Android Application 개발 예시

상명대학교 컴퓨터과학과 졸업프로젝트 MinD팀
고 진 권

[Index]

0. 소 개

1. 여.기.다(여행 갈 기회는 다시 오지 않는다)

숙박권 양도 어플리케이션

2018년 ‘멋쟁이 사자처럼’ 동아리 활동

개발 기간 :: 2018.05 ~ 2019.02(10개월)

2. Shalendar (Shared Calendar)

공유 달력 어플리케이션

shalendar

우리의 만남을 공유하다

2019년 컴퓨터과학과 졸업 프로젝트

개발 기간 :: 2019.04 ~ 2019.09(6개월)

0. 소 개

MinD팀 소개

지도교수

민경하

MinD 팀 총괄

고진권

PM 및 안드로이드 개발
프로젝트 총괄

박지상

최종보고서 작성 및
안드로이드 개발
최종보고서 작성 총괄

박성준

Class, UI 설계 및
안드로이드 개발
Class, UI 설계 총괄

김형택

요구사항 분석 및
서버 개발
요구사항 분석 총괄

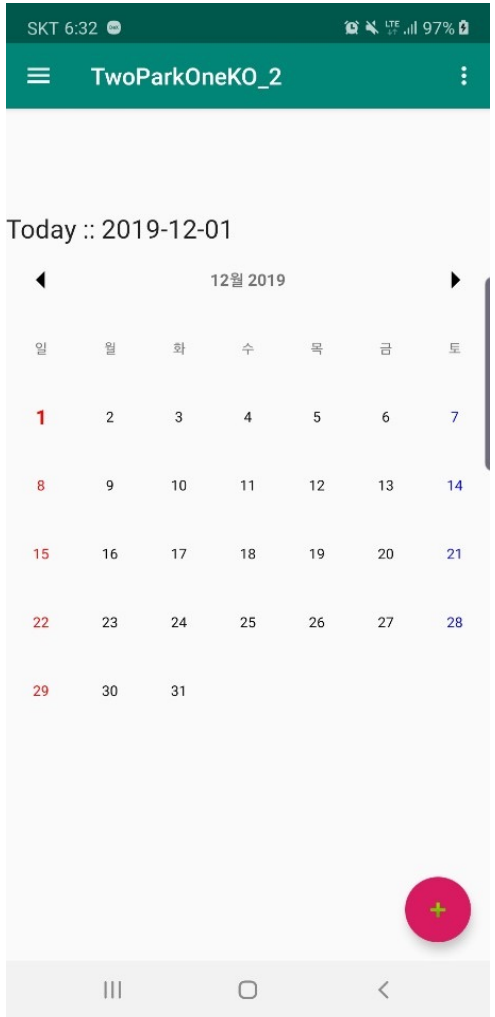
정구일

테스트 케이스 작성
및
서버 개발
테스트 케이스 총괄

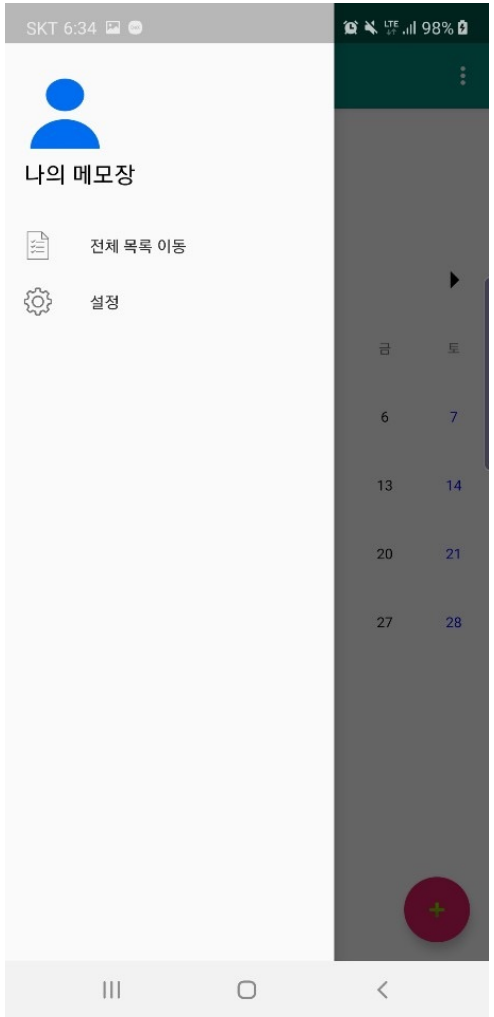
1. Kotlin 프로젝트 VS 졸업 프로젝트

1.1 Kotlin 프로젝트 :: Memo Application

왜 하필 Memo App..?? → 구현 부분에 충실



<Main Activity>



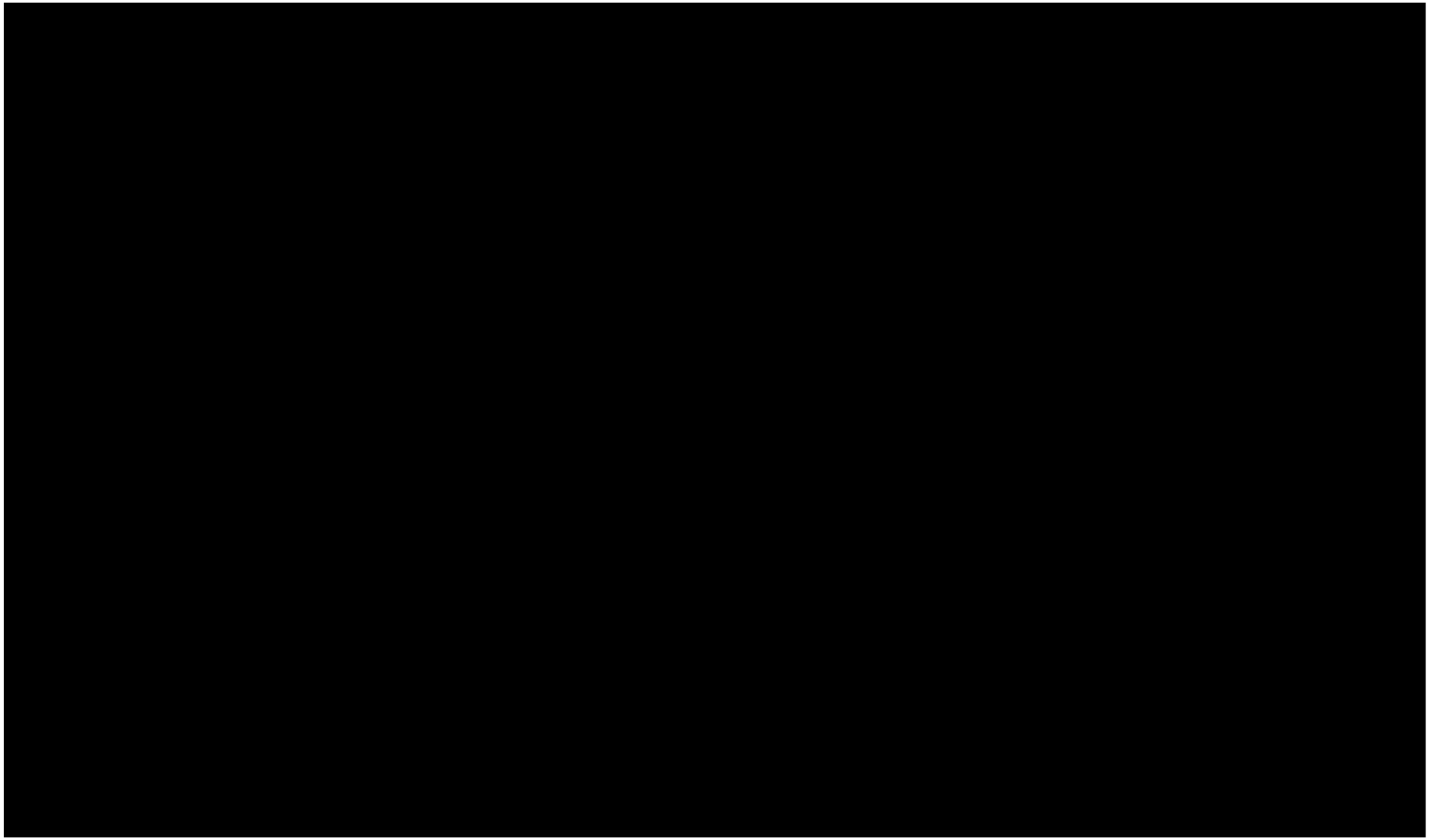
<SideBar Menu>

<RegisterPlan Activity>

<Board Activity>

1. Kotlin 프로젝트 VS 졸업 프로젝트

1.2 졸업 프로젝트 :: Shalendar



개발 동기

shalendar
우리의 만남을 공유하다

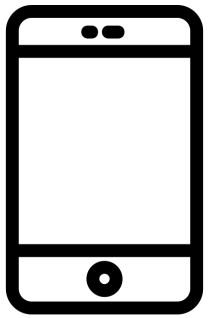
Q1. 팀 단위의 일정들을 앱에서 공유하면 편하지 않을까?

< 월간 행사표와 같은 화이트 보드 >

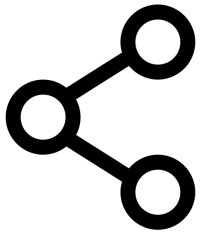
Q2. 팀원의 일정을 파악하고 모두가 가능한 시간대를 추천해준다면 편하지 않을까?

목 적

shalendar
우리의 만남을 공유하다



공유 달력 앱 개발



개인 혹은 다수의 일정을 효과적으로 공유 및 관리

1. Kotlin 프로젝트 VS 졸업 프로젝트

1.2 졸업 프로젝트 :: Shalendar

2. 완성도 있는 Android application 개발

#. 조금 더 완성도 높은 Application을 개발하기 위해서는 어떠한 요소들이 추가..??

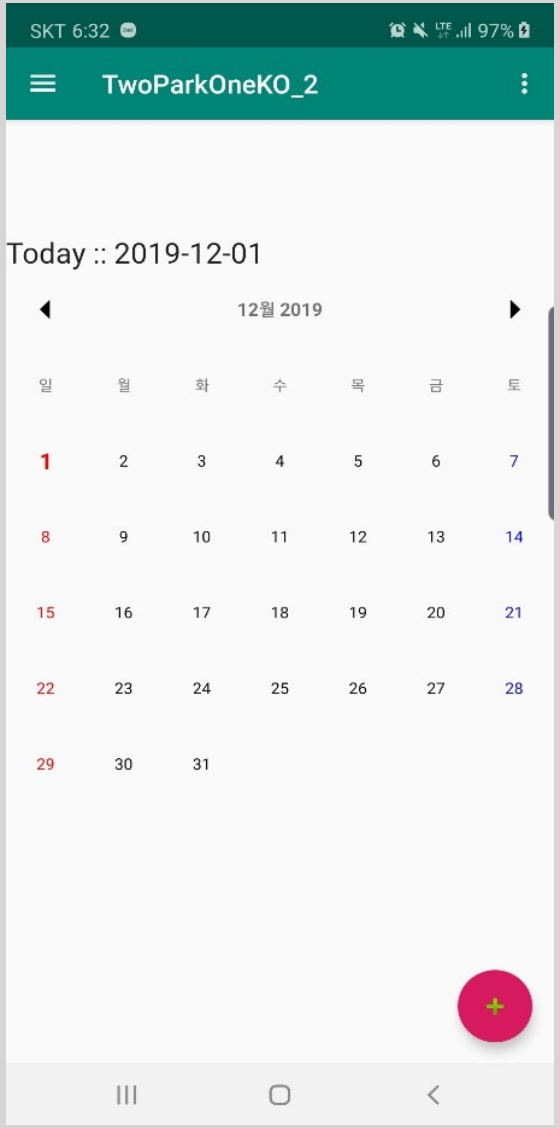
Kotlin 프로젝트 → Shalendar 프로젝트

1. 디자인
- 사용자를 압도할 수 있는 시각적인 효과
- 어플 사용의 편리성 제공

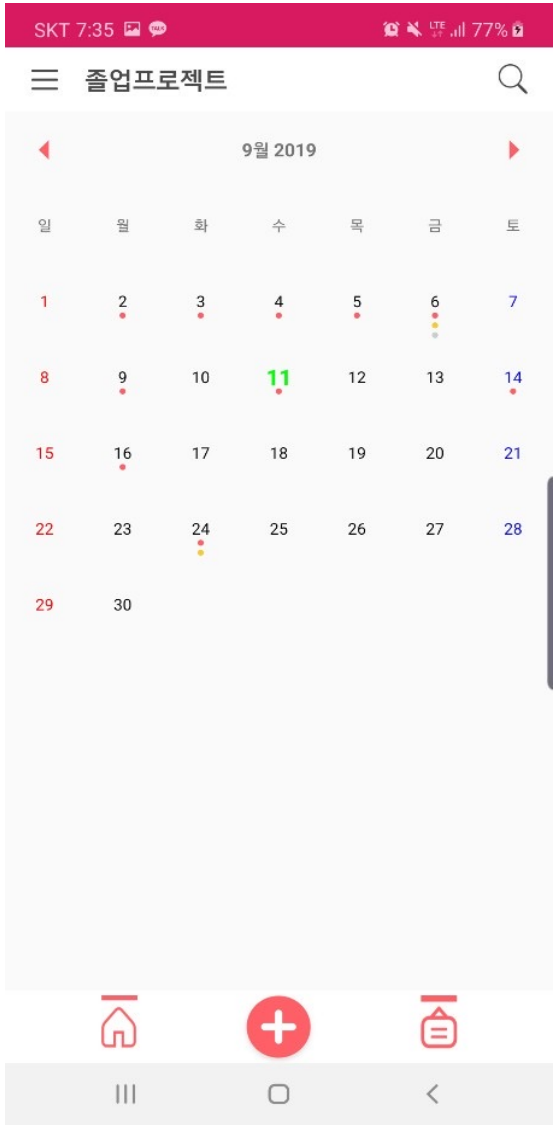
2. 추가 기능 구현
- 사용하고자 하는 라이브러리 추가
- 핵심 기능을 선정하고 부가적인 기능 추가
- Firebase, push알림 서비스 등 사용

3. 서버 구현
- 사용자 data를 저장할 서버 구축(AWS, Apache Tomcat)
  (안드로이드의 SharedPreference만으로는 한계)
- **안드로이드와 서버 간 연동(통신)**
- 서버의 저장 data를 안드로이드에서 자유롭게 사용

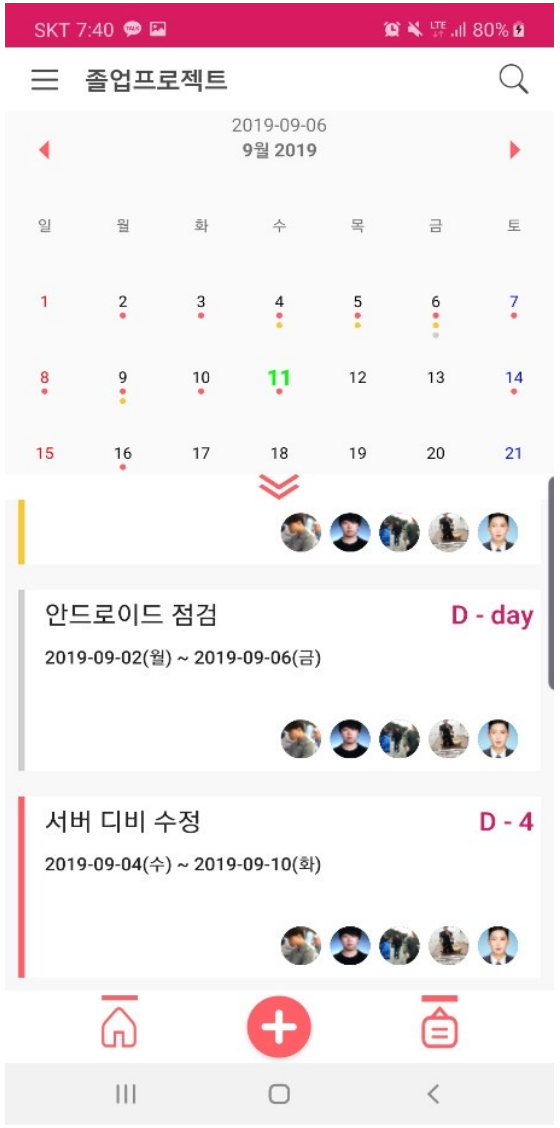
2.1 디자인 요소 추가



<Main 화면>

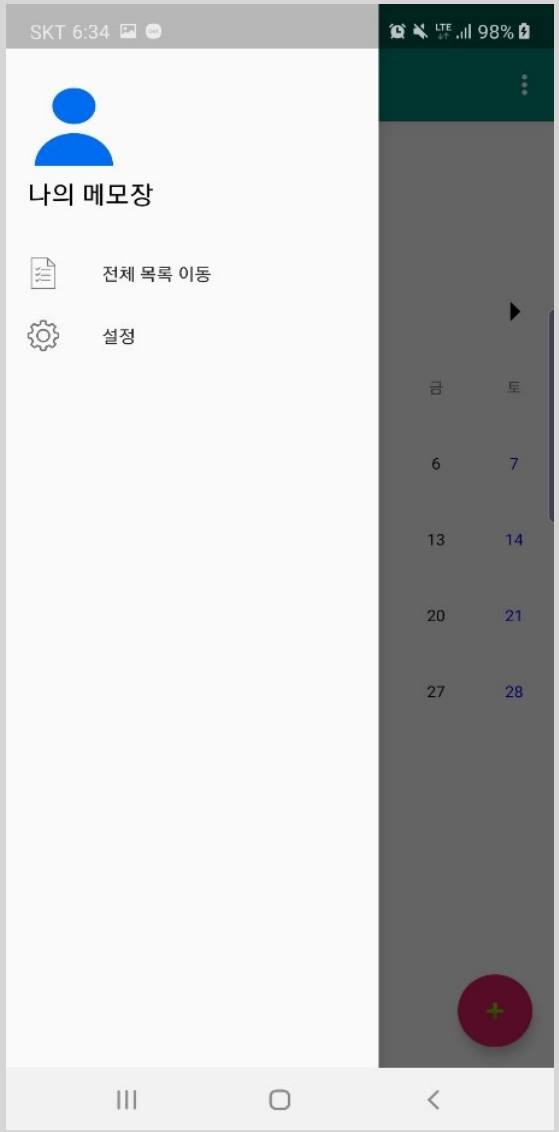


<Main 화면>

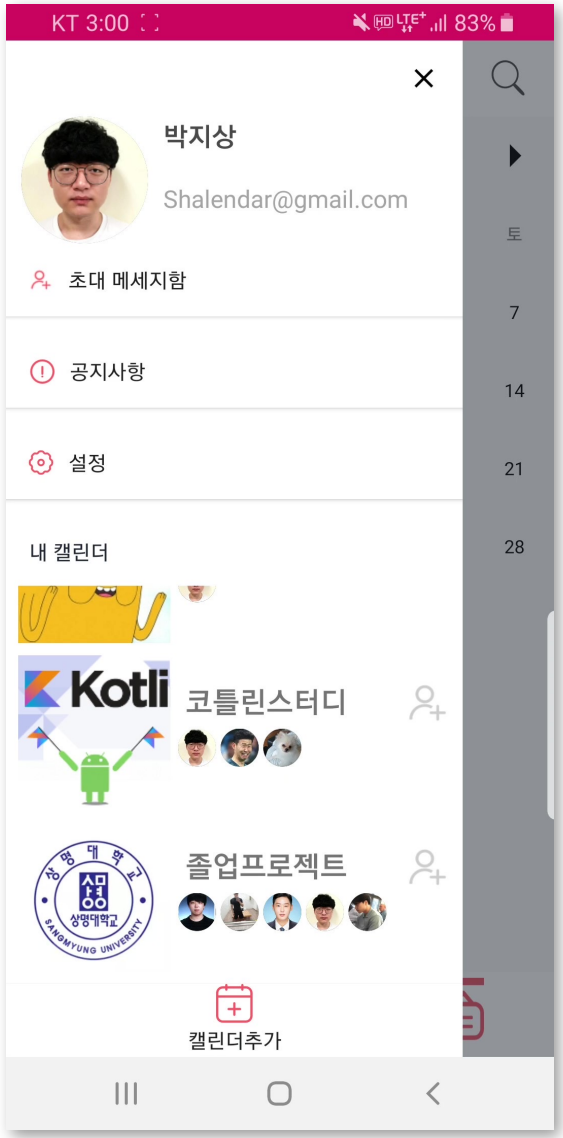


<Animation속성 추가 Main 화면>

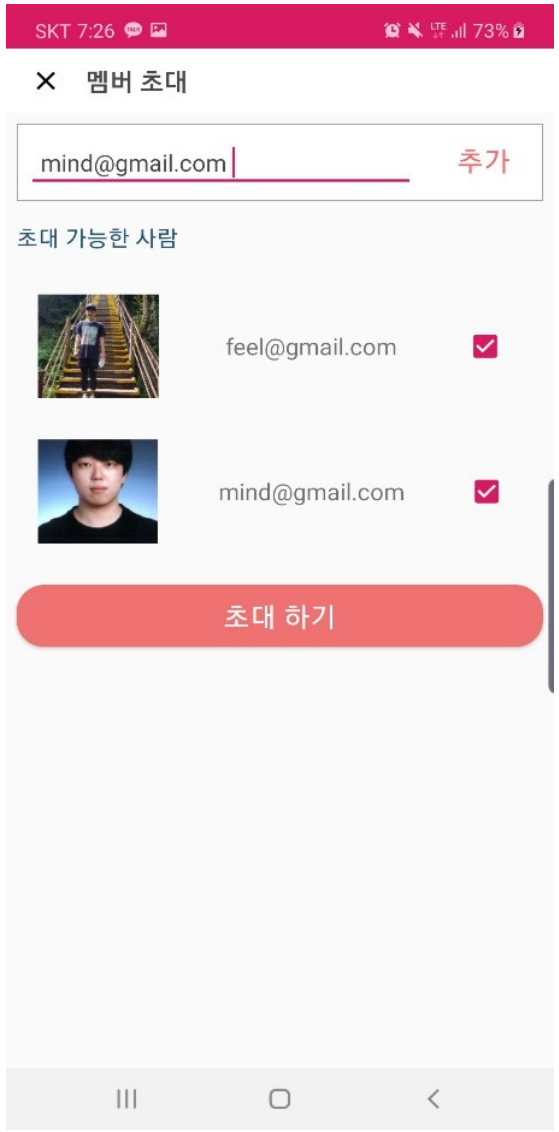
2.1 디자인 요소 추가



<SideBar Menu 화면>



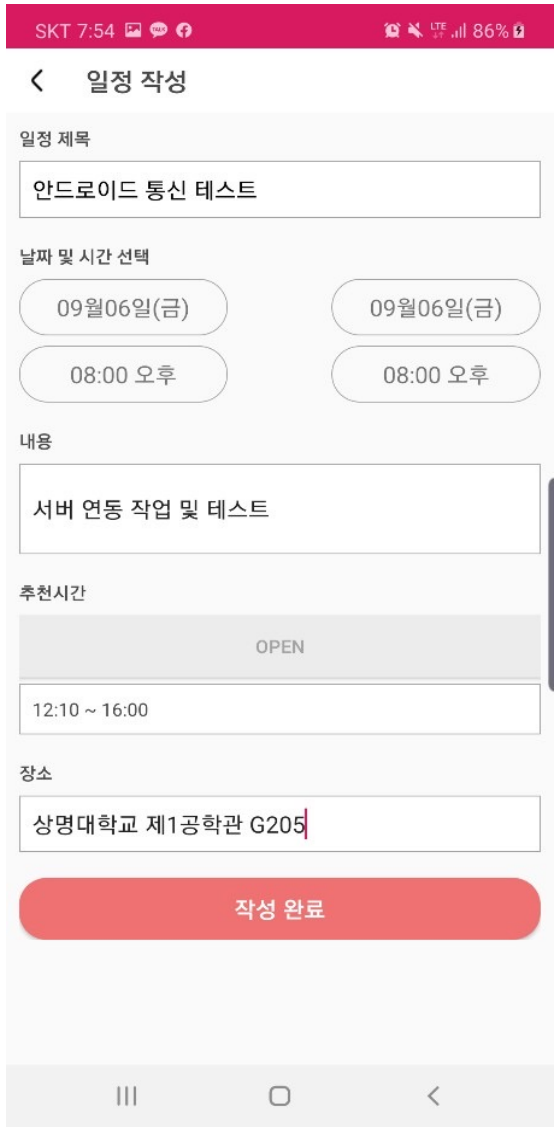
<SideBar Menu 화면>



<SideBar의 멤버 초대 화면>

2.1 디자인 요소 추가

<일정 작성 화면>



<공유 일정 작성 화면>

<공유 일정 화면 Date Picker>

2.1 디자인 요소 추가

<일정 게시판 화면>

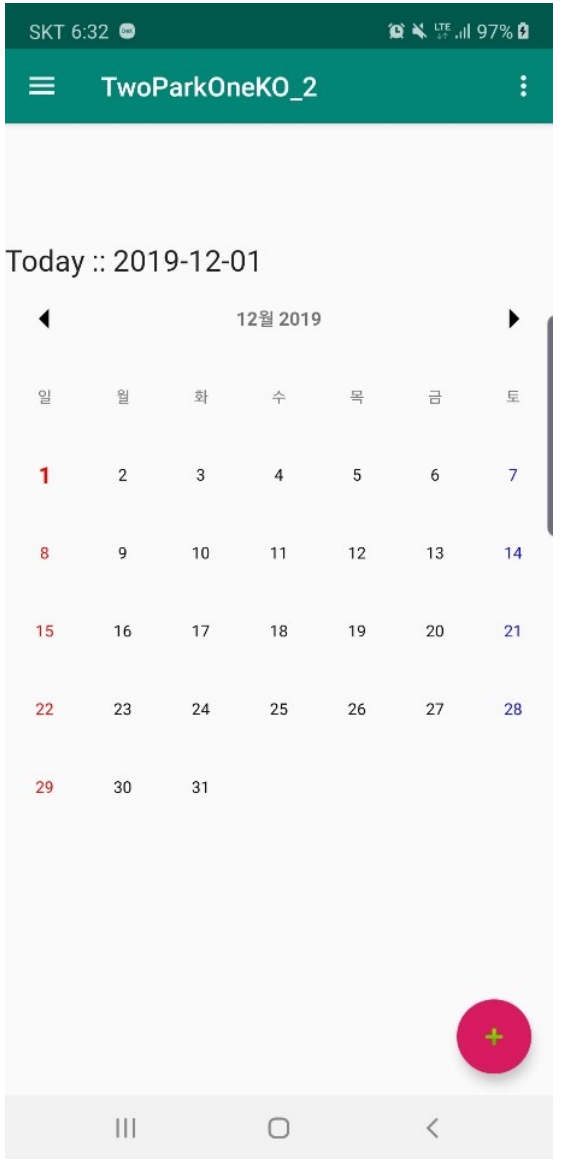
<공유달력 게시판 화면>

<공유달력 일정 상세 화면>

2.1 디자인 요소 추가

#. Kotlin 프로젝트 :: Memo Application

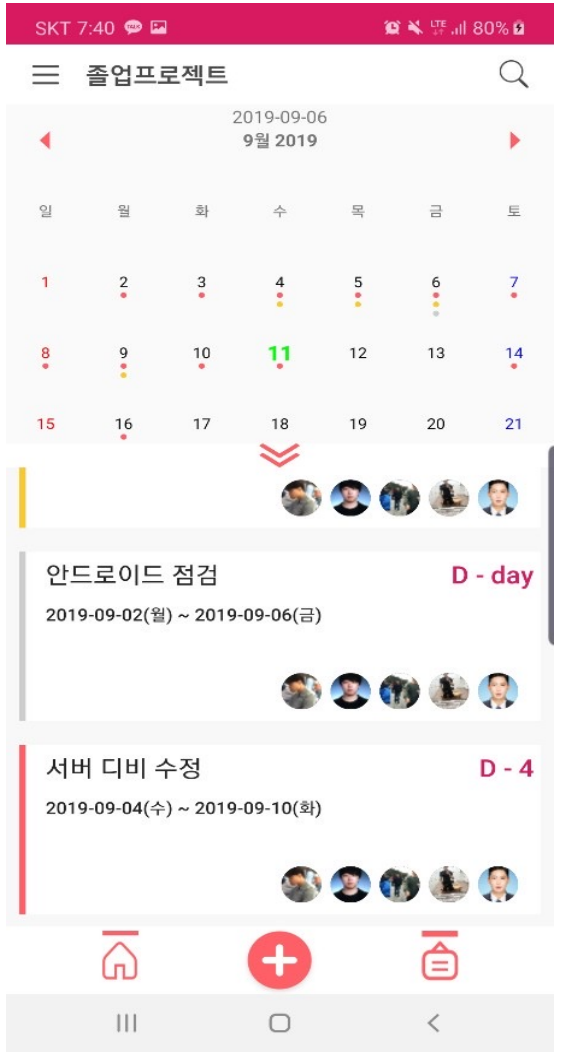
<OvenApp을 사용한 Main화면 디자인 초안>



<실제 구현 Main 화면>

2.1 디자인 요소 추가

#. 졸업 프로젝트 :: Shalendar Application



<OvenApp을 사용한 Main화면 디자인 초안>

<Zeplin 사용 디자인 최종안>

<실제 구현 Main 화면>

2.2 추가 기능 구현 및 라이브러리 적용

#. 구체적인 기능 정의 및 구현(기본 기능 & 핵심 기능 선정)

1. 기본 기능

- 로그인
- 사용자 프로필 사진 설정
- 공유 달력 배경 사진 설정
- 개인 달력 생성
- 해당 일정의 D-day 표시 기능

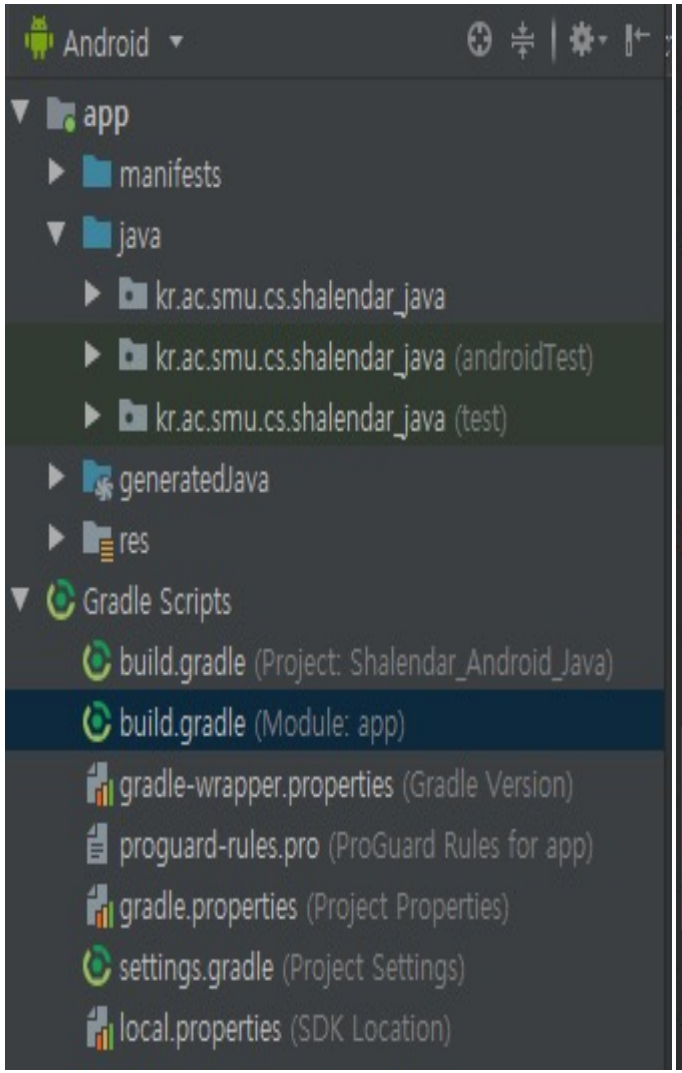
2. '공유' 목적의 기능

- 공유 달력 생성
- **공유 달력으로 다른 사용자 초대**
- Push 알림 서비스
- 공유 달력 내 일정 작성
- ★ **팀원 모두가 가능한 시간대를 파악해 주는 추천 시간 제공**
- 작성한 일정 검색
- 공유 달력 내 게시판 형식의 화면 제공
- **특정 일정에서 멤버들 간 댓글 작성**

2.2 추가 기능 구현 및 라이브러리 적용

#. Gradle Scripts → build.gradle(Module: app)에 작성

#. 라이브러리 import (안드로이드 버전 & 라이브러리 버전 확인 필수!)




```
//add ExpandableTextView Library
implementation 'com.ms-square:expandableTextView:0.1.4'

//materialCalendarView
implementation 'com.prolificinteractive:material-calendarview:1.4.3'

//JSON
//implementation files('libs/json-simple-1.1.1.jar')

//Ion Library
implementation 'com.koushikdutta.ion:ion:2.+'

//Firbase 추가
  implementation 'com.google.firebase:firebase-core:17.0.0'

//Firebase FCM추가
  implementation 'com.google.firebase:firebase-messaging:19.0.1'
implementation 'com.google.firebase:firebase-messaging:17.3.4'

// 원 이미지 추가
implementation 'de.hdodenhof:circleimageview:2.2.0'
```

2.3 서버 구축 및 Android와 연동

<안드로이드 파트>

1. 통신 라이브러리 설정

2. 서버의 host주소 설정

3. 서버와 data형식 맞추기

4. 통신
(JSON 또는 Json객체 사용)

5. 각 요청에서 응답 받은 data처리

<서버 파트>

2.3 서버 구축 및 Android와 연동 2.3.1 통신 라이브러리 설정

1. 서버 구축 후 서버 팀에서 Postman으로 test통신 수행

2. 통신 라이브러리 선정 및 설정

- DefaultHttpClient
- HttpUrlConnection
- Volley
- OkHttp
- Retrofit2
- Ion

#. Ion 통신 라이브러리

Android Asynchronous Networking and Image Loading

3. 안드로이드 코드에서 test 통신 수행

참고(안드로이드 통신 라이브러리 종류)

https://pluu.github.io/blog/android/2016/12/25/android-network/

```
//add ExpandableTextView Library
implementation 'com.ms-square:expandableTextView:0.1.4'

//materialCalendarView
implementation 'com.prolificinteractive:material-calendarview:1.4.3'

//JSON
//implementation files('libs/json-simple-1.1.1.jar')

//Ion Library
implementation 'com.koushikdutta.ion:ion:2.+'

//Firbase 추가
  implementation 'com.google.firebase:firebase-core:17.0.0'

//Firebase FCM추가
  implementation 'com.google.firebase:firebase-messaging:19.0.1'
implementation 'com.google.firebase:firebase-messaging:17.3.4'

// 원 이미지 추가
implementation 'de.hdodenhof:circleimageview:2.2.0'
```

2.3 서버 구축 및 Android와 연동 2.3.2 서버host 주소 설정

#. 서버의 host주소 필요 → "기본 URL" + "/각 요청에 따른 상세 URL"

```java
package kr.ac.smu.cs.shalendar_java;
/*
  서버 URL이 매일 바뀌니
  여기서 toServerUrl 멤버변수만 바꾸면 됨
  각 Activity에서 통신 할 때는 getServerUrl()호출하여 해당 경로 더 붙여줘야

  ex)
  CreateMember(회원 가입) Activity에서는
  통신 URL :: getServerUrl() + /signup
 */
public class NetWorkUrl {

    private String toServer_URL = "https://91dbe0f6.ngrok.io/MIND";

    public String getServerUrl() { return this.toServer_URL; }
}
```

<서버의 "기본 URL주소" 정보 담는 Class>

```java
//응답 바디 설정.
JsonObject json = new JsonObject();

//응답 바디 서버에 보낼 data 넣음
json.addProperty( property: "id", userEmail);
json.addProperty( property: "pw", userPassword);
json.addProperty( property: "deviceToken", deviceToken);
Ion.with(getApplicationContext())
        .load( method: "POST",  url: url.getServerUrl() + "/signin")
        .setHeader( name: "Content-Type",  value: "application/json")
        .progressDialog(progressDialog)
        .setJsonObjectBody(json)
        .asJsonObject()
        .setCallback((e, result) → {
```

<LoginActivity에서 "/signin" 요청 >

2.3 서버 구축 및 Android와 연동 2.3.3 API문서(서버와 Android 파트의 약속)

일정 관련 API

gooiljung edited this page on 10 Sep · 12 revisions

3. 일정 전체 검색

메소드	경로	API
POST	/showAllSche	일정 전체 검색

요청 헤더

```
Content-Type: application/json
```

통신 간 주고 받을 data 형식 지정

요청 바디

```
{
        "cid" : "해당 달력 id 주면 됩니다."
}
```

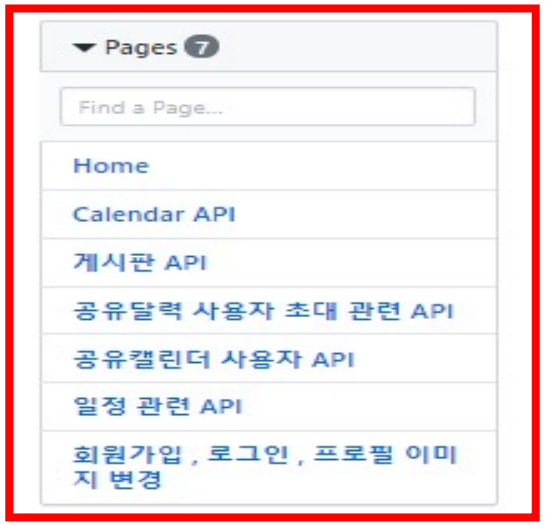
Android → Server

응답 바디

- 일전 전체 검색 성공

Server → Android

```
{
    "data": [
        {
            "id": "jacob456@hanmail.net",
            "cid": 3,
            "sid": 2,
            "title": "교수님 미팅",
            "sContent": "2학기 졸프 준비 시작 및 어플 점검",
            "startDate": "2019-09-09 18:00:00.0",
            "endDate": "2019-09-09 19:00:00.0",
            "area": "상명대학교 제1공학관 G205",
            "img_url": "https://shalendarmind.s3.ap-northeast-2.amazonaws.com/profileImage/20
            "userName": "고진권",
```



2.3 서버 구축 및 Android와 연동 2.3.3 API문서(서버와 Android 파트의 약속)

2.로그인

메소드	경로	API
POST	/signin	로그인

요청 헤더

```
Content-Type: application/json
```

요청 바디

```
{
        "id":"ko@",
        "pw":"11111",
        "deviceToken":"deviceToken 값"
}
```

응답 바디

- 로그인 성공

```
{
    "img_url": null,
    "message": "login success",
    "userName": "고진권",
    "token": "eyJ0eXAiOiJKV1QiLCJyZWdEYXRlIjoxN
}
```

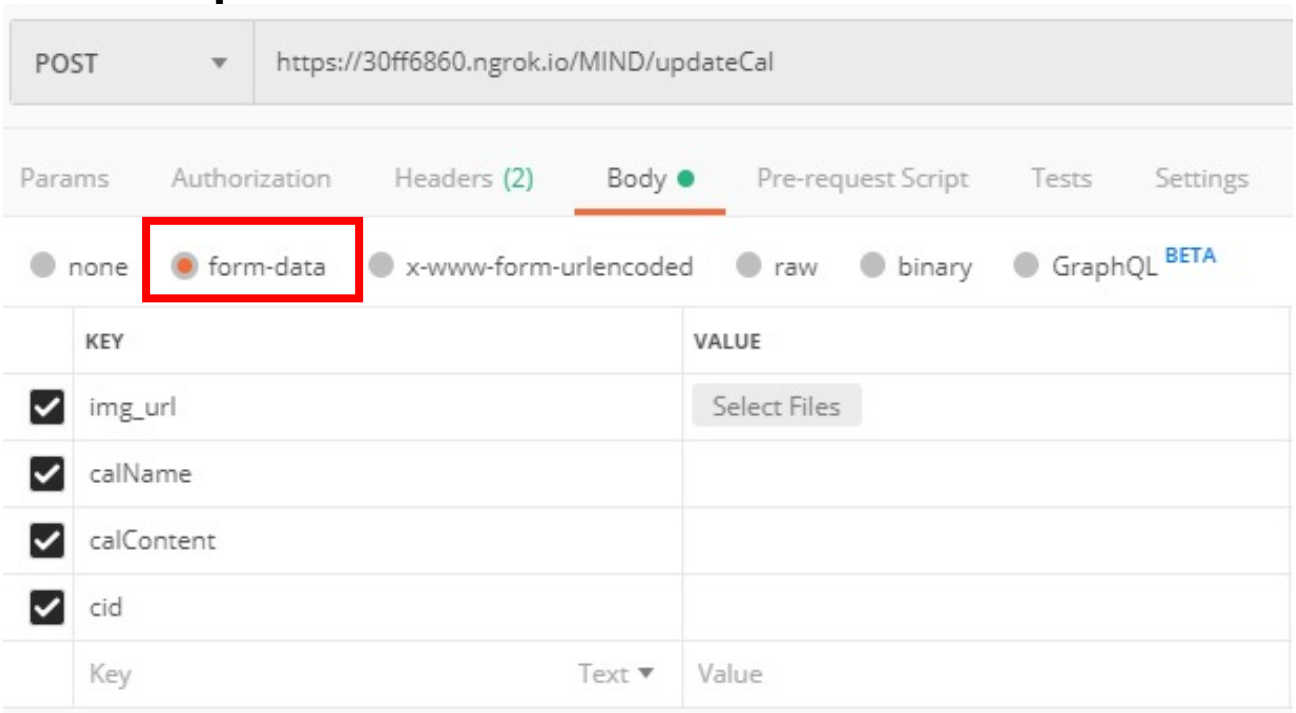
#. String, int등의 data 전송

```
//요청 바디 설정.
JsonObject json = new JsonObject();

//요청 바디 서버에 보낼 data 넣음
json.addProperty( property: "id", userEmail);
json.addProperty( property: "pw", userPassword);
json.addProperty( property: "deviceToken", deviceToken);
Ion.with(getApplicationContext())
        //상세 url설정
        .load( method: "POST", url: url.getServerUrl() + "/signin")
        //요청 헤더 설정
        .setHeader( name: "Content-Type", value: "application/json")
        //로딩 창 설정
        .progressDialog(progressDialog)
        //요청 바디에 JsonObject넣기
        .setJsonObjectBody(json)
        //서버로 부터 응답받을 형식 설정
        .asJsonObject()
        //서버로 요청 후 응답 받아오기
        .setCallback(new FutureCallback<JsonObject>() {
            @Override
            public void onCompleted(Exception e, JsonObject result) {

                if( e!= null) {
                    Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH_LONG).show();
                }
                else {
                    progressDialog.dismiss();
                    String message = result.get("message").getAsString();
                    Toast.makeText(getApplicationContext(), message, Toast.LENGTH_LONG).show();
                    parseFromServer(message, result);
                }
            }
        });
```

2.3 서버 구축 및 Android와 연동 2.3.3 API문서(서버와 Android 파트의 약속)

#. 이미지 파일과 같은 파일 형식은 어떻게 서버로 전송..??

1. Multipart Form data형식 사용




```
final Future ion = Ion.with(getApplicationContext())
        .load( method: "POST",  url: url.getServerUrl() + "/updateCal")
        //요청 헤더 지정
        //.setHeader("Content-Type","application/json")
        .setHeader( name: "Authorization", userToken)
        .setTimeout(60000)
        .setMultipartFile( name: "file", file)
        .setMultipartParameter( name: "calName", calName)
        .setMultipartParameter( name: "calContent", aboutCal)
        .setMultipartParameter( name: "cid", cid_update)
        //응답형식
        .asJsonObject()
```

2. Multipart Form data
- 이미지 파일
- 이미지 파일, String, int 등의 data 동시에 전송 가능

2.3 서버 구축 및 Android와 연동 2.3.4 서버로부터 받은 응답 data처리

#. 서버로 부터 받은 응답data를 Parsing 해야 한다. (Json형식으로 wrap되어 있는 data 꺼내야)

1. JsonObject안에서 parsing을 한번만

```
{
    "img_url": null,
    "message": "login success",
    "userName": "고진권",
    "token": "eyJ0eXAiOiJKV1QiLCJyZWdEYXRlIjoxN
}
```

2. JsonObject안에서 parsing을 여러 번

- **JsonObject안에 JsonArray, JsonObject가 섞여 있음.**
  → 반복문 등을 이용하여 data 파싱

3. JsonNull

- **서버로 부터 받은 응답data에 null이 있을 경우 사용.**

```
{
    "sharePeopleNum": 2,
    "shareUserData": [
        {
            "id": "213",
            "pw": null,
            "userName": "ab",
            "img_url": "bb"
        },
        {
            "id": "1234",
            "pw": null,
            "userName": "1234",
            "img_url": null
        },

    ],
    "calendarData": {
        "id": null,
        "cid": null,
        "calName": "졸업프로젝트",
        "calContent": "쉘린더 프로젝트",
        "userCount": 0,
        "img_url": "exampleImage.jpg"
    },
    "scheduleData": [
        {
            "id": null,
            "cid": 2,
            "sid": 18,
            "title": "graduation@@",
            "sContent": "dddddd",
```

2.3 서버 구축 및 Android와 연동 2.3.4 서버로부터 받은 응답 data처리

2.로그인

메소드	경로	API
POST	/signin	로그인

요청 헤더

```
Content-Type: application/json
```

요청 바디

```
{
        "id":"ko@",
        "pw":"11111",
        "deviceToken":"deviceToken 값"
}
```

응답 바디

- 로그인 성공

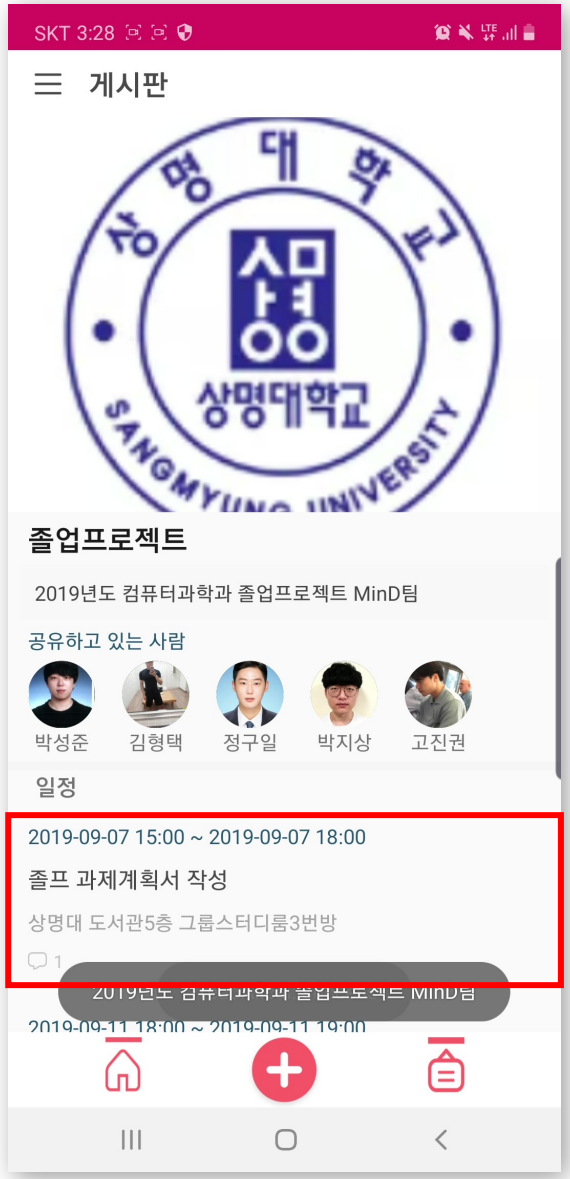
```
{
    "img_url": null,
    "message": "login success",
    "userName": "고진권",
    "token": "eyJ0eXAiOiJKV1QiLCJyZWdEYXRlIjoxNTY
}
```

```
//서버 응답 처리
//응답으로 받은 userToken, getSharedPreference에 저장.
public void parseFromServer(String message, JsonObject result) {
    if(message.equals("login success")) {


        if(result.get("img_url").isJsonNull())
            img_url = "DEFAULT :: profile_IMAGE";
        else
            img_url = result.get("img_url").getAsString();

        userName = result.get("userName").getAsString();
        userToken = result.get("token").getAsString();

        Log.i( tag: "로그인시 받은 이미지 URL", img_url);

        SharedPreferences pref = getSharedPreferences( name: "pref_USERTOKEN", Context.MODE_PRIVATE);
        SharedPreferences.Editor editor = pref.edit();


        editor.putString("userToken", userToken);
        editor.putString("userName", userName);
        editor.putString("userEmail", userEmail);
        editor.putString("img_url", img_url);
        editor.apply();

        Intent intent = new Intent(getApplicationContext(), MainActivity.class);
        startActivityForResult(intent, CodeNumber.TO_MAIN_ACTIVITY);
    }
```

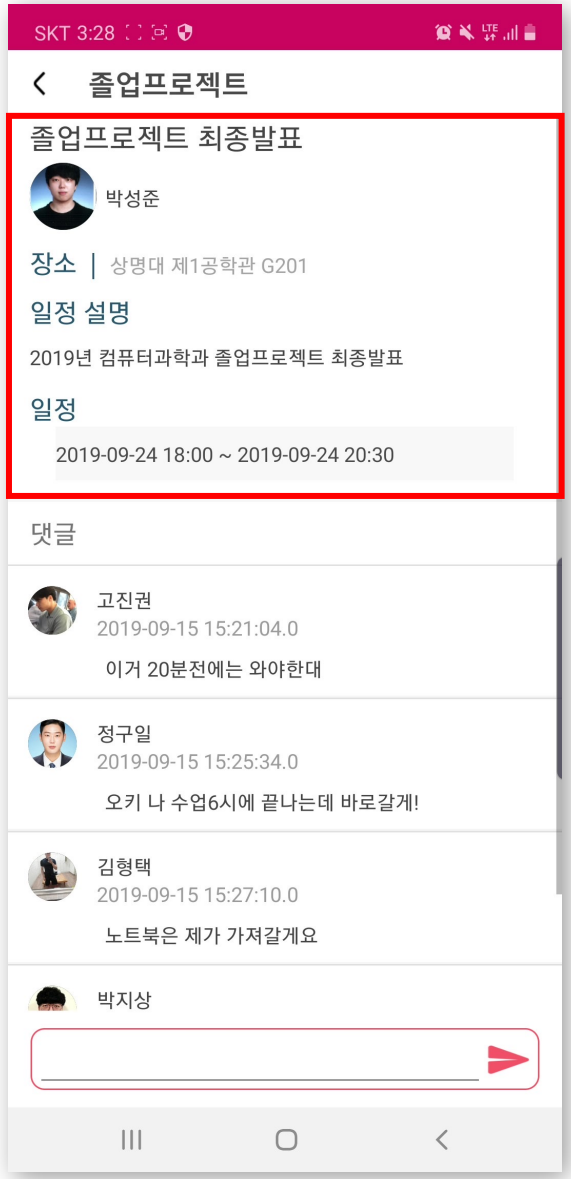
2.3 서버 구축 및 Android와 연동 2.3.4 서버로부터 받은 응답 data처리

#. 서로 다른 Activity에서 동일한 data 처리

Intent사용 VS 서버에 한번 더 요청.
(data load의 속도 차이가 있음)



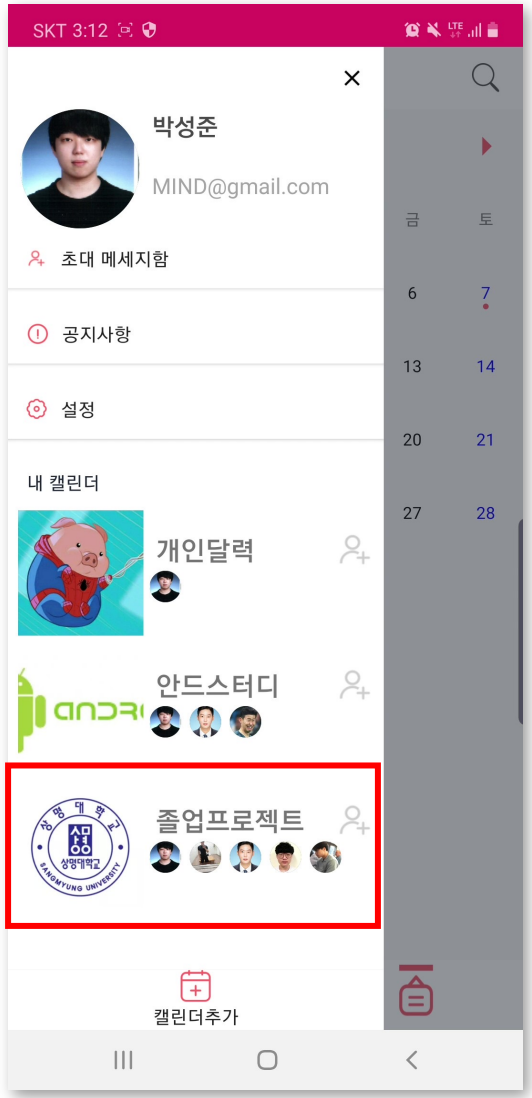
<게시판 Activity>



<일정 상세 Activity>

2.3 서버 구축 및 Android와 연동

2.3.4 서버로부터 받은 응답 data처리



<SideBar Activity>

<Main Activity>

- SideBar에서 요청 URL :: /readAllCal
- Main에서 요청 URL :: /showAllSched

서버에서 응답 data를 받아오는데 시간이 소요

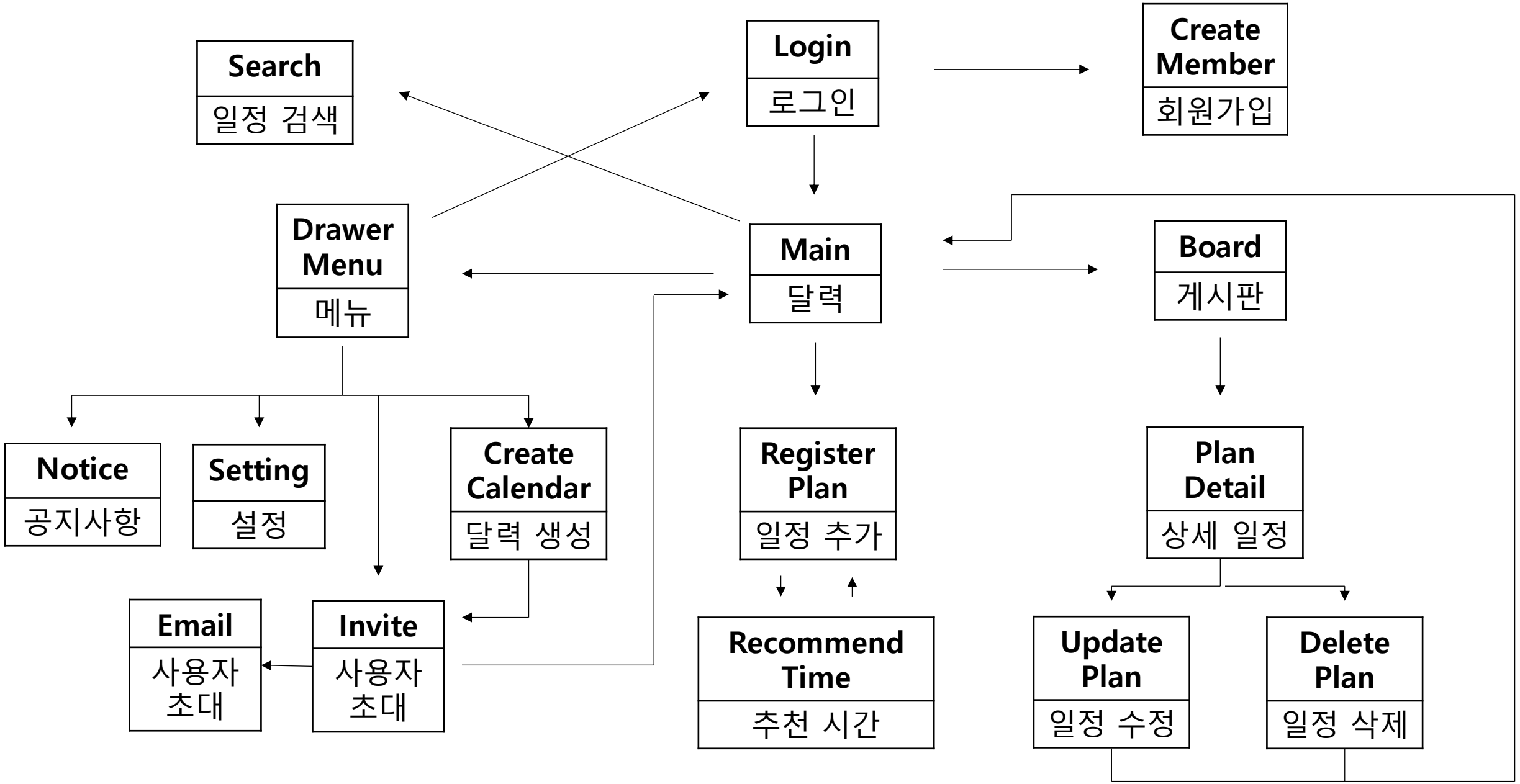
- Background방식
→ 서버로 부터 응답data가 모두 전송되지 않을 수 있음.

- 이러한 상황에서 다른 Activity로 이동하는 경우,
  응답 data전송이 안될 가능성 존재.

- Ion에서는 해당 요청의 응답data를 모두 받을 때까지
  해당 thread를 멈추게 메소드 존재.

2.4 Activity Stack 관리

#. 시중에 나와있는 어플 중 하나 선정 후 흐름 파악



3. Android 개발 역할 분배

고진권

박지상

박성준

1단계	Android Studio 개발 환경 통일 및 github 생성		
2단계	필요한 Activity정의 및 Activity초안 연결		
3단계	Activity 난이도 별 역할 분배 및 구현		
4단계	통신 담당	디자인 담당	View 담당
5단계	코드 수정		

프로젝트를 진행하면서...

1. API문서의 변경.
   - 예상 했던 부분과는 다르게 data형식이 바뀌는 경우.
   - 추가 API를 생성.,

2. 서버 파트, 안드로이드 파트 기능 구현의 분배
   - ex) 일정이 2일이상 걸쳐 있는 경우.
     서버에 저장된 날짜는 String형식. → 사이의 날짜는 생략됨 → 안드로이드에서 사이 날짜 생성.

3. 예상치 못한 상황의 발생
   - 서버의 과부화 및 다운
   - 노트북의 고장
   - 어제만 해도 잘 돌아갔던 코드가 갑자기 실행이 안되거나 오류가 발생.

4. 개발 기간의 지연

5. 구글링
   - 분명 똑같이 구현 했는데 나한테는 실행되지 않거나 오류가 발생.

추가) Shalendar에서 사용한 라이브러리

1. Ion 라이브러리(통신)
- 통신 구현 코드의 길이가 적당함.(이미지 통신에 적합하지 않나 생각)

https://github.com/koush/ion

2. Material Calendar라이브러리(달력 뷰 제공)
- 달력 내 data처리는 적절한 함수 사용하여 직접 구현해야

https://github.com/prolificinteractive/material-calendarview

3. Expandable TextView라이브러리(글자 내용이 펼쳐지도록)
- 일반 text내용이 많을 때 일부분만 보여주고 터치하면 모든 내용을 볼 수 있도록.

https://github.com/Manabu-GT/ExpandableTextView

4. Google Firebase – FCM(Firebase cloud Messaging)
- push알림 서비스 구현(Foreground, Background)

https://firebase.google.com/

감사합니다.