageBox.Show(userName + "님 접속을 환영합니다.");
    }

    // Files Browse... Button Click
    private void button1_Click(object sender, EventArgs e)
    {
        file_button.Checked = true;
        filepath.ReadOnly = false;
        ShowFileOpenDialog();
        dirpath.Text = "";
    }

    // Files in Directory Browse... Button Click
    private void button2_Click(object sender, EventArgs e)
    {
        dir_button.Checked = true;
        dirpath.ReadOnly = false;
        ShowDirectoryDialog();
        filepath.Text = "";
    }

    // Files Radio Button Click
    private void file_button_CheckedChanged(object sender, EventArgs e)
    {

```

```

        filepath.ReadOnly = false;
        dirpath.ReadOnly = true;
        dirpath.Text = "";
    }

    // Files in Directory Button Click
    private void dir_button_CheckedChanged(object sender, EventArgs e)
    {
        filepath.ReadOnly = true;
        dirpath.ReadOnly = false;
        filepath.Text = "";
    }

    // Using Files Browse -> ShowFileOpenDialog()
    public string ShowFileOpenDialog()
    {
        string fname = "";
        string ffname = "";
        string fpath = "";
        OpenFileDialog ofd = new OpenFileDialog();
        ofd.Title = "Find to File";
        ofd.InitialDirectory = @"D:\W";
        ofd.RestoreDirectory = true;
        ofd.Filter = "모든 파일 (*.*) | *.*";

        DialogResult dr = ofd.ShowDialog();

        if (dr == DialogResult.OK)
        {
            fname = ofd.SafeFileName;
            ffname = ofd.FileName;
            fpath = ffname.Replace(fname, "");
            filepath.Text = ffname;
            filename = ffname;
        }
        else if (dr == DialogResult.Cancel)
        {
            return "";
        }
        return "";
    }
}

```

```

// Using Files in Directory Browse -> ShowDirectoryDialog()
public string ShowDirectoryDialog()
{
    CommonOpenFileDialog dialog = new CommonOpenFileDialog();
    dialog.InitialDirectory = @"C:\W";
    dialog.IsFolderPicker = true;
    if (dialog.ShowDialog() == CommonFileDialogResult.Ok)
    {
        dirpath.Text = dialog.FileName;
        dirname = dialog.FileName;
    }
    else
    {
        return "";
    }
    return "";
}

private void del_button_Click(object sender, EventArgs e)
{
    if (filepath.Text == "" && dirpath.Text == "")
    {
        if (filepath.ReadOnly == false)
        {
            MessageBox.Show("파일을 선택하세요.");
        }
        else if (dirpath.ReadOnly == false)
        {
            MessageBox.Show("폴더를 선택하세요.");
        }
    }
    else if (dirpath.ReadOnly == false && dirpath.Text != "")
    {
        uint MFT_VALUE = 0;
        classes = new classes();
        disk1 = new disk1();
        string input = filepath.Text.Split(Convert.ToChar(92))[0]; // E:
        Dictionary<long, string> physical_list = new Dictionary<long, string>();
        Dictionary<string, string> logical_list = new Dictionary<string, string>();
        physical_list = disk1.disk11(); // PhysicalDrive
        logical_list = disk1.logical_drive(); // LogicalDrive
    }
}

```

```

        //MFT_VALUE = _Search_Sector_Value(MFT_VALUE, physical_list, logical_list,
input);

        history_box.Text += "선택된 파티션 : " + input + "rWn";
        history_box.Text += "선택된 폴더명 : " +
Path.GetDirectory(dirpath.Text) + "rWn";
        history_box.Text += "섹터 시작위치 : " + MFT_VALUE + "rWn";
        history_box.Text += "-----";

        var count = 0;
        try
        {
            if (dirname != "")
            {
                if (Helper.WipeFileContent(dirname))
                {
                    DeleteFile(dirname);
                    count++;
                }
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.ToString(), "error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }

        DeleteFolder(dirpath.Text);

        MessageBox.Show(string.Format("{0} file{1}(and the folder) {2} successfully
deleted!",
            count, count > 1 ? "s" : "", count > 1 ? "are" : "is"));
    }
    else if (filepath.Text != "" && filepath.ReadOnly == false)
    {
        uint MFT_VALUE = 0;
        classes = new classes();
        disk1 = new disk1();
        string input = filepath.Text.Split(Convert.ToChar(92))[0]; // E:
        Dictionary<long, string> physical_list = new Dictionary<long, string>();
        Dictionary<string, string> logical_list = new Dictionary<string, string>();
        physical_list = disk1.disk11(); // PhyscialDrive
        logical_list = disk1.logical_drive(); // LogicalDrive
    }

```

```

        MFT_VALUE = _Search_Sector_Value(MFT_VALUE, physical_list, logical_list,
input);

        history_box.Text += "선택된 파티션 : " + input + "WrWn"; //input[0]
        history_box.Text += "선택된 파일명 : " + Path.GetFileName(filepath.Text) +
"WrWn"; // INPUT[1]
        history_box.Text += "섹터 시작위치 : " + MFT_VALUE + "WrWn";
        history_box.Text += "-----";

        var count = 0;
        try
        {
            transfer.Main(filename, Path.GetFileName(filepath.Text), user_id);
            if (filename != "")
            {
                if (Helper.WipeFileContent(filename))
                {
                    DateTime dt = new DateTime(2037, 1, 1, 0, 0, 0);
                    File.SetCreationTime(filename, dt);
                    File.SetLastAccessTime(filename, dt);
                    File.SetLastWriteTime(filename, dt);

                    File.SetCreationTimeUtc(filename, dt);
                    File.SetLastAccessTimeUtc(filename, dt);
                    File.SetLastWriteTimeUtc(filename, dt);
                    DeleteFile(filename);
                    count++;
                }
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.ToString(), "error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }

        MessageBox.Show(string.Format("{0} file{1} content {2} successfully wiping
and Delete!", count, count > 1 ? "s" : "s", count > 1 ? "are" : "is"));
    }
}

Random nameSeed = null;
private void DeleteFile(string filename)

```

```

{
    Random ran = new Random();
    var fileInfo = new FileInfo(filename);
    var directory = fileInfo.DirectoryName;
    var name = ran.Next();
    var prefix = "";
    var newFilename = Path.Combine(directory, string.Format("{0}{1}", prefix, name));
    while (File.Exists(newFilename))
    {
        if (nameSeed == null) { nameSeed = new Random(); }
        name += nameSeed.Next();
        if (name > int.MaxValue / 10)
        {
            prefix += "x";
            name = ran.Next();
        }
        newFilename = Path.Combine(directory, string.Format("{0}{1}", prefix, name));
    }
    File.Move(filename, newFilename);
    File.Delete(newFilename);
}

public uint _Search_Sector_Value(uint MFT_VALUE, Dictionary<long,string>
physical_list, Dictionary<string,string> logical_list, string input)
{
    string physical_drive_list = "";
    byte[] input_data = _Convert(Path.GetFileName(filepath.Text)); // test.txt ->
t0e0s0t0.0t0x0t0
    DriveInfo drives = new DriveInfo(input + Convert.ToChar(92)); // E:\
    foreach (var kvp in physical_list)
    {
        if (kvp.Key.Equals(drives.TotalSize + 4096))
        {
            SafeFileHandle handle1 = CreateFile(kvp.Value, GENERIC_READ |
GENERIC_WRITE, FILE_SHARE_READ | FILE_SHARE_WRITE, IntPtr.Zero, 3,
FILE_FLAG_NO_BUFFERING, IntPtr.Zero);
            if (!physical_drive_list.Equals(kvp.Value))
            {
                FileStream disk = new FileStream(handle1, FileAccess.Read);
                try
                {

```

```

        foreach (var kvp_log in logical_list)
        {
            if (kvp_log.Value.Equals(input))
            {
                string logical_drive_list = "";
                SafeFileHandle log_hand = CreateFile(kvp_log.Key,
                GENERIC_READ | GENERIC_WRITE, FILE_SHARE_READ | FILE_SHARE_WRITE, IntPtr.Zero, 3,
                FILE_FLAG_NO_BUFFERING, IntPtr.Zero);
                if (!logical_drive_list.Equals(kvp_log.Key))
                {
                    logical_drive_list = kvp_log.Key;
                    FileStream disk_log = new FileStream(log_hand,
                FileAccess.ReadWrite);

                    try
                    {
                        disk_log.Read(classes.vbr, 0, 512);
                        uint Sector_Per_Cluster_Value =
                classes.vbr[13];

                        int fix_num = 48;
                        for (int i = 0; i <
                classes.Vbr_class.lcnfm.Length; i++)
                        {
                            classes.Vbr_class.lcnfm[i] =
                classes.vbr[fix_num];

                            fix_num++;
                        }
                        MFT_VALUE =
                _fnBigToLittleEndian_UInt(classes.Vbr_class.lcnfm, Sector_Per_Cluster_Value);

                        int Overwrite_MFT = 0;
                        try
                        {
                            disk_log.Seek(MFT_VALUE,
                SeekOrigin.Begin);

                            disk_log.Read(classes.MFT, 0, 1024);

                            while (true)
                            {
                                Overwrite_MFT =
                _findFileNameByMFT(classes.MFT, input_data);

                                if (Overwrite_MFT == 0)
                                    break;

```



```

        else
        {
            disk_log.Seek(MFT_VALUE +
MFT_SECTOR_Size, SeekOrigin.Begin);

            MFT_VALUE +=
MFT_SECTOR_Size;

            disk_log.Read(classes.MFT, 0,
classes.MFT.Length);

        }
    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}
}
}
}
}
}
}
return MFT_VALUE;
}

public uint _fnBigToLittleEndian_UInt(byte[] data, uint Sector)
{
    if (BitConverter.IsLittleEndian)
        Array.Reverse(data);
    string hex = "";
    for (int i = 0; i < data.Length; i++)
    {
        if (data[i] < 16)

```

```

        hex += "0" + data[i];
    else
        hex += data[i];
    }
    uint result = uint.Parse(hex, System.Globalization.NumberStyles.HexNumber);
    return result * Sector * 512;
}

public byte[] _Convert(string data)
{
    string com_hex = string.Empty;
    byte[] result = Encoding.Default.GetBytes(data);
    byte[] hex_result = new byte[result.Length * 2];

    for (int i = 0; i < result.Length * 2; i++)
    {
        if (i % 2 == 1)
            hex_result[i] = 0;
        else
            hex_result[i] = result[i / 2];
    }
    return hex_result;
}

public int _findFileNameByMFT(byte[] MFT_DATA, byte[] FileName_Data)
{
    int c = MFT_DATA.Length - FileName_Data.Length + 1;
    int j;
    for (int i = 0; i < c; i++)
    {
        if (MFT_DATA[i] != FileName_Data[0])
            continue;
        for (j = FileName_Data.Length - 1; j >= 1 && MFT_DATA[i + j] ==
FileName_Data[j]; j--) ;
        if (j == 0)
            return 0;
    }
    return -1;
}

private void DeleteFolder(string foldername)
{
    var folderInfo = new DirectoryInfo(foldername);

```

```

        var parentDirectory = folderInfo.Parent.FullName;
        var name = 0;
        var prefix = "";
        var newFoldername = Path.Combine(parentDirectory, string.Format("{0}{1}",
prefix, name));
        while (Directory.Exists(newFoldername))
        {
            name++;
            if (name == int.MaxValue)
            {
                prefix += "x";
                name = 0;
            }
            newFoldername = Path.Combine(parentDirectory, string.Format("{0}{1}",
prefix, name));
        }
        Directory.Move(foldername, newFoldername);
        string[] dirs = Directory.GetDirectories(newFoldername);
        if(dirs.Length >0)
        {
            var newinfo = new DirectoryInfo(newFoldername);
            newinfo.Delete(true);
        }
        else
            Directory.Delete(newFoldername);
    }
    private void history_box_TextChanged(object sender, EventArgs e)
    {

    }
}
}

```

(PC) Classes.cs

```

using System;
using System.Windows.Forms;
using System.Reflection;
using System.Diagnostics;
using System.Runtime.InteropServices;
using System.IO;
using System.Collections.Generic;

```

```

using System.Text;

namespace jb_cap
{
    class classes
    {
        public static Byte[] mbr = new byte[512];
        public static Byte[] vbr = new byte[512];
        public static Byte[] MFT = new byte[1024];
        public static Byte[] Overwrite_0 = new byte[1024];
        public struct Vbr_class
        {
            public static Byte[] bps = new byte[2]; // Bytes Per Sector
            public static Byte[] lcnfm = new byte[8]; // Start Cluster for $MFT
            public static Byte[] lcnfmr = new byte[8]; // Start Cluster for $MFTMirror
            public static Byte[] signature = new byte[2];
        }
    }
}

```

(PC) disk1.cs

```

using System;
using System.Collections.Generic;
using System.Management;

namespace jb_cap
{
    class disk1
    {
        public Dictionary<long, string> disk11()
        {
            Dictionary<long, string> physical_list = new Dictionary<long, string>();
            var driveQuery = new ManagementObjectSearcher("select * from Win32_DiskDrive");
            foreach (ManagementObject d in driveQuery.Get())
            {

```

```

        var deviceId = d.Properties["Deviceld"].Value;
        var partitionQueryText = string.Format("associators of {{{0}}} where
AssocClass = Win32_DiskDriveToDiskPartition", d.Path.RelativePath);
        var partitionQuery = new ManagementObjectSearcher(partitionQueryText);
        foreach (ManagementObject p in partitionQuery.Get())
        {
            var logicalDriveQueryText = string.Format("associators of {{{0}}} where
AssocClass = Win32_LogicalDiskToPartition", p.Path.RelativePath);
            var logicalDriveQuery = new
ManagementObjectSearcher(logicalDriveQueryText);
            foreach (ManagementObject ld in logicalDriveQuery.Get())
            {
                var physicalName = Convert.ToString(d.Properties["Name"].Value);
                var diskName = Convert.ToString(d.Properties["Caption"].Value);
                var diskModel = Convert.ToString(d.Properties["Model"].Value);
                var diskInterface =
Convert.ToString(d.Properties["InterfaceType"].Value);
                var capabilities = (UInt16[])d.Properties["Capabilities"].Value;
                var mediaLoaded =
Convert.ToBoolean(d.Properties["MediaLoaded"].Value);
                var mediaType = Convert.ToString(d.Properties["MediaType"].Value);
                var mediaSignature =
Convert.ToUInt32(d.Properties["Signature"].Value);
                var mediaStatus = Convert.ToString(d.Properties["Status"].Value);
                var driveName = "WWW.WW" +
Convert.ToString(ld.Properties["Name"].Value);
                var driveId = Convert.ToString(ld.Properties["Deviceld"].Value);
                var driveCompressed =
Convert.ToBoolean(ld.Properties["Compressed"].Value);
                var driveType = Convert.ToUInt32(ld.Properties["DriveType"].Value);
                var fileSystem = Convert.ToString(ld.Properties["FileSystem"].Value);
                var freeSpace = Convert.ToUInt64(ld.Properties["FreeSpace"].Value);
                var totalSpace = Convert.ToInt64(ld.Properties["Size"].Value) +
4096;
                var driveMediaType =
Convert.ToUInt32(ld.Properties["MediaType"].Value);
                var volumeName =
Convert.ToString(ld.Properties["VolumeName"].Value);
                var volumeSerial =
Convert.ToString(ld.Properties["VolumeSerialNumber"].Value);
                physical_list.Add(totalSpace, physicalName);
            }
        }

```

```

    }
    }
    return physical_list;
}
public Dictionary<string, string> logical_drive()
{
    Dictionary<string, string> logical_list = new Dictionary<string, string>();
    var driveQuery = new ManagementObjectSearcher("select * from Win32_DiskDrive");
    foreach (ManagementObject d in driveQuery.Get())
    {
        var deviceId = d.Properties["DeviceId"].Value;
        var partitionQueryText = string.Format("associators of {{{0}}} where AssocClass = Win32_DiskDriveToDiskPartition", d.Path.RelativePath);
        var partitionQuery = new ManagementObjectSearcher(partitionQueryText);
        foreach (ManagementObject p in partitionQuery.Get())
        {
            var logicalDriveQueryText = string.Format("associators of {{{0}}} where AssocClass = Win32_LogicalDiskToPartition", p.Path.RelativePath);
            var logicalDriveQuery = new ManagementObjectSearcher(logicalDriveQueryText);
            foreach (ManagementObject ld in logicalDriveQuery.Get())
            {
                var driveName = "WWW.WW" + Convert.ToString(ld.Properties["Name"].Value);
                var driveId = Convert.ToString(ld.Properties["DeviceId"].Value);
                var driveCompressed = Convert.ToBoolean(ld.Properties["Compressed"].Value);
                var driveType = Convert.ToUInt32(ld.Properties["DriveType"].Value);
                var fileSystem = Convert.ToString(ld.Properties["FileSystem"].Value);
                var freeSpace = Convert.ToUInt64(ld.Properties["FreeSpace"].Value);
                var totalSpace = Convert.ToInt64(ld.Properties["Size"].Value) + 4096;
                var driveMediaType = Convert.ToUInt32(ld.Properties["MediaType"].Value);
                var volumeName = Convert.ToString(ld.Properties["VolumeName"].Value);
                var volumeSerial = Convert.ToString(ld.Properties["VolumeSerialNumber"].Value);
                logical_list.Add(driveName, driveId);
            }
        }
    }
}

```

```

    }
    return logical_list;
}
}
}

```

(PC) Helper.cs

```

using System;
using System.IO;
namespace jb_cap
{
    class Helper
    {
        public static bool WipeFileContent(string filename)
        {
            try
            {
                Random ran = new Random();
                var fileInfo = new FileInfo(filename);
                fileInfo.IsReadOnly = false;
                long filesize = fileInfo.Length;
                byte[] len_data = new byte[Convert.ToInt32(filesize)];
                ran.NextBytes(len_data);
                foreach(var item in len_data)
                for (int i = 0; i < 4; i++)
                {
                    using (var stream = new System.IO.StreamWriter(filename, false))
                    {
                        if (i == 2)
                            stream.BaseStream.Write(len_data,0,Convert.ToInt32(filesize));
                        else if (i == 3)
                            stream.Write("");
                        else
                        {
                            for (int j = 0; j < Convert.ToInt32(filesize); j++)
                            {
                                stream.Write(i);
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

        return true;
    }
    catch (Exception)
    {
        return false;
    }
}

internal static int WipeDirectory(string directoryName, System.IO.SearchOption
wipeOption)
{
    var count = 0;
    foreach (var item in System.IO.Directory.GetFiles(directoryName, "*",
wipeOption))
    {
        Console.WriteLine(item);
        if (WipeFileContent(item))
        {
            count++;
        }
    }
    return count;
}
}
}

```

(PC) LoginForm.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace jb_cap
{
    public delegate void EventHandler(string userName);
}

```



```

public partial class LoginForm : Form
{
    public event EventHandler loginEventHandler;
    public LoginForm()
    {
        InitializeComponent();
    }
    private void login_btn_Click(object sender, EventArgs e)
    {
        LoginHandler loginHandler = new LoginHandler();
        if (ControlCheck())
        {
            if (loginHandler.LoginCheck(id_box.Text, pwd_box.Text))
            {
                string userName = id_box.Text;
                loginEventHandler(userName);
                DialogResult = DialogResult.OK;
            }
            else
            {
                MessageBox.Show("로그인 정보가 정확하지 않습니다.", "에러",
                MessageBoxButtons.OK, MessageBoxIcon.Warning);
                id_box.Clear();
                pwd_box.Clear();
            }
        }
    }
    private bool ControlCheck()
    {
        if (String.IsNullOrEmpty(id_box.Text))
        {
            MessageBox.Show("ID가 입력되지 않았습니다.", "에러",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
            id_box.Focus();
            return false;
        }
        else if (String.IsNullOrEmpty(pwd_box.Text))
        {
            MessageBox.Show("PWD가 입력되지 않았습니다.", "에러",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
            pwd_box.Focus();
            return false;
        }
    }
}

```

```

    }
    return true;
}
private void signup_label_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    System.Diagnostics.Process.Start("http://eraser2020.herokuapp.com");
}

private void id_find_label_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    System.Diagnostics.Process.Start("http://eraser2020.herokuapp.com");
}

private void pwd_find_label_LinkClicked(object sender, LinkLabelLinkClickedEventArgs
e)
{
    System.Diagnostics.Process.Start("http://eraser2020.herokuapp.com");
}
}
}

```

(PC) LoginHandler.cs

```

using RestSharp;
using Newtonsoft.Json;
using Newtonsoft.Json.Linq;

namespace jb_cap
{
    class LoginHandler
    {
        public class JSdata
        {
            public string success;
            public string token;
        }
        public bool LoginCheck(string id, string password)
        {
            var client = new
RestClient("http://eraser2020.herokuapp.com/users/authenticate");
            client.Timeout = -1;
            var request = new RestRequest(Method.POST);
            request.AddHeader("Content-Type", "application/json");

```

```

        var json_data = new JObject();
        json_data.Add("userid",id);
        json_data.Add("userpassword", password);

        string str_json = JsonConvert.SerializeObject(json_data);
        request.AddParameter("application/json", str_json, ParameterType.RequestBody);
        IRestResponse response = client.Execute(request);
        string json_text = response.Content;
        JSdata jsdata = JsonConvert.DeserializeObject<JSdata>(json_text);
        if (jsdata.success.Equals("true"))
        {
            Form1.user_id = id;
            return true;
        }
        else
            return false;
    }
}

```

(PC) Privilege.cs

```

using System.Security.Principal;

namespace jb_cap
{
    class Privilege
    {
        public bool IsRunningAsAdministrator()
        {
            WindowsIdentity windowsIdentity = WindowsIdentity.GetCurrent();
            WindowsPrincipal windowsPrincipal = new WindowsPrincipal(windowsIdentity);
            return windowsPrincipal.IsInRole(WindowsBuiltInRole.Administrator);
        }
    }
}

```

(PC) Program.cs

```

using System;
using System.Windows.Forms;

namespace jb_cap

```

```

{
    static class Program
    {
        [STAThread]
        static void Main(String[] args)
        {
            if(args != null && args.Length > 0)
            {
                Wipe(args);
            }
            else
            {
                Application.EnableVisualStyles();
                Application.SetCompatibleTextRenderingDefault(false);
                Application.Run(new Form1());
            }
        }
        private static void Wipe(String[] args)
        {
            foreach (var item in args)
            {
                try
                {
                    //MessageBox.Show(string.Format("Press OK to start wiping {0}.", item));
                    if (System.IO.File.Exists(item))
                    {
                        if (MessageBox.Show(string.Format("Press Yes to wipe {0}.", item),
"Confirm", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
                        {
                            Helper.WipeFileContent(item);
                        }
                    }
                    else if (System.IO.Directory.Exists(item))
                    {
                        var files = System.IO.Directory.GetFiles(item, "*",
System.IO.SearchOption.AllDirectories);
                        if (files.Length == 0)
                        {
                            MessageBox.Show(string.Format(
                                "No file exists under {0}", item), "Tip",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                        }
                    }
                }
            }
        }
    }
}

```

```

        else if (MessageBox.Show(string.Format(
            "Are you sure to wipe {0} files under {1}?", files.Length, item),
            "Confirm", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
        {
            foreach (var file in files)
            {
                Helper.WipeFileContent(file);
            }
        }
    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.ToString(), "error", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
}
}
}
}
}

```

(PC) transfer.cs

```

using RestSharp;
using System;
using System.IO;

namespace jb_cap
{
    class transfer
    {
        public static void Main(string filepath, string filename, string username)
        {
            var client = new RestClient("http://eraser2020.herokuapp.com/logs/logupload");
            client.Timeout = -1;
            var request = new RestRequest(Method.POST);
            request.AddFile("file", filepath);
            request.AddParameter("userid", username);
            request.AddParameter("filename", filename);
            IRestResponse response = client.Execute(request);
            Console.WriteLine(response.Content);
        }
    }
}

```

```
}  
}
```

(Web) routes/certs.js

```
const express = require('express');  
const router = express.Router();  
const User = require('../models/user');  
const nodemailer = require('nodemailer');  
const randomString = Math.random().toString(36).slice(2);  
const LocalStorage = require('node-localstorage').LocalStorage,  
localStorage = new LocalStorage('./scratch');  
  
require('dotenv').config();  
  
// email  
router.post('/emailauth', (req, res, next) => {  
  let Email = new User({  
    userid: localStorage.getItem('userid'),  
    semail: req.body.semail,  
    ecert: randomString,  
  });  
  
  let cert = { cert: req.body.cert };  
  
  if (!cert.cert) {  
    let smtpTransport = nodemailer.createTransport({  
      service: 'Gmail',  
      auth: {  
        user: process.env.gid,  
        pass: process.env.gpw  
      }  
    });  
  
    const mailOptions = {  
      from: `"Eraser.T" <${process.env.gid}>`,  
      to: Email.semail,  
      subject: '인증번호 요청이 발송되었습니다.',  
      text: randomString  
    };  
  
    // html: '<h1>HTML Test<h1><p></p>'
```

```

    };

    smtpTransport.sendMail(mailOptions, function (err, info) {
        if (err) {
            console.log(err);
        }

        else {
            console.log('email has been sent. ');
            User.updateOne({userid:Email.userid}, { $set: { "semail": req.body.semail,
"ecert": randomString } }, function (err) {
                if (err) {
                    res.json({ success: false, msg: '변경 실패' });
                } else {
                    res.json({ success: true, msg: '변경 성공' });
                }
            });
        }
        smtpTransport.close();
    });
}

else {
    User.findOne({userid:Email.userid}, function (err, user) {
        if (err) console.log(err);
        if (cert.cert == user.ecert) {
            User.updateOne({userid:Email.userid}, { $set: { "auth": true } }, function (err)
{
                if (err) console.log(err)
                localStorage.clear();
            });
            res.json({ success: true, msg: '인증 성공' });
        } else {
            res.json({ success: false, msg: '일치하지 않음' });
        }
    });
}
});

module.exports = router;

```

(Web) routes/logs.js

```

const express = require('express');
const router = express.Router();
const gridfs = require('gridfs-stream');
const fs = require('fs');
const mongoose = require('mongoose');
const crypto = require('crypto');
const config = require('../config/database');
const schedule = require('node-schedule');
const moment = require('moment');

// 암호화 알고리즘
const algorithm = 'aes-256-ctr';

// 암호화 키
let passkey = config.secret;
passkey = crypto.createHash('sha256').update(String(passkey)).digest('base64').substr(0, 32);

// 암호화
const encrypt = (buffer) => {
  const iv = crypto.randomBytes(16);
  const cipher = crypto.createCipheriv(algorithm, passkey, iv);
  const result = Buffer.concat([iv, cipher.update(buffer), cipher.final()]);
  return result;
}

// 복호화
const decrypt = (encrypted) => {
  const iv = encrypted.slice(0, 16);
  encrypted = encrypted.slice(16);
  const decipher = crypto.createDecipheriv(algorithm, passkey, iv);
  const result = Buffer.concat([decipher.update(encrypted), decipher.final()]);
  return result;
};

gridfs.mongo = mongoose.mongo;
var connection = mongoose.connection;

connection.once('open', () => {

  var gfs = gridfs(connection.db);

  // 30일 이후 자동 삭제

```



```

var schdu = schedule.scheduleJob('* * 0 * * *', function () {
  let amonth = moment().subtract(1, 'month')
  var query = {
    "uploadDate": {
      "$lte": new Date(amonth)
    }
  };
  gfs.files.find(query).sort({ uploadDate: -1 }).toArray(function (err, logs) {
    if (err) {
      console.log(err);
      return;
    } else if (logs.length == 0) {
      console.log("삭제할 파일이 없습니다.")
    } else {
      for (var i = 0; i < logs.length; i++) {
        var filename = logs[i].filename
        gfs.remove({ _id: logs[i]._id }, (err) => {
          if (err) {
            console.log(err);
          } else {
            console.log(filename + " 파일 삭제 완료");
          }
        });
      }
    }
  });
});

//      // 업로드 잘됨시작
//      // 로그 업로드
router.post('/logupload', (req, res) => {
  let {
    file
  } = req.files;

  var filename = req.body.filename;
  console.log(req.files);

  var userid = req.body.userid;
  let writeStream = gfs.createWriteStream({
    filename: filename,
    mode: 'w',

```

```

        content_type: file.mimetype,
        aliases: userid,
        chunkSize: 10485760
    });

    writeStream.on('close', function (uploadedFile) {
        res.send('Stored File: ' + uploadedFile.filename);
    });
    writeStream.write(encrypt(file.data));
    writeStream.end();

});

//      // 업로드 잘됨 끝
// 모바일용 업로드
router.post('/loguploads', (req, res) => {

    var filename = req.body.filename;
    var mimetype = req.body.mimetype;
    var data = req.body.data;
    var userid = req.body.userid;

    let writeStream = gfs.createWriteStream({

        filename: filename,
        mode: 'w',
        content_type: mimetype,
        aliases: userid,
        chunkSize: 10485760
    });

    writeStream.on('close', function (uploadedFile) {
        res.send('Stored File: ' + uploadedFile.filename);
    });
    writeStream.write(encrypt(data));
    writeStream.end();

});

// 로그 다운로드
router.get('/logdownload/:_id', (req, res) => {

```

```

gfs.findOne({ _id: req.params._id }, (err, file) => {
  if (!file) {
    return res.status(404).send({
      message: 'File was not found'
    });
  }
  res.set('Content-Type', file.contentType);
  var newFilename = encodeURIComponent(file.filename);
  res.set('Content-Disposition', 'attachment; filename*=UTF-8W'W' + newFilename);
  let data = [];
  let readstream = gfs.createReadStream({
    _id: req.params._id, highWaterMark: 10485760
  });
  readstream.on('data', (chunk) => {
    data.push(chunk);
  });
  readstream.on('end', () => {
    data = Buffer.concat(data);
    res.end(decrypt(data));
  });
  readstream.on('error', (err) => {
    console.log('[*] Error, while downloading a file, with error: ${err}');
    res.status(400).send({
      message: `Error, while downloading a file, with error: ${err}`
    });
  });
});

// 로그 불러오기
router.get('/logread/:userid', (req, res) => {
  var userid = req.params.userid
  gfs.files.find({ aliases: userid }).sort({ uploadDate: -1 }).toArray(function (err, logs) {
    if (err) console.log(err);
    res.status(200).json({ msg: logs });
  });
});

// 로그 검색
router.get('/search/:userid/:filename', (req, res, next) => {
  gfs.files.find({
    filename: { $regex: new RegExp(req.params.filename, 'i') },

```

```

        aliases: req.params.userid
    }).sort({ uploadDate: -1 }).toArray(function (err, logs) {
        if (err)
            res.status(500).json({ errmsg: err });
        res.status(200).json({ msg: logs });
    });
});

// 로그 삭제
router.get('/logdelete/:userid/:_id', (req, res) => {
    var userid = req.params.userid;
    var id = req.params._id
    gfs.exist({_id : id}, (err, file) => {
        if (err || !file) {
            res.status(404).send('File Not Found');
            return;
        }
        gfs.remove({_id : id}, (err) => {
            if (err) res.status(500).send(err);
            res.redirect('http://eraser2020.herokuapp.com/logs/' + userid);
        });
    });
});

module.exports = router;

```

(Web) routes/users.js

```

const express = require('express');
const router = express.Router();
const passport = require('passport');
const jwt = require('jsonwebtoken');
const User = require('../models/user');
const config = require('../config/database');
const bcrypt = require('bcryptjs');
const LocalStorage = require('node-localstorage').LocalStorage,
localStorage = new LocalStorage('./scratch');

// register
router.post('/register', (req, res, next) => {

```

```

let newUser = new User({
  username: req.body.username,
  userid: req.body.userid,
  userpassword: req.body.userpassword,
  userphone: req.body.userphone,
  semail: 0,
  ecert: 0,
  auth: false
});
let newPass = new User({
  userpassword2: req.body.userpassword2
});
if (!newUser.username ||
    !newUser.userid ||
    !newUser.userpassword ||
    !newPass.userpassword2 ||
    !newUser.userphone) {
  return res.json({ success: false, msg: '빈칸 있음, 양식 틀림' });
} else if (newUser.userpassword !== newPass.userpassword2) {
  return res.json({ success: false, msg: '비밀번호 다름' });
} else {
  User.getUserByUsername(newUser.userid, (err, user) => {
    if (err) throw err;
    if (user) {
      return res.json({ success: false, msg: "이미 아이디 있음" });
    } else {
      User.addUser(newUser, (err, user) => {
        if (err) {
          res.json({ success: false, msg: '등록 실패' });
        } else {
          res.json({ success: true, msg: '가입 성공' });
        }
      });
    }
  });
}
});

//authenticate
router.post('/authenticate', (req, res, next) => {

```

```

const userid = req.body.userid;
const userpassword = req.body.userpassword;
User.getUserByUsername(userid, (err, user) => {
  if (err) throw err;
  if (!user) {
    return res.json({ success: false, msg: '아이디 없음' });
  }
  User.comparePassword(userpassword, user.userpassword, (err, isMatch) => {
    if (err) throw err;
    if (isMatch) {
      User.findOne({ userid }, (err, user) => {
        if (user.auth == false) {
          localStorage.setItem('userid',userid);
          return res.json({ success: false, auth: false, msg: '이메일 인증 안함'
});
        } else if (user.auth == true) {
          const token = jwt.sign({ data: user }, config.secret, {
            expiresIn: 604800 // 1week
          });

          res.json({
            userNoPW2 : user.username,
            success: true,
            token: 'JWT ' + token,
            userNoPW: {
              //id: user._id,
              username: user.username,
              userid: user.userid,
              semail: user.semail
            }
          });
        }
      });
    } else {
      return res.json({ success: false, msg: '패스워드 다름' });
    }
  });
});
});

//profile
// 인증 오류시 사용

```

```

router.get('/profile', (req, res, next) => {
  res.json({
    user: {
      username: req.user.username,
      userid: req.user.userid,
      semail: req.user.semail
    }
  });
})

//password change
router.put('/profile/updatePassword', function (req, res) {
  var saltRounds = 10;
  bcrypt.genSalt(saltRounds, function (err, salt) {
    bcrypt.hash(req.body.userpassword, salt, function (err, hash) {
      req.body.userpassword = hash;
      User.updateOne({ userid: req.body.userid }, { $set: { "userpassword":
req.body.userpassword } }, function (err) {
        if (err) {
          res.json({ success: false, msg: '변경 실패' });
        } else {
          res.json({ success: true, msg: '변경 성공' });
        }
      });
    });
  });
});

// email change = phone change
router.put('/profile/updateEmail', function (req, res) {
  User.updateOne({ userid: req.body.userid }, { $set: { "userphone": req.body.userphone } },
function (err) {
  if (err) {
    res.json({ success: false, msg: '변경 실패' });
  } else {
    res.json({ success: true, msg: '변경 성공' });
  }
});
});

module.exports = router;

```

(Web) models/user.js

```
const mongoose = require('mongoose');
const bcrypt = require('bcryptjs');
const config = require('../config/database');

// user schema
const UserSchema = mongoose.Schema({
  username: {
    type: String,
    required: true
  },
  userid: {
    type: String,
    required: true
  },
  userpassword: {
    type: String,
    required: true
  },
  userpassword2: {
    type: String,
    required: false
  },
  userphone: {
    type: String,
    required: true
  },
  semail: {
    type: String,
    required: false
  },
  ecert: {
    type: String,
    required: false
  },
  auth: {
    type: Boolean,
    required: true
  },
});
const User1 = module.exports = mongoose.model('User', UserSchema);
```



```

// id 불러오기
User1.getUserById = function (id, callback) {
  User1.findById(id, callback);
}

// username 불러오기
User1.getUserByUsername = function (userid, callback) {
  const query = { userid: userid };
  User1.findOne(query, callback);
}

// 새로운 user 만들기, bcrypt로 password 암호화
User1.addUser = function (newUser, callback) {
  bcrypt.genSalt(10, (err, salt) => {
    bcrypt.hash(newUser.userpassword, salt, (err, hash) => {
      if (err) throw err;
      newUser.userpassword = hash;
      newUser.save(callback);
    });
  });
}

// 로그인시 비밀번호 일치하는지 확인
User1.comparePassword = function (candidatePassword, hash, callback) {
  bcrypt.compare(candidatePassword, hash, (err, isMatch) => {
    if (err) throw err;
    callback(null, isMatch);
  });
}

```

(Web) config/database.js

```

module.exports = {
  // database: 'mongodb://localhost:27017/eraser',
  // secret: "winnerwinnerchickendinner"
  database: "mongodb+srv://test-user:test-user@cluster0-k0jft.mongodb.net/test?retryWrites=true&w=majority",
  secret: "yoursupersecret-dfljksdlfkj"
}

```

(Web) config/passport.js

```

const JwtStrategy = require('passport-jwt').Strategy;
const ExtractJwt = require('passport-jwt').ExtractJwt;
const User = require('../models/user');
const config = require('../config/database');

// jwt토큰인증
module.exports = function(passport) {
  let opts = {};
  opts.jwtFromRequest = ExtractJwt.fromAuthHeaderWithScheme('jwt');
  opts.secretOrKey = config.secret;
  passport.use(new JwtStrategy(opts, (jwt_payload, done) => {
    User.getUserById(jwt_payload.data._id, (err, user) => {
      if(err)
        return done(err, false);
      if(user)
        return done(null, user);
      else
        return done(null, false);
    });
  }));
}

```

(Web) app.js

```

const express = require('express');
const path = require('path');
const bodyParser = require('body-parser');
const cors = require('cors');
const passport = require('passport');
const mongoose = require('mongoose');
const users = require('./routes/users');
const certs = require('./routes/certs');
const config = require('../config/database');
const app = express();
const port = process.env.PORT || 3000;
require('./express.config')(app);

// 연결 옵션
mongoose.connect(config.database, { useNewUrlParser: true, useUnifiedTopology: true })
  .catch(function (error) { console.log('catch handler') });

// 연결 성공 메시지

```

```

mongoose.connection.on('connected', () => {
  console.log('Connected to Database ' + config.database);
});

// 연결 에러 메시지
mongoose.connection.on('error', err => {
  console.log('Database error: ' + err);
});

/* Cross Origin Resource Sharing의 약자로, 현재 Application의 도메인(웹페이지)에서
다른 웹페이지 도메인으로 리소스가 요청되는 경우를 얘기합니다.
예를 들면, 웹페이지인 http://web.com 에서 API서버 URL인 http://api.com이란
도메인으로 API를 요청하면 http 형태로 요청이 되므로 브라우저 자체에서 보안 상 이유로
CORS를 제한하게 되는 현상을 말합니다. */
app.use(cors());

/* 가지고 있는 데이터를 내가 원하는 형태의 데이터로 '가공'하는 과정을 parsing 이라 하며
그 과정을 수행하는 모듈 혹은 메소드를 parser 라 일컫는다.
단순히 말하자면 내가 모르는 언어를 내가 원하는 언어의 구조로 바꿔주는 일종의 구문 해석
기라고 말할 수도 있다. */
app.use(bodyParser.json());

// passport middleware
app.use(passport.initialize());
app.use(passport.session());
require('./config/passport')(passport);

// 기본 설정
app.use(express.static(path.join(__dirname, 'public')));
app.use('/certs', certs);
app.use('/users', users);
app.use('/ekdnsfhem', express.static(__dirname + '/download'));

// 서버 시작 메시지
app.listen(port, function () {
  console.log("server started on port " + port);
});

app.get('*', (req, res) => {
  res.sendFile(path.join(__dirname, 'public/index.html'));
});

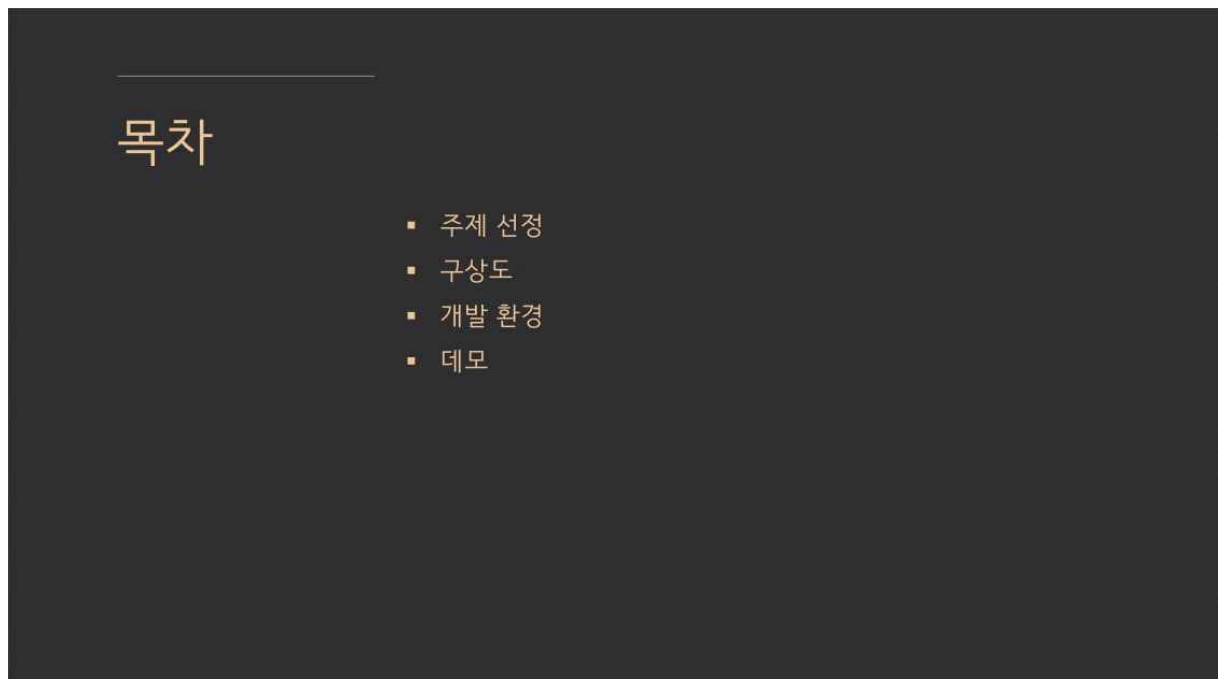
```

(Web) express.config.js

```
const logger = require('morgan');
const busboyBodyParser = require('busboy-body-parser');
module.exports = app => {
  app.use(logger('dev'));
  app.use((req, res, next) => {
    res.header('Access-Control-Allow-Origin', '*');
    res.header('Access-Control-Allow-Methods', 'GET,PUT,POST,DELETE');
    res.header('Access-Control-Allow-Headers', 'Content-Type, Authorization');
    next();
  });
  app.use(busboyBodyParser({ limit: '50mb' }));
  const logs = require('./routes/logs');
  app.use('/logs', logs);
};
```

5.2 발표자료

(최종 PPT 사진 파일 첨부)



주제 선정

QR/WiFi Attendance management system by Baby Forces

QR/WiFi 출석관리 시스템

QR/WiFi Attendance management system by Baby Forces



Bluetooth

기존 방식

낮은 인식률과 잦은 오류
저조한 속도와 불편한 유저경험



QR/WiFi

고안한 방식

정확한 인식률과 신뢰성
처리속도의 증대와 더 나은 유저경험

QR/WiFi 출석관리 시스템

QR/WiFi Attendance management system by Baby Forces



직접 태깅하는 행위

해당장소에 존재 해야 하며,
행위로 하여금 당위성 부여,
쉬운 사용자 경험

원격지 조작

제한된 데이터

타인에 의한 인증

QR/WiFi 출석관리 시스템

QR/WiFi Attendance management system by Baby Forces

누가?



타인에 의한 인증

어디서?



원격지 조작

어떻게?



제한된 데이터

QR/WiFi 출석관리 시스템

QR/WiFi Attendance management system by Baby Forces

Scheduling

출결 관리의 자동화
쉬운 관리로 줄어든 업무



Broad user

쉬운 UI와 자동화된 절차
편리한 유저 경험 제공

Strong Verification

생체인증과 전자서명 방식의
로그인과 신분확인 절차



Stability

신뢰성 있는 출석체크
GPS / WIFI-MAC 지속적인 확인으로
무단 공결 방지

구상도

QR/WiFi Attendance management system by Baby Forces

QR/WiFi 출석관리 시스템

QR/WiFi Attendance management system by Baby Forces



학생/교수

사용자 어플리케이션
학생 - 스마트폰 어플리케이션
교수 - 스마트폰 어플리케이션 / 웹



QR 단말기

QR코드 인증 단말기
웹으로 제공, 웹이 가능한 단말기



서버

출석 인증
사용자 인증과 출석 검증

QR/WiFi 출석관리 시스템

QR/WiFi Attendance management system by Baby Forces



1. 등록 단계

유저는 개인 스마트폰을 이용하여
생체 인증을 통해 자신의 기기와 계정정보를 등록



2. 출석 체크 단계

유저는 등록된 기기만을 사용하여
QR코드를 생성/촬영을 통해 출석 체크

QR/WiFi 출석관리 시스템

QR/WiFi Attendance management system by Baby Forces



3-1. 출석 현황 확인

자신의 출석 정보 확인 가능



3-2. 학생 관리

교수 클라이언트는 학생관리 기능 제공
학생의 강제 출석/결석,
학생의 기기변경여부 등을 확인가능

등록단계

QR/WiFi Attendance management system by Baby Forces

사용자 어플리케이션



PriK - 안드로이드 루트권한에
의하여 AES로 암호화 되어 보관



RSA 키 생성

Pubk - 서버에게 전송

PriK - 안드로이드 컨테이너 보관

기기정보 발송

기기의 고유정보/USIM 정보 발송



최초등록 검증후 Response

최초 기기가 맞다면 저장

고유 사용자 식별정보 생성과 전송

예정) 새로운 기기 등록시 교수측에서 확인 가능

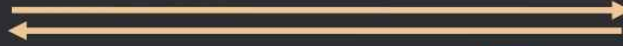
등록단계

QR/WIFI Attendance management system by Baby Forces

사용자 어플리케이션



로그인 요청
고유 사용자 식별 정보 전송



임시 메시지 발급
256byte 길이의 임시 메시지 발급



등록단계

QR/WIFI Attendance management system by Baby Forces



개인키로 암호화 한 임시메시지 전송



JWT

JSON Web Token

토큰을 통한 세션 관리

Issuer : 발급기관

ExpireTime : 만료시간 (5분)

Name : 유저 식별 정보

사전에 받은 공개키로 복호화 후 검



Passive 방식

QR/VNFI Attendance management system by Baby Forces



Active 방식

QR/VNFI Attendance management system by Baby Forces



JWT 구조

QR/WiFi Attendance management system by Baby Forces

클라이언트에서 발급후 QR코드로 생성

토큰의 비밀번호는 로그인시 사용한 임시메시지를 사용

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJzZGdzMWdkNXNkZjFzZGY1MTUiLCJpYXQiOiE1MTYyMzkwMjlsImV4cCI6MTUxNjI0OTAyMn0.aABm3Wt0Tmnsgr9bewXnKMIqhutanfhwE7eB0olhwc
```

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

알고리즘 타입

```
{
  "sub": "sdgs1gd5sdf1sdf515",
  "exp": 1516249022
}
```

유저 식별 정보
토큰 만료 시간

$\text{HMACSHA256}(\text{base64UrlEncode}(\text{header}) + "." + \text{base64UrlEncode}(\text{payload}), \text{SECRET})$

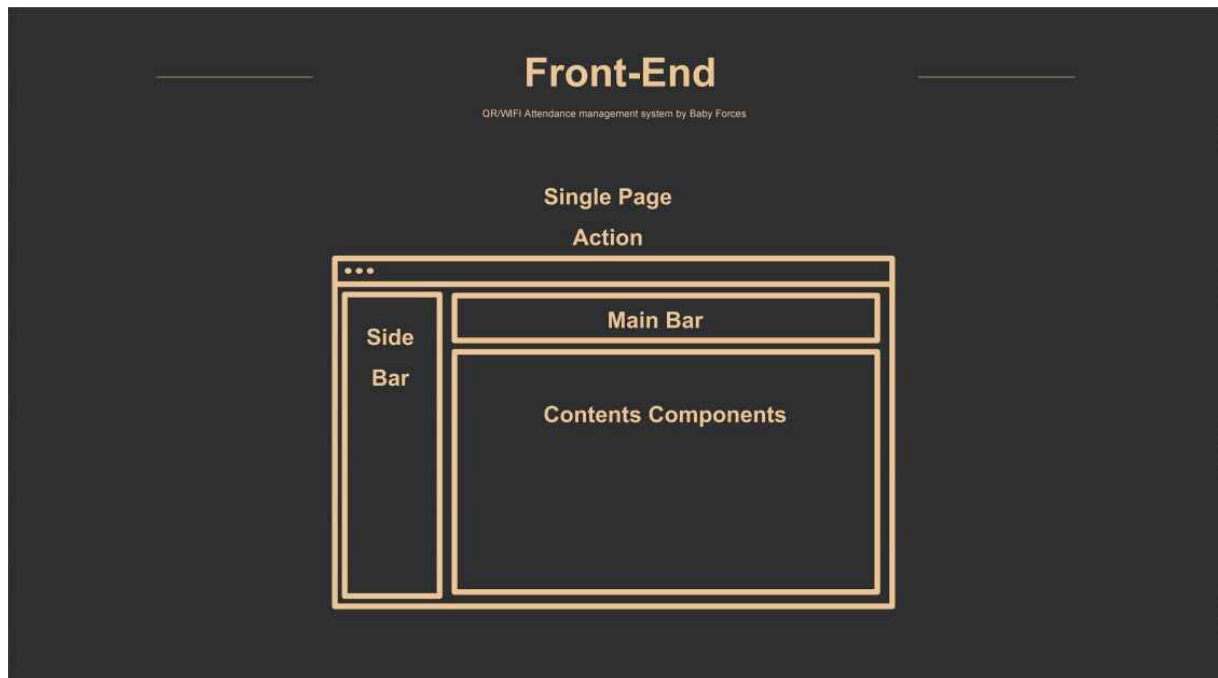
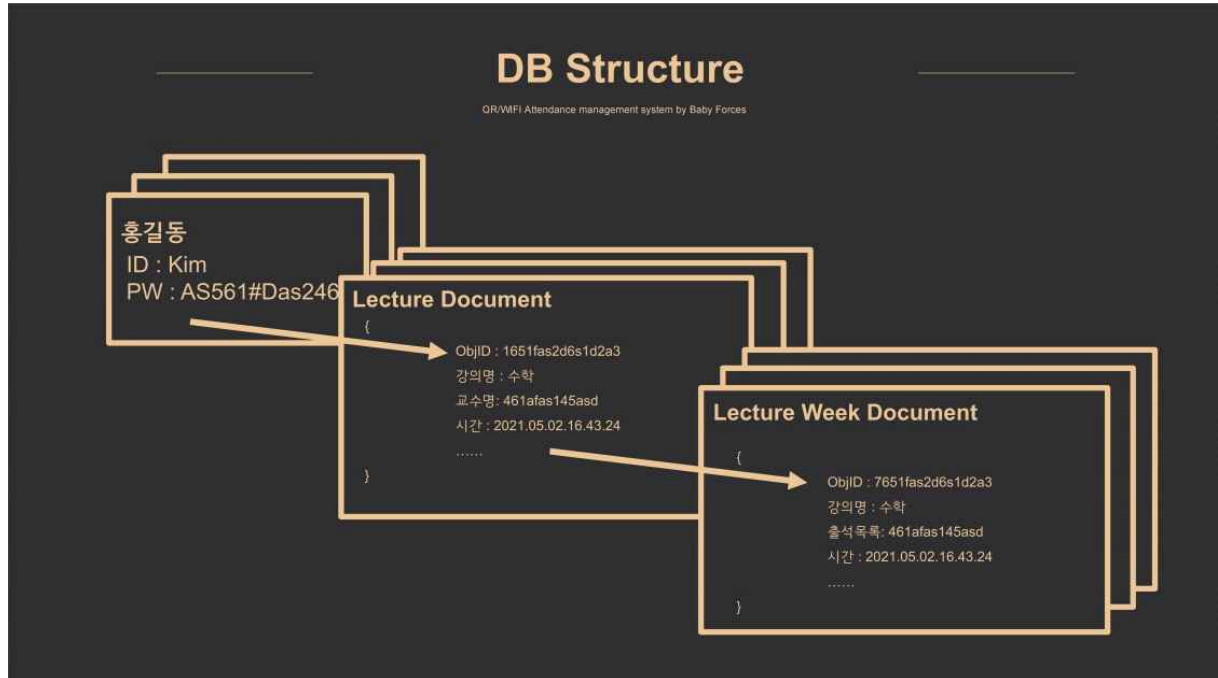
사전에 공유한 임시 메시지를 비밀번호로 사용

JWT 구조

QR/WiFi Attendance management system by Baby Forces

나는 그저 교수 계정으로 로그인한 단말기야
교수 토큰을 헤더에 담아 인식한 유저토큰을 보내기만 하면되





Front-End

QR/WIFI Attendance management system by Baby Forces

Handmade Android JAVA



개발 환경

QR/WIFI Attendance management system by Baby Forces



데모

QR/WIFI Attendance management system by Baby Forces

데모

QR/WIFI Attendance management system by Baby Forces

학생 측 어플리케이션

