

I-Brand 열정*학기제 주차별 활동 보고서

주차	11주차	제출 일자	2023. 11. 19
팀명	CS 아카이브	학과	소프트웨어학과
프로젝트명	웹사이트 및 안드로이드 앱을 이용한 CS-아카이브 제작	학번	2022564004
		성명	김다빈
활동 내용	<p>주요 활동 내용</p> <ul style="list-style-type: none"> • 앱 개발 관련 이론 지식 학습 • 안드로이드 스튜디오 설치 및 기본 세팅 • 안드로이드 앱 제작 실습 <hr style="border-top: 1px dashed black;"/> <p>앱 개발 관련 이론 지식 학습</p> <p><u>앱 개발 방식에 따른 앱 분류</u></p> <p>- 네이티브 앱(Native App)</p> <p style="padding-left: 20px;">: 특정 플랫폼 또는 운영 체제에 최적화된 앱으로, 특정 기기 혹은 운영 체제에 직접 설치되어 실행된다. 특정 플랫폼에 최적화되어 있기에 앱의 성능이 우수한 측에 속한다.</p> <p style="padding-left: 20px;">[사용 언어] iOS - Swift, Objective-C Android - Java, Kotlin</p> <p>- 크로스 플랫폼 앱(Cross-Platform App)</p> <p style="padding-left: 20px;">: 여러 플랫폼에서 동일한 코드베이스로 실행되는 앱으로, 하나의 코드베이스로 여러 플랫폼을 대상으로 하여 개발한다. 플랫폼에 최적화된 네이티브 앱에 비해 다소 성능이 떨어질 수 있다.</p> <p style="padding-left: 20px;">[사용 언어] React Native, Xamarin, Flutter, JavaScript, C#, Dart 등</p> <p>- 웹 앱(Web App)</p> <p style="padding-left: 20px;">: 웹 브라우저에서 실행되는 앱으로, 별도의 설치 과정 없이 웹을 통해 접근할 수 있다. Native나 Cross-Platform에 비해 성능이 낮을 수 있다.</p> <p style="padding-left: 20px;">[사용 언어] HTML, CSS, JavaScript</p>		

Android 와 IOS

- 안드로이드 앱 개발

: 개발에 Java 또는 Kotlin을 주로 사용

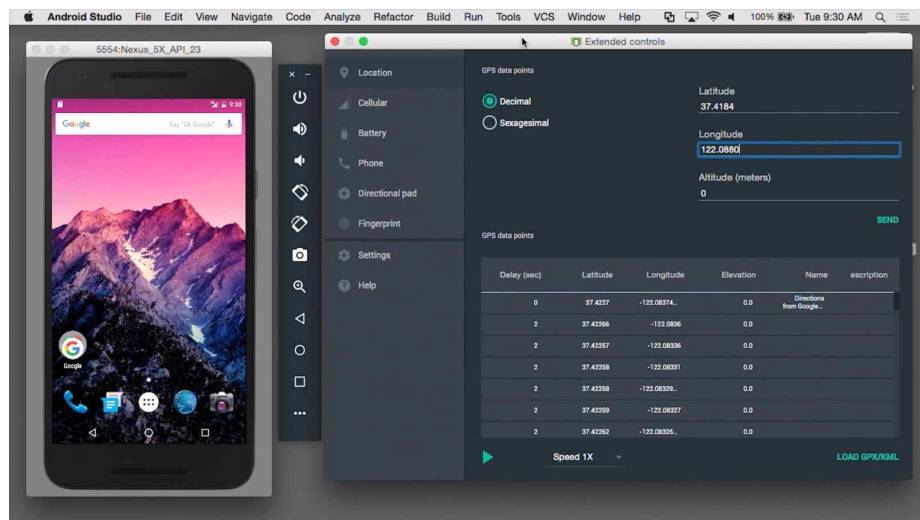
APK(Android Package) 형식으로 패키징되어 배포되며,
여러 안드로이드 기기에서 동작

- iOS 앱 개발

: Swift 언어로 주로 개발, Objective-C도 지원

IPA(iOS App Store Package) 형식으로 패키징되어 배포되며,
iOS 기기에서만 어플리케이션의 실행이 가

※ 개발에 사용할 툴(Tool)



Android Studio

: 안드로이드 애플리케이션을 개발하기 위한 공식 통합 개발 환경(IDE)

[기능]

- Gradle 빌드 시스템을 사용하여 프로젝트를 빌드하고 종속성 관리
- XML 기반의 레이아웃 에디터, 디바이스 에뮬레이터 등을 지원

안드로이드 스튜디오 설치 및 기본 세팅

Android Studio 설치

- 1) Android Studio 홈페이지에서 설치 파일 다운로드
- 2) Android Studio 설치
- 3) Android SDK 다운로드

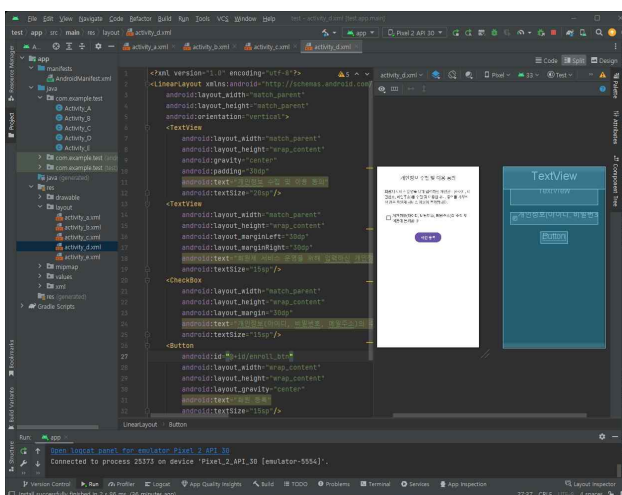
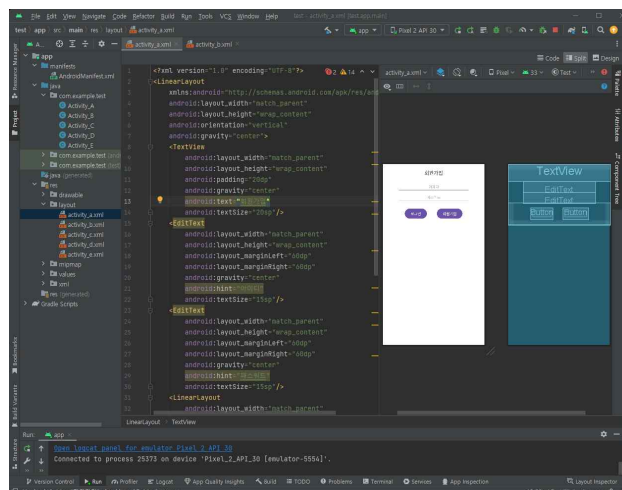
Virtual Device 생성

- 1) Create Device 선택
- 2) 원하는 사양으로 Device 설정
- 3) system image 다운로드 및 선택
- 4) 추가 설정

안드로이드 앱 제작 실습

※ Java 언어를 이용하여 실습

- xml을 이용한 사용자 인터페이스(사용자 화면) 구성



• intent를 이용한 Activity 이동

```

2. usages
public class Activity_A extends AppCompatActivity {

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

        Button signin_btn = findViewById(R.id.signin_btn);
        signin_btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(getApplicationContext(), Activity_B.class));
            }
        });
    }
}

```

```

public class Activity_B extends AppCompatActivity {

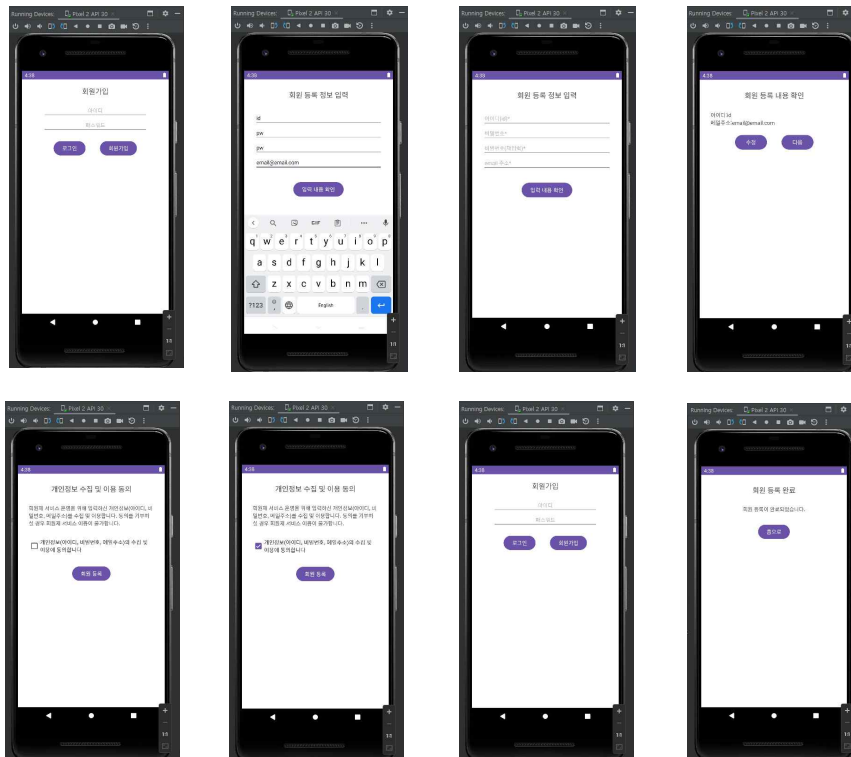
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_b);

        EditText id_et=findViewById(R.id.id_et);
        EditText email_et=findViewById(R.id.email_et);
        Button confirm_btn=findViewById(R.id.confirm_btn);

        confirm_btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new Intent(getApplicationContext(), Activity_C.class);
                String s="아이디:"+id_et.getText();
                s+="\n패밀주소:"+email_et.getText();
                intent.putExtra( name: "ORDER_INFO", s);
                startActivity(intent);
            }
        });
    }
}

```

• 전체 구현 화면



느낀 점

앱 개발에 대해서는 아직 아는 게 많지 않아서 작업할 때 버그라도 발생하면 어느 세월에 디버깅하고 남은 작업할지 걱정이 좀 있었는데, 선배와 함께 작업하다 보니 다행히 큰 문제 없이 1차적인 실습을 잘 마무리한 것 같아 다행이라는 생각이 들었다. 그리고 매번 사용자 입장에서만 앱을 대하다가 개발자의 입장이 되어 기능이나 UI를 구현할 때 어떤 식으로 동작하도록 만들 것인지 고민하고 개발 작업도 해 보니 앱 개발도 너무 어렵게만 생각하지 않아도 되겠다는 생각이 들었다. 이후 작업들도 큰 문제 없이 진행할 수 있었으면 좋겠다.