PORTFOLIO

Joomin Lee 2020

Skills

JAVA

- JAVA 문법과 객체지향 프로그래밍 학습
- 입출력 스트림(I/O Stream)을 이용한 응용프로그램 작성
- 인터페이스 구현 및 상속 구조 작성

Android

- Google Play store에 앱 배포 및 버전관리
- 외부 API 연동 (SNS 로그인, 맵)
- 오픈 API 사용을 위한 XML / JSON 데이터 파싱
- POST / GET 방식을 통한 REST API 사용
- 호스팅 서버(https://www.dothome.co.kr/)를 통한 MySQL DB서버 활용
- 외부라이브러리 GSON을 활용하여 HTTP 통신으로 데이터 업/다운 다운로드
- 마테리얼 디자인 인터페이스 구현 및 라이브러리 활용

Kotlin

- open API 데이터를 REST 방식으로 받아와 recycler view 구현
- 함수형 언어에 대한 기본 이해 및 프로그래밍 작성

HTML 5 / CSS 3

- HTML5 기본 태그 문법 숙지 및 Layout 구성에 관한 이해
- CSS3를 이용한 기초적인 Front End 관련 스킬
- 시맨틱 요소를 활용한 웹페이지 기본 구성 학습
- Java Script 기본 문법 학습

React-Native

- React-Native 기본 태그 문법 학습 및 화면 구현

C / C++

- 절차지향 프로그래밍에 대한 이해 및 연산자 등 프로그래밍의 기본 문법 학습

형상관리

- Git hub repository를 이용한 개인/팀 프로젝트 형상관리
- https://github.com/jmnl225?tab=repositories

1인 개발

소사 프로젝트

개발 인원	1인
개발기간	4주
프로젝트 설명	학생 출결관리 안드로이드 앱

소사는 소리사랑의 준말로, 클라이언트가 운영하는 '소리사랑 음악학원'의 출결관리 애플리케이션으로 제작하게 되었습니다.

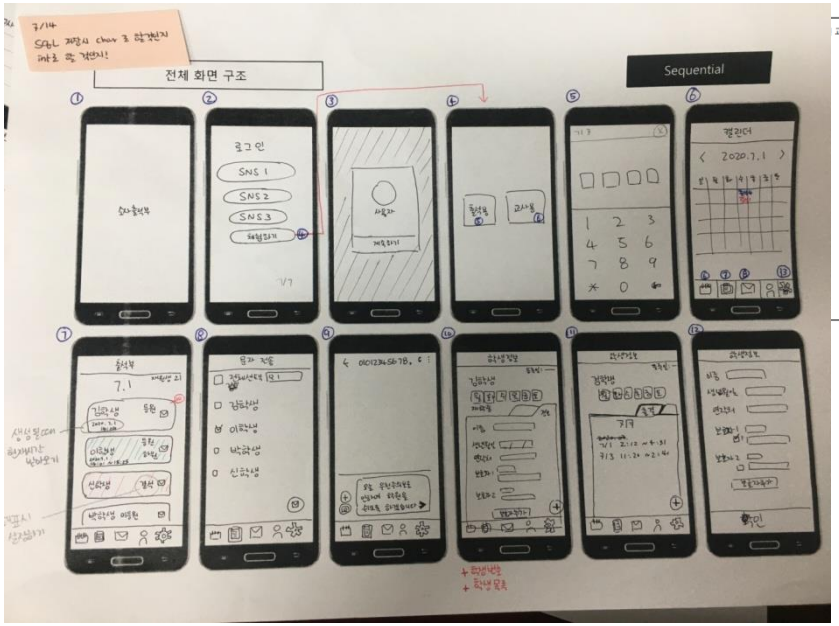
학생의 등원/퇴원 시 등록된 학생번호를 입력하고, 출석이 되었다는 메시지가 학부모에게 자동으로 문자가 전송되는 것이 주요 기능입니다.

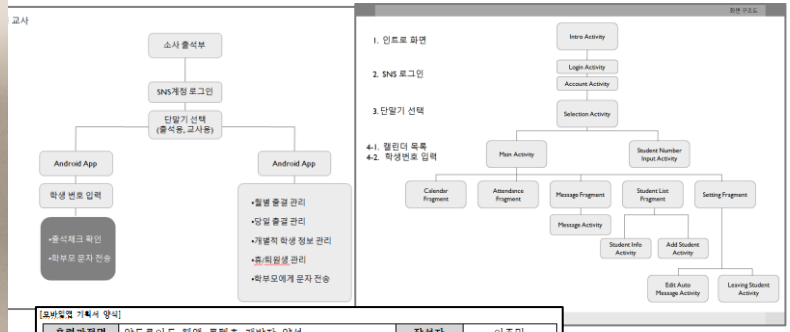
PlayStore: https://play.google.com/store/apps/details?id=com.jmnl2020.attendanceapp3
GitHub: https://github.com/jmnl225/AttendanceAppProject

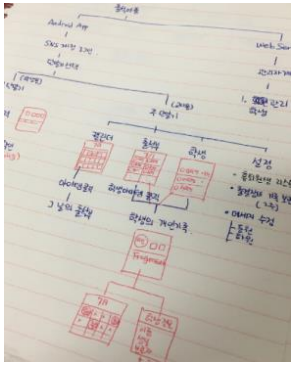
1인 개발

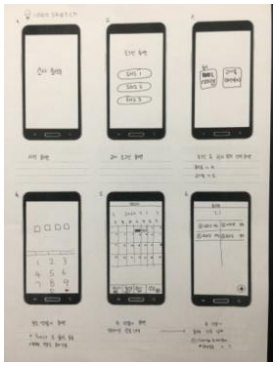
기획서 작성

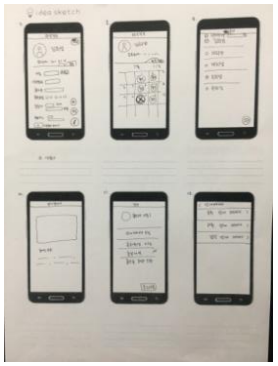
- 클라이언트의 요구사항에 맞춰 소규모 학원, 교사를 위한 출석 애플리케이션 기획
- 기존의 앱 리서치 및 분석 후 화면, 프레임 설계











[포바일앱 기획서 양식]

훈련과정명	안드로이드 웹앱 콘텐츠 개발자 양성		작성자	이주민
팀 구성	1인	팀장	이주민 - 총괄 제작	
		팀원-server	이주민 - 서버, DB 및 관리자단 페이지 설계	
		팀원-client	이주민 - 앱 기능구현 및 레이아웃	

1. 애플리케이션 명

소사 출석부

2. 애플리케이션 기획 배경(의도)

한국의 초등학교 저학년 학생들은 피아노, 미술, 그 외의 다양한 취미 활동을 원할하게 해주는 소규모 학원에 재학중이다. 소규모 학원들을 위해 학원과 학부모가 더 원활하게 학원생들을 관리할 수 있도록 학생 관리 차원의 출결 애플리케이션을 제작한다.

클라이언트의 요구: 아직 핸드폰을 가지지 못한 학생들의 학원 출결 상황 관리 애플리케이션 필요

1. 선생님의 학원생 출결을 일별, 월별로 확인 가능하게 할 것
2. 학부모들에게 원생의 출결상황을 실시간으로 전송
3. 학생의 생일을 교사에게 알려주는 이벤트

3. 애플리케이션 주요기능

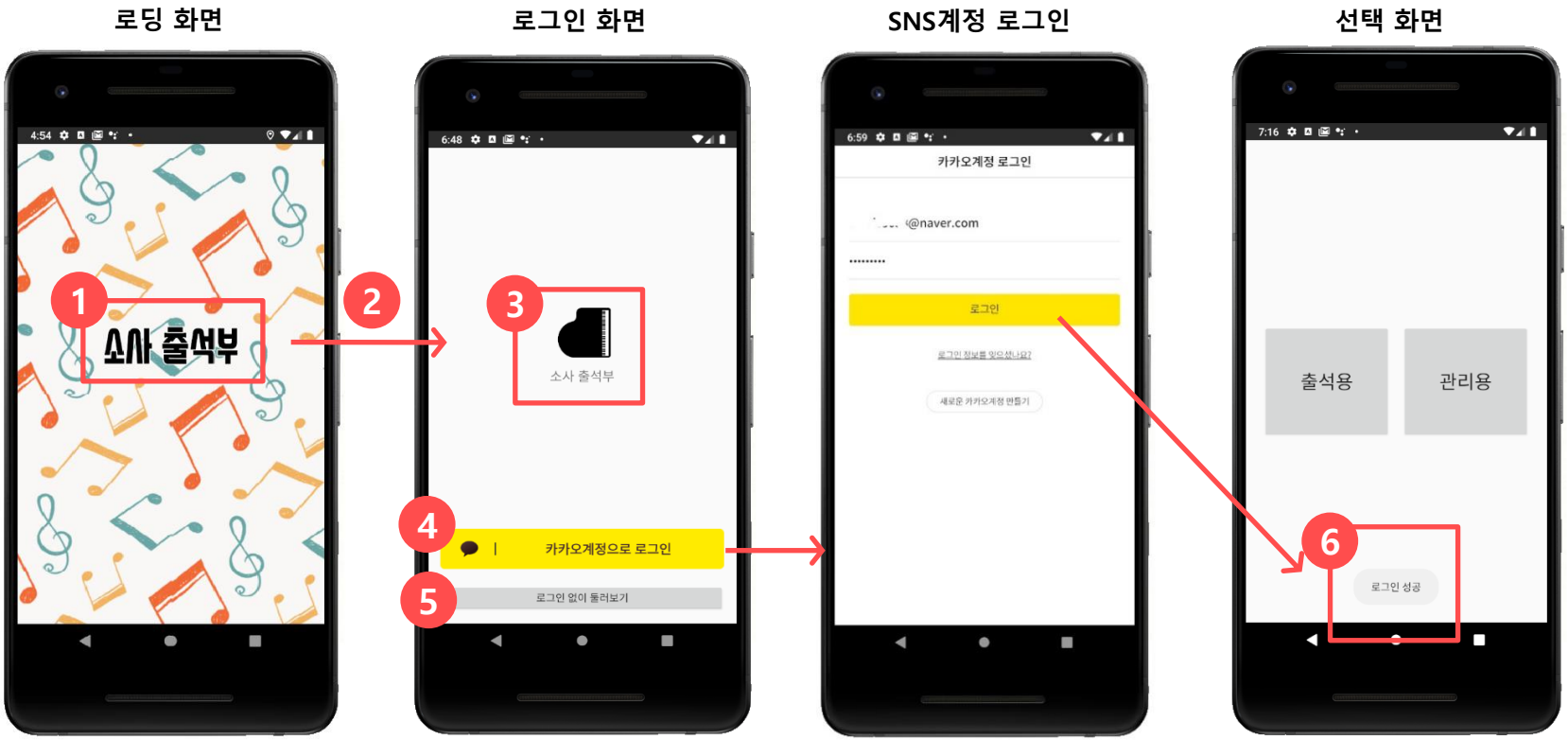
소규모 학원을 위한 출결 앱으로서 다음과 같은 기능을 포함한다.

1. 학생들의 출결을 한 눈에 알아볼 수 있도록 일별로 출결 상황 제공
2. 개별적 출결 관리를 위해 학생 개인 정보 수정/ 일람
3. 월별 재원생들의 출결 상황을 파악할 수 있도록 캘린더 화면
4. 학생의 등원에 흥미를 가질 수 있도록 생일 이벤트를 교사에게 알림

4. 주요 사용대상(타겟)

소규모 학원의 교사

화면 설명



① 무료 배포된 폰트 사용 : font 폴더를 생성하여 다운받은 폴더를 특정 TextView에 적용

② 2초 후 자동으로 로그인화면으로 이동 : handler 를 이용한 Activity 전환을 지연시킨 뒤 실행

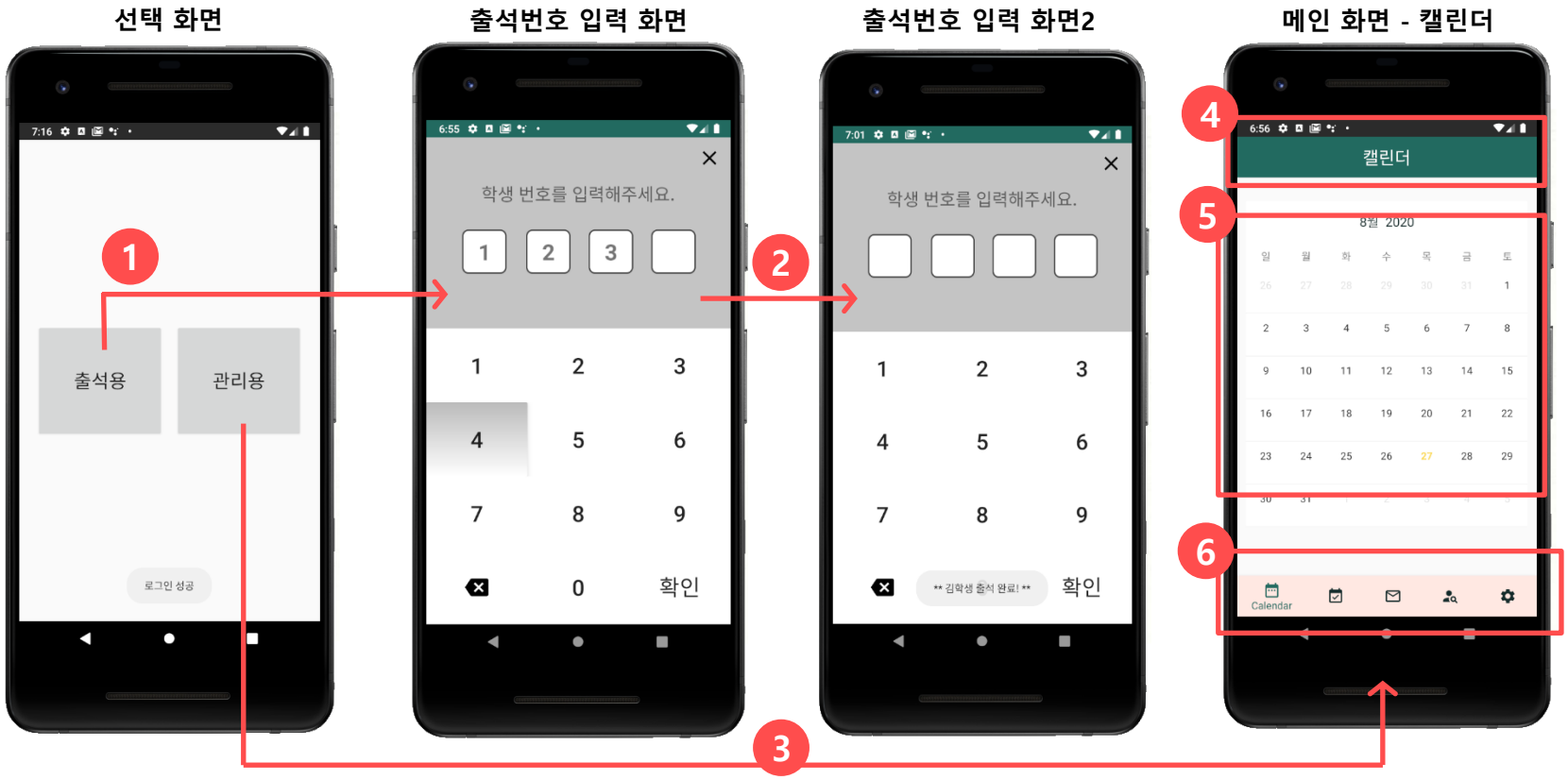
③ 일러스트레이터를 활용하여 그린 이미지 활용

④ SNS 계정 연동 로그인 : 오픈 API 사용. Shared preference에 로그인 데이터를 저장하여 한번 로그인을 한 뒤에는 자동로그인 설정

⑤ 로그인 없이 앱의 기능을 둘러볼 수 있도록 샘플 데이터 제공

⑥ 용도에 따라 화면 선택: 로그인에 성공하면 Toast로 확인메세지 팝업

화면 설명



① 출석용 버튼 클릭시 출석번호 입력화면으로 이동

② 등록된 학생들의 학생번호를 입력하면 출석체크등록.

③ 관리용 버튼 클릭시 메인 Fragment로 이동.

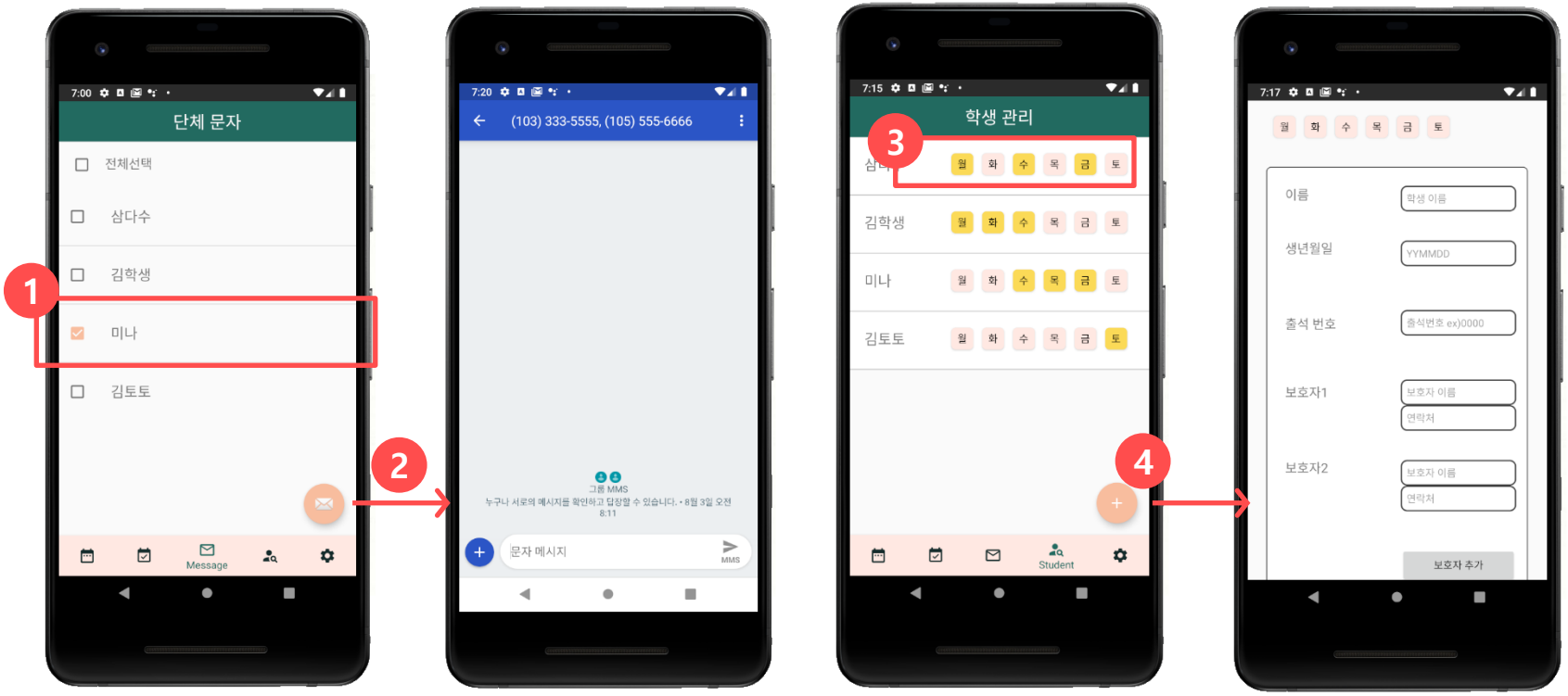
④ 커스텀 Appbar : 앱의 테마 컬러에 맞춰 Fragment에 적용

⑤ GitHub 오픈소스 캘린더 활용

⑥ 마테리얼디자인 Bottom Navigation bar : Main Activity에 다수의 Fragment를 적용

화면 설명

학생 선택 화면 | 문자 전송 화면 | 학생 관리 화면 | 학생 추가 화면



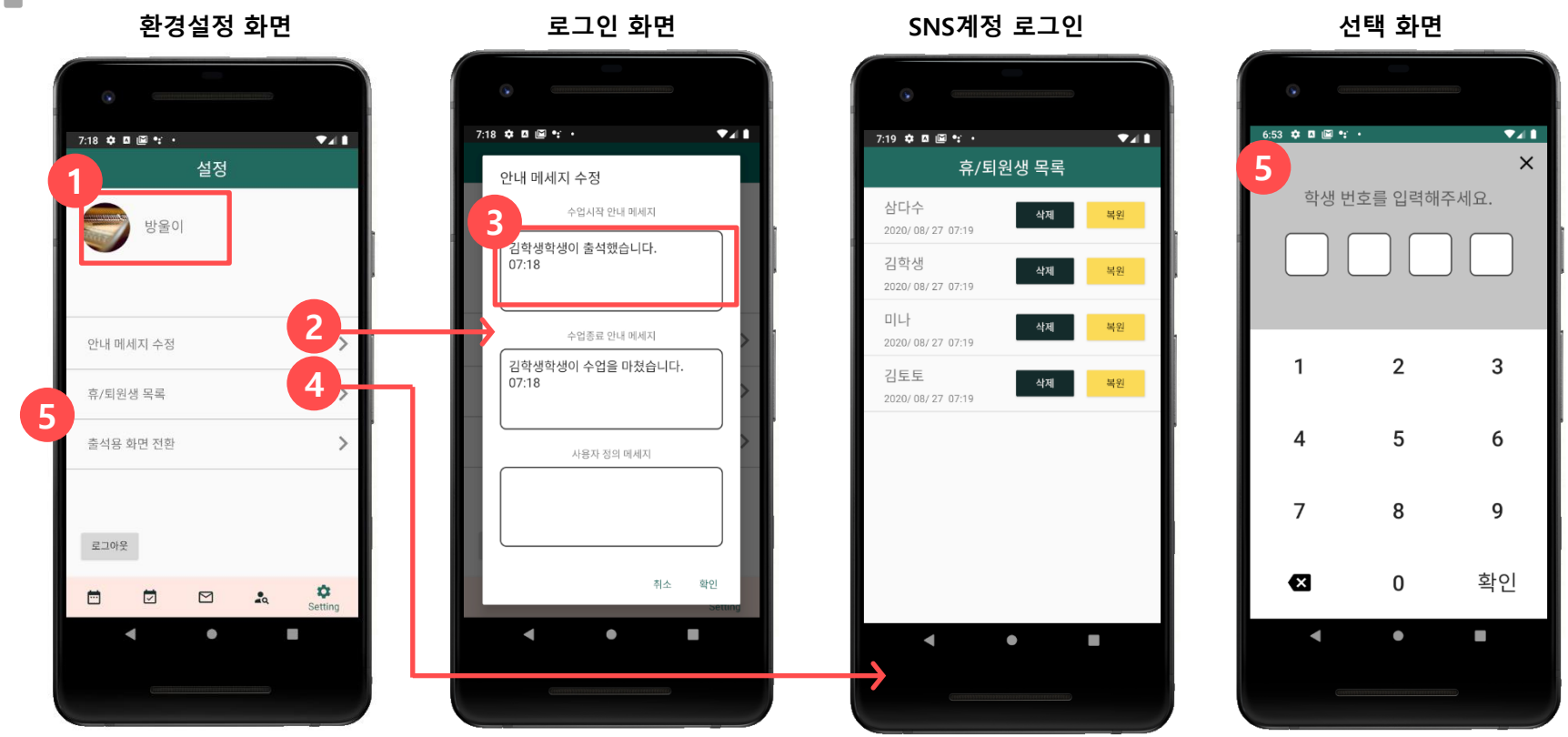
① 학생 목록에서 문자보낼 학생 개별 선택 : Recycler View에 커스텀 Adapter를 활용하여 학생 목록 구현

② 선택된 학생데이터에 저장된 번호로 메시지 전송 : Floating Action Button 구현, Intent를 사용하여 Native Message 앱 실행

③ 학생 수업 요일 기록 : 비트연산자를 활용하여 INT변수에 학생별 요일 데이터 저장

④ 학생 데이터 추가 : Floating Action Button을 통해 Activity 전환. / DB에 학생 데이터 업로드

화면 설명



① SNS 계정의 정보를 불러와서 현재 사용자 정보 표시

② 전송되는 안내 메시지 수정 : Custom Dialog 적용

③ 학생번호를 등록한 시간 저장, 안내 메시지에 출력

④ 기존 학생들의 정보 삭제 , 복원할 수 있는 화면

⑤ 출석 번호 입력 화면으로 이동

기술 설명

학생 정보 저장 화면

2 외부 라이브러리를 활용한 데이터 업로드 코딩

```
//레트로핏 라이브러리로 데이터 전송
Retrofit retrofit = RetrofitHelper.getInstance();
Log.i( tag: "TAG",  msg: "retrofit");

//추상메소드 활용
RetrofitService retrofitService = retrofit.create(RetrofitService.class);
Log.i( tag: "TAG",  msg: "retrofitService");

//데이터
Map<String, String> dataPart = new HashMap<>(); //모내야하는 값이 int, String 등 하나 이상일때는 object
dataPart.put("day", day + "");
dataPart.put("name", name);
dataPart.put("birthday", birthday + "");
dataPart.put("contact", contact + "");
dataPart.put("par1name", par1name);
dataPart.put("par1phone", par1phone + "");
dataPart.put("par2name", par2name);
dataPart.put("par2phone", par2phone + "");
Log.i( tag: "TAG",  msg: "Put data");

//데이터 전송!
Call<String> call = retrofitService.postData(dataPart);
Log.i( tag: "TAG",  msg: "Before enqueue"); //확인


call.enqueue(new Callback<String>() {
    @Override
    public void onResponse(Call<String> call, Response<String> response) {
        if (response.isSuccessful()) {
            Log.i( tag: "TAG",  msg: "enqueue");

            String s = response.body();
//              AlertDialog.Builder builder = new AlertDialog.Builder(StudentEditActivity.this);
//              builder.setMessage(s+"").show();
            Toast.makeText( context: StudentEditActivity.this,  text: s + "", Toast.LENGTH_SHORT).show();

            //액티비티 종료
            finish();
        }
    }
```

```
package com.jmnl2020.attendanceapp3;

import ...

public class RetrofitHelper {

//    static OkHttpClient client = new OkHttpClient();

    public static Retrofit getInstance(){

        Retrofit.Builder builder = new Retrofit.Builder();
        builder.baseUrl("http://projectjm.dothome.co.kr/");
        builder.addConverterFactory(ScalarsConverterFactory.create());

        Log.i( tag: "TAG",  msg: "addConverterFactory");

        return builder.build();
    }

    public static Retrofit getInstance2(){

        Retrofit.Builder builder = new Retrofit.Builder();
        builder.baseUrl("http://projectjm.dothome.co.kr/");
        builder.addConverterFactory(GsonConverterFactory.create());

        return builder.build();
    }

}
```

```
package com.jmnl2020.attendanceapp3;

import ...

public interface RetrofitService {

    //데이터와 파일을 동시에 전송
    @Multipart
    @POST("/Retrofit/insertDB.php")
    Call<String> postData(@PartMap Map<String, String> dataPart);

    // 서버에서 데이터 json을 읽어와서 GSON라이브러리를 통해 곧바로 자바 객체로 응답결과를 주는 추상메소드

    @GET("/Retrofit/loadDB.php")
    Call<ArrayList<StudentDTO>> loadData();

}
```

기술 설명

3 PHP 파일 작성

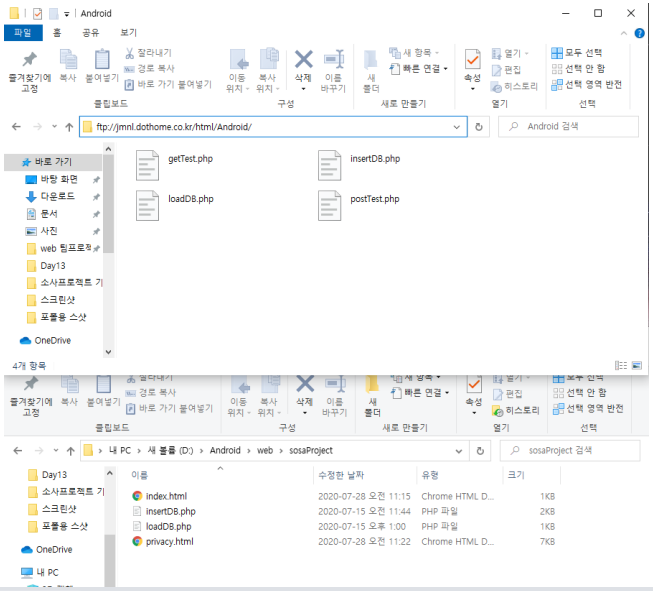
insertDB.php

```php
<?php

    header('Content-Type:text/plain; charaset=utf-8');

    $day = $_POST['day'];
    $name = $_POST['name'];
    $birthday = $_POST['birthday'];
    $contact = $_POST['contact'];
    $par1name = $_POST['par1name'];
    $par1phone = $_POST['par1phone'];
    $par2name = $_POST['par2name'];
    $par2phone = $_POST['par2phone'];


    $now = date('YYmmdd');

    //DB테이블에 데이터 저장
    $conn = mysqli_connect('localhost', 'projectjm', [redacted]');
    mysqli_query($conn, "set names utf8");

    //쿼리문 작성 [주의!!! tinyint는 value 입력시 ''가 없어야한다!]
    $sql="INSERT INTO students(day, name, birthday, contact, par1name, par1phone, par2name, par2phone)
    VALUES('$day','$name','$birthday','$contact','$par1name','$par1phone','$par2name','$par2phone')";

    $result = mysqli_query($conn, $sql);

    if($result) echo "데이터 업로드가 완료되었습니다.";
    else echo "업로드에 실패했습니다.";

    mysqli_close($conn);
```

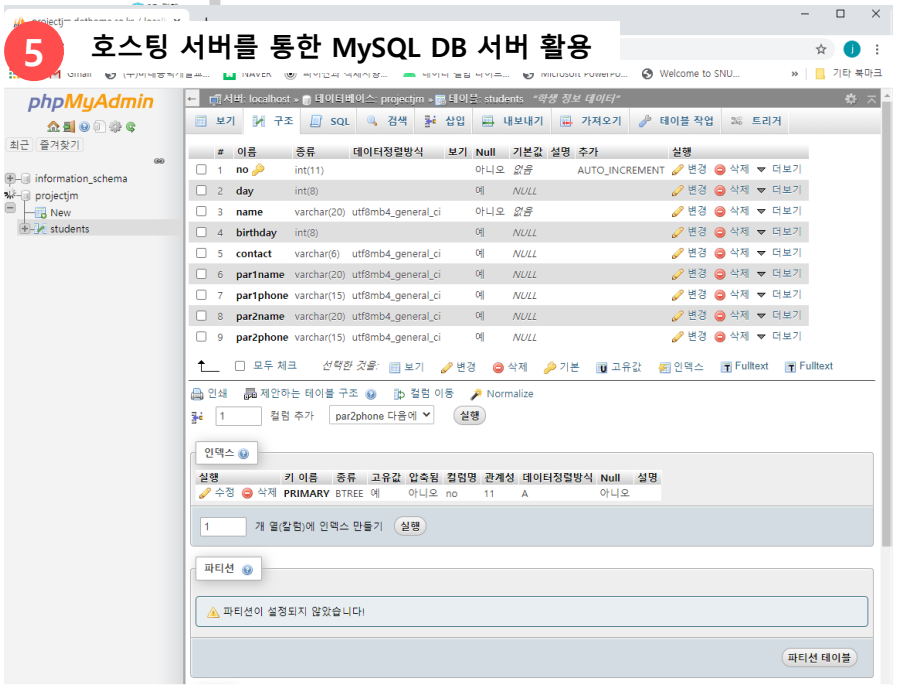
loadDB.php

```php
<?php

    header('Content-Type:Application/json; charset=utf-8');

    $conn = mysqli_connect('localhost', 'projectjm', [redacted]');
    mysqli_query($conn, "set names utf8");

    $sql="SELECT*FROM students";
    $result = mysqli_query($conn, $sql);

    //결과표($result)의 총 레코드 수
    $row_num=mysqli_num_rows($result);

    //여러줄을 읽어야 하므로 각 줄($row배열)요소를 가질 인덱스배열 준비
    $rows= array();
    for($i=0; $i<$row_num; $i++){
        $row= mysqli_fetch_array($result, MYSQLI_ASSOC);
        $rows[$i] = $row;
    }

    //2차원 배열 --> json array로 변환
    echo json_encode($rows);

    mysqli_close($conn);
?>
```

4 파일 업로드



5 호스팅 서버를 통한 MySQL DB 서버 활용



1인 개발

주간 농사정보

개발 인원 1인

개발기간 1주 미만

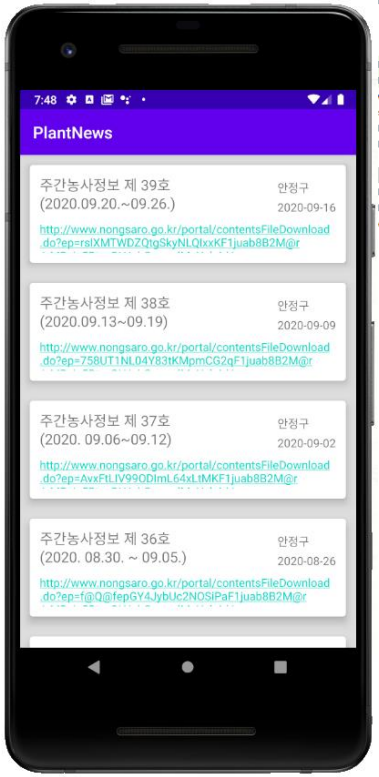
프로젝트 설명 농림청 공공 API 정보 데이터 파싱 연습용 앱

Kotlin 을 이용하여 공공 api 파일을 XML파싱하는 연습을 위해 만든 앱입니다. 서버에서 정보를 받아와서 Recycler View로 구현했습니다.

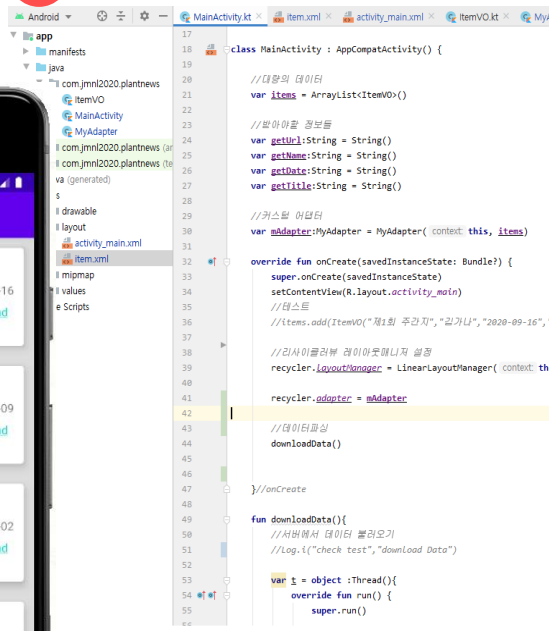
GitHub: https://github.com/jmnl225/Android-practice/tree/master/PlantNews

기술 설명

1 메인 화면



2 XmlPullParser 코딩 화면




```
//url만들면서 URL과 연결될 인터넷 주소
var url: URL = URL(address)

var inputStream:InputStream = url.openStream()
var isr:InputStreamReader = InputStreamReader(inputStream)

//XML문서를 분석해주는 객체 소환
var xpp: XmlPullParser
//xpp 생성
var factory:XmlPullParserFactory = XmlPullParserFactory.newInst
xpp = factory.newPullParser()
xpp.setInput(isr)

Log.i( tag: "check test", msg: "파싱 준비 완료")

//여기까지 파싱 준비//////////////////////////////////////

//가져와야할 정보 : download url, writer, title, date

var eventType:Int = xpp.eventType

while (eventType != XmlPullParser.END_DOCUMENT){

    when (eventType){
        XmlPullParser.START_DOCUMENT -> runOnUiThread(object :R
            override fun run() {
                Toast.makeText( context: this@MainActivity, text: "
                //Log.i("check test","파싱시작")
            }
        })

        XmlPullParser.START_TAG -> {
            var tagName: String = xpp.name

            //Log.i("check test","스타트 태그")

            if(tagName == "downUrl"){
                xpp.next()
                getUrl=xpp.text
            }else if(tagName == "regDt"){
                xpp.next()
                getDate=xpp.text
```

```
XmlPullParser.START_TAG -> {
    var tagName: String = xpp.name

    //Log.i("check test","스타트 태그")

    if(tagName == "downUrl"){
        xpp.next()
        getUrl=xpp.text
    }else if(tagName == "regDt"){
        xpp.next()
        getDate=xpp.text
    }else if(tagName == "subject"){
        xpp.next()
        getTitle=xpp.text
    }else if(tagName == "writerNm"){
        xpp.next()
        getName = xpp.text
    }
}//startTag

XmlPullParser.TEXT -> { }
XmlPullParser.END_TAG -> {
    //Log.i("check test"," ============ 파싱 끝

    var tagName2 : String!  = xpp.name
    if (tagName2 == "item" ){
        //Log.i("check test", "태그네임2")
        items.add(ItemVO(getTitle, getName, get

        runOnUiThread(object:Runnable{
            override fun run() {
                //Log.i("check test","run on ui

                //items.add(ItemVO(getTitle, ge
                // ->아이템을 여기서 추가하면 갱
                mAdapter.notifyDataSetChanged()
```

---

① MainActivity 에서 데이터 파싱을 통해 받은 정보를 Recycler View로 구현

② Api key를 발급받아 공공데이터 주소에 접속하여 필요한 정보를 받아오는 코드 작성

그룹 개발

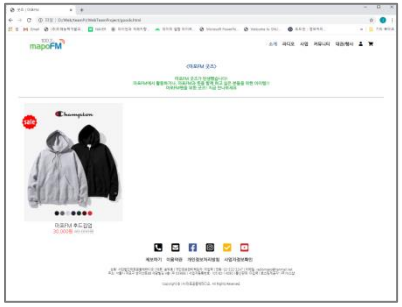
웹 개발 프로젝트

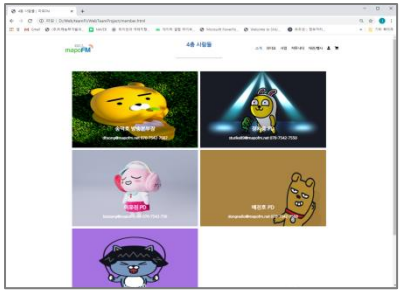
개발 인원	5인
개발기간	3주
프로젝트 설명	마포 FM 웹 퍼블리싱 프로젝트
	JavaScript, Html 5, CSS 3 을 활용하여 구현했습니다. 모바일에서도 웹을 사용자의 편의에 맞춰 제공하기 위해 반응형 웹 제작 프로젝트입니다.
	현재 서비스를 제공중인 https://www.mapofm.net/ 를 참조했습니다.

GitHub: https://github.com/JasonOh93/WebTeamProject.git

화면 설명

1 웹 화면 구성





2 미디어쿼리를 통한 반응형 웹 적용

member.css

Web > teamPJ > WebTeamProject > css > # member.css > ...

```
}


/* 반응형 웹 적용 (RWD) */
/* 미디어쿼리 적용 (해상도에 반응하기) */
/* 화면너비가 640px 보다 작을 때 {}안의 스타일 적용 */
@media screen and (max-width: 640px){

    body{
        padding-left: 5px;
        padding-right: 5px;

        text-align: justify;
    }

    section{
        padding: 20px;
    }

    .row{
        flex-direction: column;
    }


}


@media screen and (max-width: 320px){

    article{
        width: 300px;
    }

}
```

```
@media screen and (max-width: 640px){

    p{
        font-size: 0.9em;
    }

    body{
        /* padding-left: 5%; */
        padding-left: 0%;
        padding-right: 0%;
    }

    section{
        padding-left: 0%;
        padding-right: 0%;
    }

    article{
        width: 100%;
    }

    #img{
        width: 100%;
        height: 100%;
    }

    .subtitle{
    width: 300px;
    font-size: 1.5em;
    font-weight: bold;
```

① 시멘틱 요소를 사용하여 웹사이트 UI 구성, 구글 API 활용.

② CSS 3을 통해 해상도에 따라 화면 구성이 달라지는 반응형 웹