

Open Calender

최종 보고서

6 월 13 일

팀장: 장준규

팀원: 박우석, 송영기, 임규진



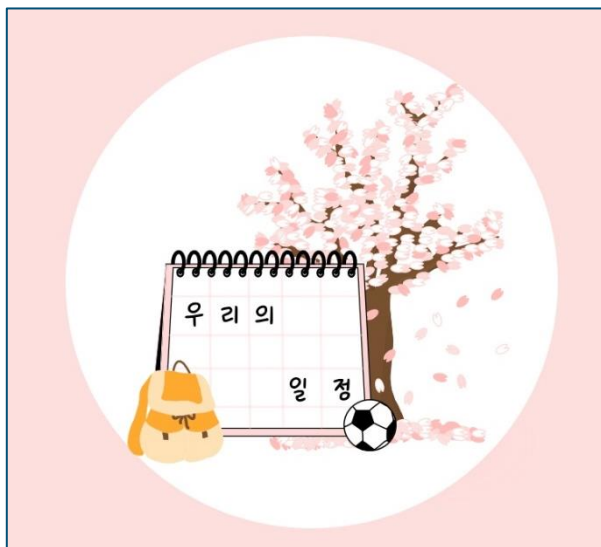
목차

| | |
|---------------------------------|----|
| I. 프로젝트 개요..... | 1 |
| II. 사용자 가이드..... | 2 |
| A) 제공 기능 상세 설명 | 2 |
| B) 설치 및 활용 방법 | 10 |
| III. 개발자 가이드..... | 14 |
| A) 시스템 구성도 및 아키텍처 | 14 |
| B) 주요 활용 기술 | 15 |
| C) 주요 알고리즘 및 데이터 설명 | 18 |
| D) 소스 코드 구조(GitHub 링크 포함) | 20 |
| E) 빌드 및 테스트 방법..... | 29 |
| IV. 테스트 결과 및 분석 | 30 |
| V. 문제점 및 개선 방안 | 32 |
| VI. 버그 리포트 | 34 |
| VII. 프로젝트 후기..... | 35 |
| VIII. 용어 해설 및 참고 문헌 | 36 |

I. 프로젝트 개요

현대 사회의 인류는 데이터 홍수 속에서 살아가고 있습니다. 그리고 사람과 사람, 그룹과 그룹끼리 많은 약속과 만남을 가지고 있습니다. 이 속에서 이뤄지는 많은 일정과 약속을 효율적으로 관리하기 위해 저희 조는 일정을 그룹원들과 공유할 수 있는 “우리의 일정: 우정” 공유 캘린더를 제작하고자 했고 그에 따른 부가 기능으로 로그인, 일정 공유, 게시판 기능 등을 제작했습니다.

각 기능을 온라인 서버 상에서 효율적으로 작동시키기 위해 NHN toast 서버에 DB 형태로 모든 데이터를 저장했고, 안드로이드 기기와 MySQL의 통신을 위해 php를 사용했습니다. 이를 통해 회원가입 기능, 로그인 시 회원 정보 받아 오기, 사용자 일정 로드, 게시판 기능 등을 구현했습니다.



우리의 일정 메인 로고



우리의 일정 어플리케이션 로고

II. 사용자 가이드

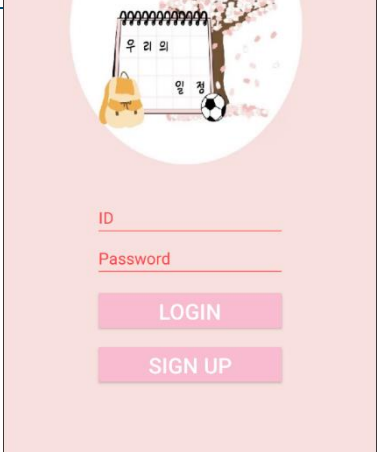
A) 제공 기능 상세 설명

- 로그인 & 회원가입

어플리케이션을 켜올 때 가장 먼저 나오는 화면입니다. 로그인 버튼과 회원가입 버튼으로 다음 화면으로 전환이 가능합니다.

로그인에서는 정보가 입력되지 않으면 정보를 입력해달라는 메시지가 출력됩니다. 또한, 아이디나 비밀번호가 입력될 경우 개인정보를 저장하는 데이터베이스로 입력된 정보를 이동시키고, 아이디를 먼저 대조합니다. 입력한 아이디와 동일한 아이디가 없거나, 동일한 아이디는 있으나 비밀번호가 다른 경우에 입력된 정보에 오류가 있다는 메시지를 출력합니다.

```
20 public class LoginActivity extends AppCompatActivity {
21     private EditText login_ID, login_PW;
22     private Button btn_login, btn_register;
23
24     @Override
25     protected void onCreate(Bundle savedInstanceState) {
26         super.onCreate(savedInstanceState);
27         setContentView(R.layout.activity_login);
28         NetworkUtil.setNetworkPolicy();
29
30         login_ID = (EditText)findViewById(R.id.login_etID);
31         login_PW = (EditText)findViewById(R.id.login_etPW);
32         btn_login = (Button)findViewById(R.id.login_btLogin);
33         btn_register = (Button)findViewById(R.id.login_btRegister);
34
35         btn_login.setOnClickListener(new View.OnClickListener() {
36             @Override
37             public void onClick(View v) {
38                 try {
39                     if(login_ID.length() == 0 || login_PW.length() == 0) {
40                         Toast.makeText(getApplicationContext(), text: "빈칸을 채워주세요.", Toast.LENGTH_SHORT).show();
41                         return;
42                     }
43                     PHPRequest request = new PHPRequest( url: "http://133.186.229.67/Calander_Login.php");
44                     String result = request.PhpLogin(String.valueOf(login_ID.getText()), String.valueOf(login_PW.getText()));
45
46                     PHPRequest requestGetName = new PHPRequest( url: "http://133.186.229.67/Calender_Getname.php");
47                     String Myname = requestGetName.PhpGetname(String.valueOf(login_ID.getText()));
```



The mockup shows a light pink background with a circular logo at the top center. The logo contains a calendar, a tree with pink blossoms, a yellow bag, and a soccer ball. Below the logo, there are two input fields: 'ID' and 'Password', each with a red underline. At the bottom, there are two pink buttons: 'LOGIN' and 'SIGN UP'.

```

49 JSONArray jarray = new JSONArray(Myname);
50
51 JSONObject jobject = jarray.getJSONObject( index: 0);
52 String name = jobject.getString( name: "Name");
53
54 if(result.equals("ok")){
55     Toast.makeText(getApplicationContext(), text: "로그인 성공", Toast.LENGTH_SHORT).show();
56     Intent intent = new Intent( packageContext: LoginActivity.this, MainActivity.class);
57     intent.putExtra( name: "userName",name); //05240951 추가
58     // Log.d("phpgetname", Myname);
59     startActivity(intent);
60 }
61 else{
62     Toast.makeText(getApplicationContext(), text: "입력정보 오류!", Toast.LENGTH_SHORT).show();
63 }
64 }catch (MalformedURLException | JSONException e) {
65     e.printStackTrace();
66 }
67
68 }
69
70 );
71 btn_register.setOnClickListener(new View.OnClickListener() {
72     @Override
73     public void onClick(View v) {
74         Toast.makeText(getApplicationContext(), text: "회원가입 화면으로 이동합니다", Toast.LENGTH_SHORT).show();
75         Intent intent = new Intent( packageContext: LoginActivity.this, RegisterActivity.class);
76         startActivity(intent);
77     }
78 });

```

입력한 정보를 나중에도 사용하기 위해서는 서버로 입력된 데이터를 전송해야 합니다. 이를 해결하기 위해 php 파일을 이용해 안드로이드 스튜디오와 서버를 연결하는데, 안드로이드 스튜디오에서 Internet permission 을 한 뒤, 해당 php 파일의 주소를 입력해줍니다. 그 뒤, 입력한 정보가 php 파일 내의 형식에 의해 저장이 되고 그 정보들이 서버 테이블에 저장이 됩니다.

```

77 public String PhPreRegister(final String userID, final String userPW, final String userName, final String userAge) {
78     try {
79         String postData = "userID=" + userID + "&" + "userPW=" + userPW + "&" + "userName=" + userName + "&" + "userAge=" + userAge;
80         HttpURLConnection conn = (HttpURLConnection)url.openConnection();
81         conn.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
82         conn.setRequestMethod("POST");
83         conn.setConnectTimeout(5000);
84         conn.setDoOutput(true);
85         conn.setDoInput(true);
86         OutputStream outputStream = conn.getOutputStream();
87         outputStream.write(postData.getBytes( charsetName: "UTF-8"));
88         outputStream.flush();
89         outputStream.close();
90         String result = readStream(conn.getInputStream());
91         conn.disconnect();
92         return result;
93     }
94     catch (Exception e) {
95         Log.i( tag: "PHPRequest", msg: "request was failed.");
96         return null;
97     }
98 }

```

해당 코드는 입력된 정보를 php 파일로 보내는 함수 중 일부입니다.

```
Terminal
File Edit View Search Terminal Help
<?php
$userID = $_POST['userID'];
$userPW = $_POST['userPW'];
header("Content-Type: text/html; charset=UTF-8");
$conn = mysqli_connect("127.0.0.1", "root", "swcu1234", "php");
$query = "select ID from users where ID='$userID' and PW='$userPW'";
$result = mysqli_query($conn, $query);

if (mysqli_num_rows($result) > 0)
    echo "ok";
else
    echo "fail";

mysqli_close($conn);
?>
```

```
Terminal
File Edit View Search Terminal Help
?>?php
header("Content-Type: text/html; charset=UTF-8");
$conn = mysqli_connect("127.0.0.1", "root", "swcu1234", "php");
$data_stream = "''.$_POST['userID'].','.$_POST['userPW'].','.$_POST['userName'].','.$_POST['userAge'].''";
$query = "insert into users (ID,PW,Name,Age) values (".$data_stream.")";
$result = mysqli_query($conn, $query);

if($result)
    echo "Input OK";
else
    echo "Input Wrong";

mysqli_close($conn);
?>
```

1,1 All

위 사진은 Register.php 와 Login.php 파일들입니다.

회원가입 버튼을 누르면 아이디, 비밀번호, 이름, 나이를 입력하는 창이 나오고 하나의 정보라도 입력되지 않으면, 다시 입력해달라는 토스트 메시지가 출력됩니다. 여기에 입력된 정보는 위에서 언급한 php 파일로 정보를 이동시키는 코드를 통해 데이터베이스에 저장이 되고 첫 화면인 로그인 화면에서 입력된 정보와 대조되게 됩니다.

```

16 public class RegisterActivity extends AppCompatActivity {
17     private EditText reg_ID, reg_PW, reg_Name, reg_Age;
18     // String으로 받을 경우 회색 글씨
19     private Button btn_register;
20     private String reg1_ID;
21
22     @Override
23     protected void onCreate(Bundle savedInstanceState) {
24         super.onCreate(savedInstanceState);
25         setContentView(R.layout.activity_register);
26         NetworkUtil.setNetworkPolicy();
27
28         reg_ID = (EditText)findViewById(R.id.register_etID);
29         // reg_ID = ((EditText)findViewById(R.id.register_etID)).getText().toString();
30         reg_PW = (EditText)findViewById(R.id.register_etPW);
31         reg_Name = (EditText)findViewById(R.id.register_etName);
32         reg_Age = (EditText)findViewById(R.id.register_etAge);
33         btn_register = (Button)findViewById(R.id.register_btRegister);
34
35         btn_register.setOnClickListener(new View.OnClickListener() {
36             @Override
37             public void onClick(View v) {
38                 try {
39                     if(reg_ID.length() == 0 || reg_PW.length() == 0 || reg_Name.length() == 0 || reg_Age.length() == 0) {
40                         // String일 경우 .trim() 바로 가능
41                         Toast.makeText(getApplicationContext(), text: "빈칸을 채워주세요.", Toast.LENGTH_SHORT).show();
42                         return;
43                     }
44                     else {
45                         PHPRequest request = new PHPRequest( url: "http://133.186.229.67/Calander_Register.php");
46                         String result = request.PhPregister(String.valueOf(reg_ID.getText()), String.valueOf(reg_PW.getText()),
47                             String.valueOf(reg_Name.getText()), String.valueOf(reg_Age.getText()));
48                         if (result.equals("Input OK")) {
49                             Toast.makeText(getApplicationContext(), text: "회원가입 완료", Toast.LENGTH_SHORT).show();
50                             Intent intent = new Intent( packageContext: RegisterActivity.this, LoginActivity.class);
51                             startActivity(intent);
52                         } else {
53                             Toast.makeText(getApplicationContext(), text: "사용 불가능한 아이디입니다.", Toast.LENGTH_SHORT).show();
54                         }
55                     }
56                 } catch (MalformedURLException e){
57                     e.printStackTrace();
58                 }
59             }
60         });
61     }
62 }

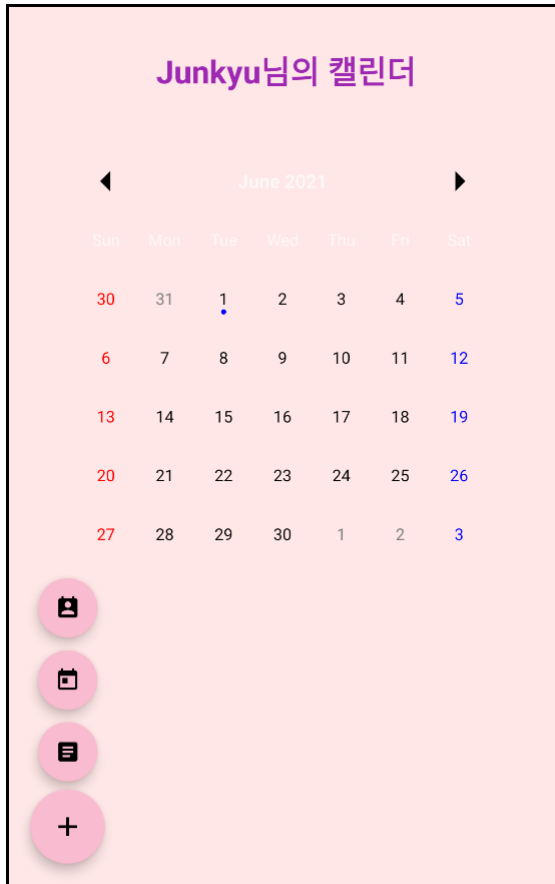
```

- 퍼스널 캘린더 & 오픈 캘린더

메인 화면인 달력 화면입니다. 로그인 시에 회원 정보에서 입력한 정보를 바탕으로 '@@님의 캘린더' 형식의 문구와 함께 달력이 출력됩니다. 오늘의 날짜는 파란색, 일정이 있는 날은 빨간색 점이 보이게 됩니다.



일정이 없는 날을 터치했을 때 일정을 추가할 수 있는 입력 창이 나오는 것이 왼쪽 사진이고, 오른쪽 사진은 등록 되어있는 일정을 수정하거나 삭제를 할 수 있는 화면입니다.



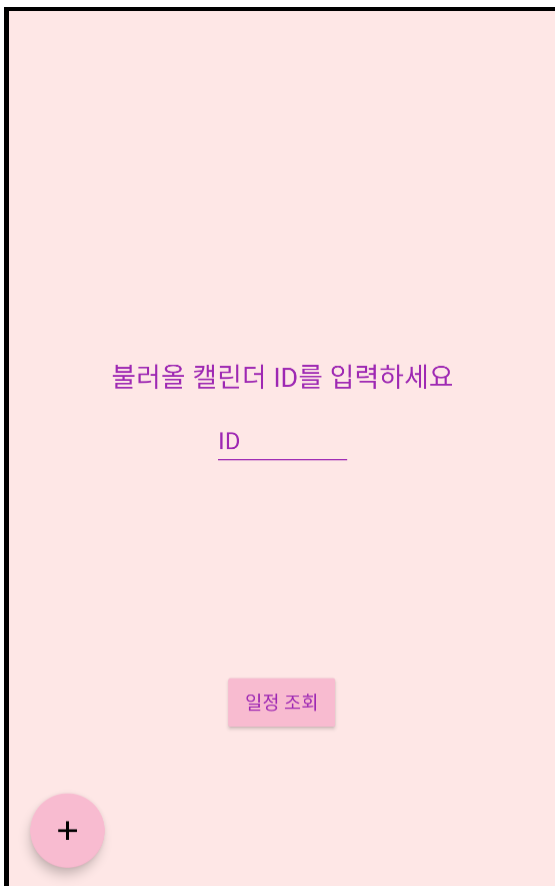
또한, 좌측 하단의 +모양의 플로팅 버튼을 누르게 되면 다른 화면으로 갈 수 있는 3 개의 버튼이 나옵니다.

첫번째 버튼은 잠시 후 설명드릴 게시판으로 가는 버튼.

두번째 버튼은 현재 보여지는 개인 캘린더 화면으로 가는 버튼.

마지막 세번째 버튼은 일정에 공유 일정을 추가하는 버튼입니다.

플로팅 버튼의 경우에는 첫번째, 두번째, 세번째 버튼을 통해 이동 가능한 모든 화면에 계속 출력됩니다.



특히, 공유 캘린더의 경우 세번째 버튼을 눌러서 아이디를 입력하는 화면으로 이동한 뒤, 일정을 공유하고 싶은 타인의 아이디를 입력하게 되면 그 아이디에 등록되어있는 일정을 나의 캘린더로 불러올 수 있습니다.

- 게시판

- 게시판 리스트

게시판을 구현함에 있어 여러 개의 게시물을 한 번에 불러오는 기능이 필요해 다음과 같이 JSONArray 를 불러오고 각 변수 i 값에 대해 게시물을 받아왔습니다.

```

MovetoPersonal.setOnClickListener(this);
MovetoOpen.setOnClickListener(this);

try {
    PHPRequest request = new PHPRequest( url: "http://133.186.229.67/Calender_NoticeGet.php");
    String result_date = request.PHPNoticeGet();
    JSONArray Noticearray = new JSONArray(result_date);
    Log.d( tag: "pickdot", msg: "Noticearray"+ Noticearray );
    for(int i=0; i<Noticearray.length();i++){
        JSONObject NoticeObject = Noticearray.getJSONObject(i);
        String not_name = NoticeObject.getString( name: "Writer");
        String not_title = NoticeObject.getString( name: "Title");
        String not_date = NoticeObject.getString( name: "Timeline");
        String not_content = NoticeObject.getString( name: "Content");

        list_itemArrayList.add(
            new list_item(R.mipmap.ic_launcher, not_name, not_title, not_date, not_content));

        Log.d( tag: "pickdot", msg: "name : " + not_name);
        Log.d( tag: "pickdot", msg: "title : " + not_title);
        Log.d( tag: "pickdot", msg: "date : " + not_date);
        Log.d( tag: "pickdot", msg: "content : " + not_content);
    }
} catch (MalformedURLException e) {
    e.printStackTrace();
} catch (JSONException e) {
    e.printStackTrace();
}

myListAdapter.notifyDataSetChanged();

FloatingActionButton fab = findViewById(R.id.fab);
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Snackbar.make(view, text: "게시물 작성", Snackbar.LENGTH_LONG)
            .setAction( text: "Action", listener: null).show();
        Intent intent = new Intent( packageContext: Noticeboard.this, UploadNoticeboard.class);
        intent.putExtra( name: "userName",userName); // 0524 추가
    }
}

```

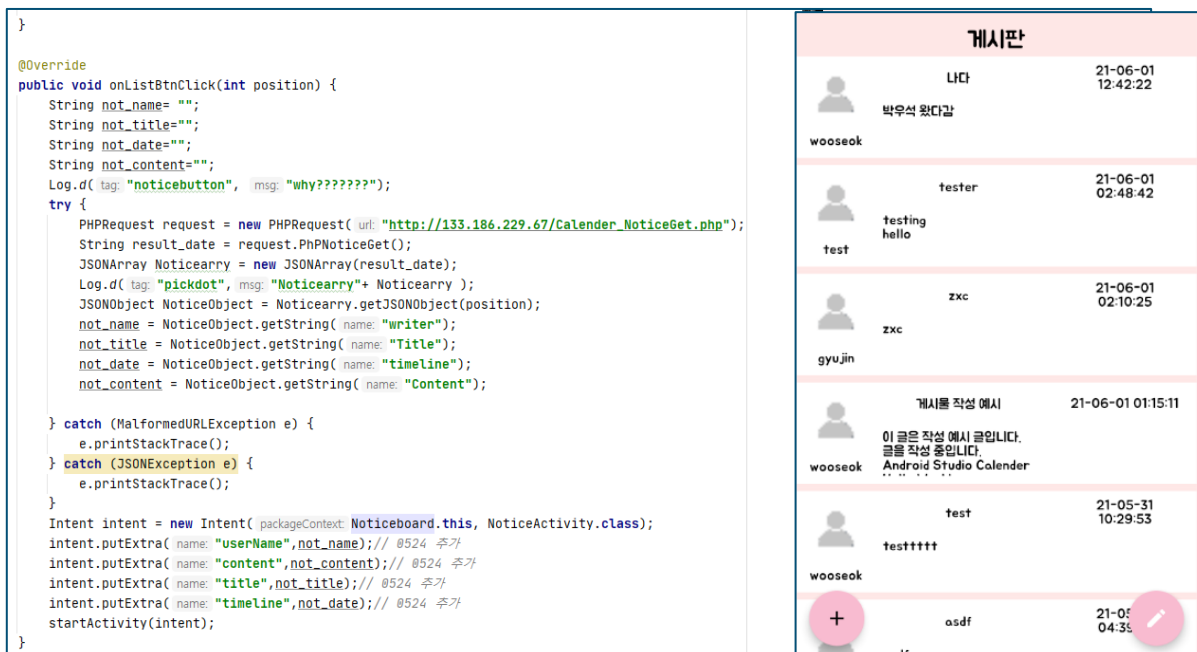
받아온 게시물을 스크롤 가능한 형태로 출력시키기 위해 Adapter, getter&setter 를 사용했고 이는 성공적이었습니다. 각 게시물은 하나의 새 레이아웃으로 여러 개의 레이아웃들이 겹쳐져 위 사진과 같은 형태를 출력시켰습니다.

- 게시물 보기


작성한 게시물의 내용이 길면 다음 그림과 같이 잘려 출력되기에 각 게시물을 클릭하면 큰 화면에서 볼 수 있게 구현했습니다.



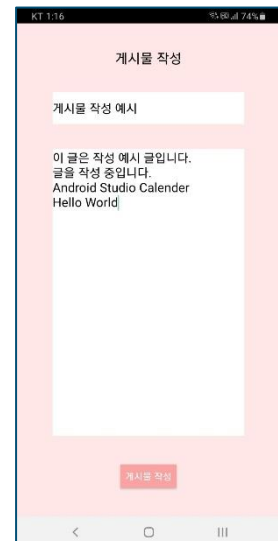
이를 구현할 때 각 게시물 위에 button 객체를 추가했습니다. 하지만 이 버튼이 모두 같은 버튼으로 인식되어 가장 최근 게시글만을 출력하는 오류가 있었으나 해당 함수에 position 인자를 받는 코드를 구성해 성공적으로 모든 게시글을 그때 그때 출력시킬 수 있게 만들었습니다.



- 게시물 작성

게시판 우측하단 게시물 작성 버튼을 클릭하면  새 글 작성 화면으로 넘어갑니다. 오른쪽 사진과 같은 화면에서 제목, 내용을 입력 후 게시물 작성 버튼을 클릭하면 사용자 이름, 현재 시각이 자동적으로 추가돼 php 를 통한 DB 로 입력되게 됩니다.

| <div>←→</div> | | | | list | Content | timeline | writer | Title | |
|--------------------------|--|--|--|-------|---------|---|-------------------|---------|-----------|
| <input type="checkbox"/> | | | | 복사 삭제 | 47 | asdf | 21-05-26 07:51:30 | asd | asdf |
| <input type="checkbox"/> | | | | 복사 삭제 | 50 | dd | 21-05-30 05:00:50 | wooseok | dd |
| <input type="checkbox"/> | | | | 복사 삭제 | 57 | testtttt | 21-05-31 10:29:53 | wooseok | test |
| <input type="checkbox"/> | | | | 복사 삭제 | 58 | shit | 21-05-31 04:28:19 | song | holy |
| <input type="checkbox"/> | | | | 복사 삭제 | 59 | wfwf | 21-05-31 04:38:57 | song | wfw |
| <input type="checkbox"/> | | | | 복사 삭제 | 60 | asdfa | 21-05-31 04:39:41 | song | asdf |
| <input type="checkbox"/> | | | | 복사 삭제 | 61 | zxc | 21-06-01 02:10:25 | gyujin | zxc |
| <input type="checkbox"/> | | | | 복사 삭제 | 62 | 박우석 왔다감 | 21-06-01 12:42:22 | wooseok | 나다 |
| <input type="checkbox"/> | | | | 복사 삭제 | 63 | 이 글은 작성 예시 글입니다. 21-06-01 01:15:11 wooseok 게시글을 작성 예시 글을 작성 중입니다. | | | 게시물 작성 예시 |
| Android Studio Calend... | | | | | | | | | |



게시물 작성 화면 구성을 위해 다음과 같이 코드를 구성했습니다. 제목이나 내용이 없다면 '빈칸을 채워주세요' 토스트 메시지가 출력되고 작성이 완료되면 PHPRequest 로 이동해 게시물을 입력합니다.

B) 설치 및 활용 방법

안드로이드 스튜디오를 활용한 어플리케이션 설치 방법

1. [Git hub](#) 의 Open_Calendar_Junkyu.zip 을 설치 후 압축을 해제합니다.
2. Calendar 어플리케이션 파일을 Android Studio 를 통해 엽니다.
3. 안드로이드 휴대폰을 컴퓨터에 연결한 뒤, 설정 > 휴대전화 정보 > 소프트웨어 정보에서 빌드 번호를 연타해 개발자 옵션을 활성화합니다.

4. 개발자 옵션의 USB 디버깅을 활성화시킵니다.

5. 이후 Android Studio 를 확인하시면 기존에  AVD 로 작동하던 코드가  입력된 휴대전화로 바뀐 것을 확인할 수 있습니다.

6. 재생 버튼을 클릭하면 Calendar 어플리케이션이 설치되었음을 확인할 수 있습니다.

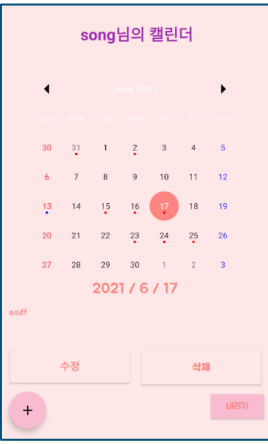
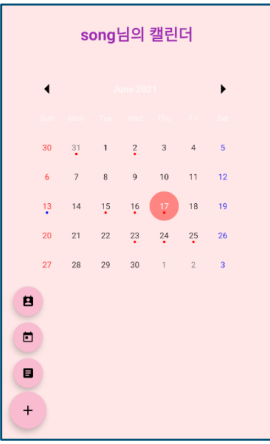

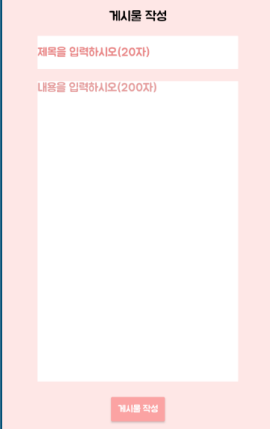




7. 이후 어플리케이션을 실행해 사용하면 됩니다.

B-2)활용 방법

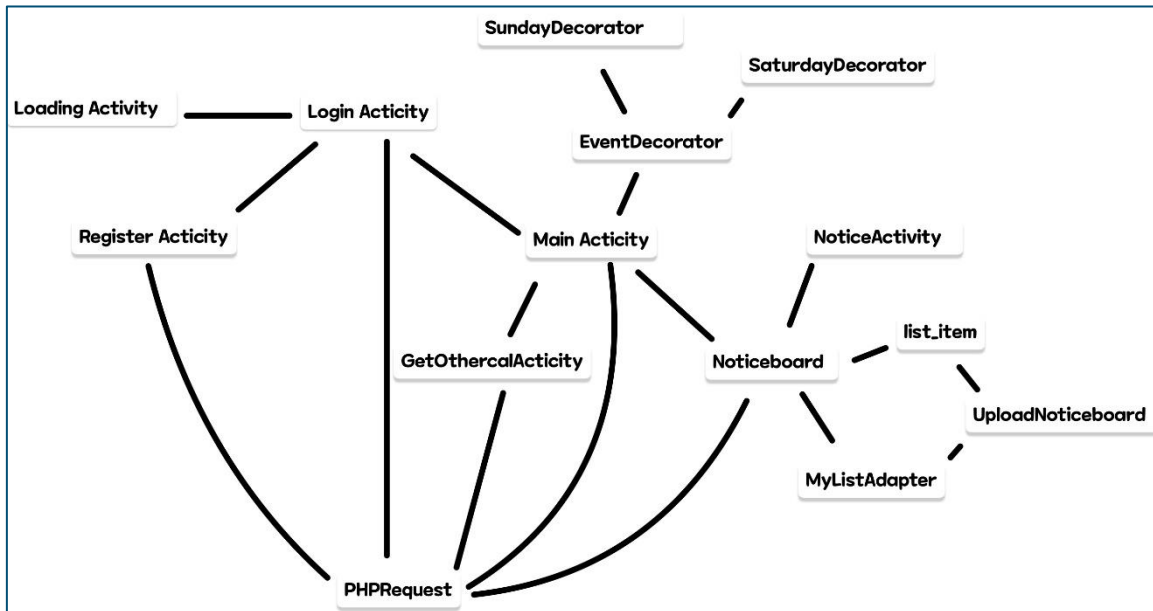
| | | | |
|---|-------------------------------|--|---|
|  | <p>어플리케이션 작동 시 나오는 화면입니다.</p> |  | <p>로그인을 하여 어플에 접속하거나, 회원가입으로 넘어갈 수 있는 화면입니다. 빈칸에 아무것도 입력하지 않으면, 빈칸을 채워 달라는 알림 메시지와 서버에 등록 되어있는 정보와 다른 정보를 입력하면, 잘못된 정보라는 알림 메시지를 출력합니다.</p> |
| <p>로딩 화면</p> | | <p>로그인 화면</p> | |

| | | | |
|---|--|--|--|
|  | <p>회원 가입을 위한 개인 정보 수집 화면입니다. 보안 관련 문제 때문에 직접적인 개인 정보는 받지 않고, 사용할 아이디와 비밀번호 그리고 이름, 나이를 확인합니다. 로그인 화면과 마찬가지로 하나의 정보라도 입력되지 않으면 빈칸을 채워 달라는 알림 메시지, 혹시나 중복된 아이디를 사용할 경우에는 이미 사용 중인 아이디라는 알림 메시지가 출력됩니다.</p> |  | <p>어플리케이션의 메인 화면인 달력 화면입니다. 오늘의 날짜는 파란색, 일정이 있는 날짜는 빨간색으로 포인트를 줘서 구분이 쉽게 만들었습니다. 캘린더 상단에는 회원가입 할 때 입력한 이름과 함께 '~~님의 캘린더' 라는 문구가 출력됩니다.</p> |
| <p>회원가입 화면</p> | | <p>일정 화면</p> | |
|  | <p>일정이 없는 날짜를 터치하면, 일정을 등록할 수 있는 UI가 캘린더 하단에 출력됩니다. 일정을 입력하고 일정 추가 버튼을 누르면 오른쪽의 화면으로 넘어가게 됩니다.</p> |  | <p>추가한 일정의 시간대를 설정할 수 있는 화면이 이어서 출력됩니다. 시간을 정해주면 일정 옆에 예를 들어 9시~11시 이런 형식으로 시간이 함께 출력됩니다.</p> |
| <p>일정 추가</p> | | <p>일정 시간 설정</p> | |

| | | | |
|---|--|--|---|
|  | <p>이미 일정이 있는 날짜를 고르면, 등록되어 있는 일정을 삭제하거나, 수정할 수 있습니다. 내리기 버튼을 누르게 되면 모든 창이 사라집니다.</p> |  | <p>메인 화면 왼쪽 하단에는 플로팅 버튼으로 다른 화면으로 넘어갈 수 있는 버튼이 있습니다. 기존에는 + 버튼 하나만 있다가, 버튼을 누르면 총 3 가지 버튼이 나오게 됩니다. 아래쪽부터 각각 게시판, 메인 캘린더, 타인 일정 추가 버튼입니다.</p> |
| <p>일정 수정, 삭제</p> | | <p>메뉴 확인</p> | |
|  | <p>게시판 버튼을 누르게 되면 다양한 사용자들이 남긴 자유로운 글 들을 볼 수 있습니다.</p> |  | <p>우측 하단 연필 모양의 버튼을 누르면 게시물을 작성할 수 있는 화면이 출력됩니다.</p> |
| <p>게시물 목록</p> | | <p>게시물 작성</p> | |
|  | <p>작성된 게시글 리스트에서 내용이 길면 하단 내용이 잘려서 보이지 않는 현상이 있습니다. 그렇기에, 게시물을 선택하면 하나의 화면에 하나의 게시글이 보이도록 할 수 있습니다. 게시글을 하나 클릭하면, 제목과 작성자 그리고 작성된 시간과 내용을 상세하게 확인할 수 있습니다.</p> |  | <p>플로팅버튼의 맨 위 버튼인 타인 일정 등록 버튼을 누르면 등록하고 싶은 사람의 아이디를 입력할 수 있는 화면으로 넘어가게 되고, 해당 인원의 아이디를 입력하면, 나의 캘린더에 해당 아이디 주인의 일정이 함께 출력되어 일정을 공유할 수 있습니다.</p> |
| <p>게시글 크게 보기</p> | | <p>공유 캘린더 로딩</p> | |

III. 개발자 가이드

A) 시스템 구성도 및 아키텍처



Calendar 어플리케이션의 시스템 구성도는 위와 같습니다.

Loading 을 거쳐 Login Activity 를 통해 사용자 ID, PW 를 확인합니다.

이때 Register 로 접속해 회원가입을 할 수 있고, ID, PW 가 일치하면 Main Activity 로 넘어갑니다.

Main Activity 는 달력의 주말을 꾸미는 Event Decorator, 게시판 레이아웃인 Notice Board, 공유 일정을 가져오는 Get Other Cal Activity 로 이동 가능합니다.

Notice Board 는 각각 Notice 작성, Notice 불러오기가 가능한 Notice Activity 와 Upload Notice Board 와 연결되어 있습니다.

마지막으로 서버와의 통신을 담당하는 PHP Request 코드는 서버 연결이 필요한 Login, Register, Main, Get Other Cal, Notice Board 와 연결되어 원활한 서버 통신을 진행시킵니다.

B) 주요 활용 기술

1. Activity

- Activity 는 일종의 애플리케이션 구성 요소.
- 액티비티 마다 창이 하나씩 주어져 액티비티에 사용자 인터페이스를 끌어올 수 있습니다.
- 액티비티는 stack 구조로 쌓이게 됨(LIFO 구조).
 - : manifest 에서 실행순서 설정 가능합니다.
- 액티비티를 생성하려면 Activity 의 서브클래스(또는 이의 기존 서브클래스)를 생성합니다.
- onCreate()
 - : 시스템은 액티비티를 생성할 때 onCreate()을 호출합니다.
 - : setContentView()를 호출해서 xml 과 연결해 주어야 액티비티의 사용자 인터페이스 레이아웃을 정의할 수 있습니다.

2. intent

- 메시지 객체.
- 앱 구성요소 2 부터 작업을 요청할 수 있습니다.
- 사용은 크게 2 가지로 나뉩니다.
 - 1) 액티비티 시작
 - : 액티비티의 새 인스턴스를 시작하려면 intent 를 startActivity()로 전달합니다.
 - : 이때 putExtra 로 인자 값 전달이 가능하고, getExtra 로 인자 값을 전달 받을 수 있습니다. (key 값으로 구별).
 - 2) 서비스 시작
 - : 서비스를 시작할 때 Intent 를 startService()에 전달합니다.
 - : 서비스가 클라이언트-서버 인터페이스로 디자인된 경우, 다른 구성 요소로부터 서비스에 연결시켜서 Intent 를 bindService()에 전달합니다.

3. phpMyAdmin, php

- php 는 프로그래밍 언어 중 하나입니다.
- 웹서버에서 해석되는 스크립트 언어입니다.
- 거의 모든 운영체제에 구현이 가능합니다.
- PHP 는 서버 측에서 사용되는 언어, 자바 스크립트는 클라이언트 측에서 사용되는 언어입니다.
- phpmyadmin 은 SQL 문을 웹 상에서 관리하기 위해 PHP 로 작성한 오픈소스 도구입니다.

4. SQL

- SQL 은 데이터베이스 관리 시스템 DBMS 에서 데이터를 관리하기 위해 설계된 프로그래밍 언어입니다.
- 데이터 정의 언어
 - CREATE 데이터베이스 개체(테이블, 컬럼)등을 생성
 - DROP 데이터베이스 개체 삭제
 - ALTER 데이터베이스 개체를 수정 또는 정의를 변경
- 데이터 조작 언어
 - INSERT 데이터를 삽입
 - UPDATE 값을 업데이트, 수정
 - DELETE 값을 삭제
 - SELECT 조건에 맞는 값들을 취합
- 데이터 제어 언어
 - GRANT 특정 사용자에게 작업에 대한 권한 부여
 - REVOKE 권한을 다시 박탈함

5. MaterialCalendarView

- 개인이 직접 커스텀을 한 캘린더 뷰입니다. Gradle 에서 dependency 를 새로이 추가해서 넣었습니다.
- 안드로이드 스튜디오에 기본적으로 삽입 되어있는 CalendarView 의 경우는 기능이 매우 제한적이기 때문에 원하는 기능들을 구현하기 쉽지 않습니다. 따라서 커스텀 된 해당 뷰를 사용해 여러가지 기능을 구현할 수 있게 합니다.

6. JSON

- 키와 값이 쌍으로 되어있는 데이터 형식입니다.

```
{ "name2": 50, "name3": "값3", "name1": true }
```

- 위처럼 name2 라는 키가 가리키는 값이 있는 포맷입니다.
- PHP 파일에서 쿼리문을 실행시켜 해당 결과 값을 JSON 형식으로 return 해서 string 형식으로 받아옵니다.
- JSONArray(string)를 사용해서 string 문을 JSON 객체 형식으로 저장합니다.
- JSONArray.JSONObject(value)를 통해서 value 번째 열에 있는 JSON 값들을 가져옵니다.
- JSONArray.JSONObject(value). getString("Key")를 통해 Key 라는 키 값을 가진 value 값을 string 로 받아옵니다.

7. Adapter

- 어댑터는 일정 클래스의 형식을 사용자가 기대하는 형식으로 로 변환시키는 기능입니다.

8. Log

- 로그는 표시로써 설정한 알고리즘이 잘 실행이 되는지 확인할 수 있는 방법입니다.
- Log.d("key","activity1")로 사용할 수 있고 이후에 log 란에 key 를 입력했을 때 만약 해당 라인이 실행이 되었다면 activity1 이 나와야 합니다. 이런 방식으로 프로그램이 잘 짜여 있는지를 확인할 수 있습니다.

C) 주요 알고리즘 및 데이터 설명

본 Calendar 애플리케이션은 데이터를 Linux 서버 상의 PhpMyAdmin 을 활용해 DBMS 로 관리합니다. 각 Table 과 그의 데이터는 다음과 같습니다.

Main

| # | 이름 | 종류 | 데이터정렬방식 | 보기 | Null | 기본값 | 설명 | 추가 |
|--------------------------|------------------|-------------|--------------------|----|------|-----|----|----------------|
| <input type="checkbox"/> | 1 list | int | | | 아니오 | 없음 | | AUTO_INCREMENT |
| <input type="checkbox"/> | 2 ID | varchar(10) | utf8mb4_0900_ai_ci | | 아니오 | 없음 | | |
| <input type="checkbox"/> | 3 Year | varchar(10) | utf8mb4_0900_ai_ci | | 아니오 | 없음 | | |
| <input type="checkbox"/> | 4 Month | varchar(10) | utf8mb4_0900_ai_ci | | 아니오 | 없음 | | |
| <input type="checkbox"/> | 5 Day | varchar(10) | utf8mb4_0900_ai_ci | | 아니오 | 없음 | | |
| <input type="checkbox"/> | 6 start_time | int | | | 예 | 9 | | |
| <input type="checkbox"/> | 7 start_time_min | int | | | 예 | 0 | | |
| <input type="checkbox"/> | 8 end_time | int | | | 예 | 9 | | |
| <input type="checkbox"/> | 9 end_time_min | int | | | 예 | 0 | | |
| <input type="checkbox"/> | 10 Content | varchar(50) | utf8mb4_0900_ai_ci | | 아니오 | 없음 | | |

- 일정을 저장하는 DB 로 날짜, 사용자 ID, 일정 진행 시간을 저장합니다.

SUB

| # | 이름 | 종류 | 데이터정렬방식 | 보기 | Null | 기본값 |
|--------------------------|------------------|-------------|--------------------|----|------|------|
| <input type="checkbox"/> | 1 list | int | | | 예 | NULL |
| <input type="checkbox"/> | 2 ID | varchar(30) | utf8mb4_0900_ai_ci | | 아니오 | 없음 |
| <input type="checkbox"/> | 3 Year | int | | | 아니오 | 없음 |
| <input type="checkbox"/> | 4 Month | int | | | 아니오 | 없음 |
| <input type="checkbox"/> | 5 Day | int | | | 아니오 | 없음 |
| <input type="checkbox"/> | 6 start_time | int | | | 아니오 | 없음 |
| <input type="checkbox"/> | 7 start_time_min | int | | | 아니오 | 없음 |
| <input type="checkbox"/> | 8 end_time | int | | | 아니오 | 없음 |
| <input type="checkbox"/> | 9 end_time_min | int | | | 아니오 | 없음 |
| <input type="checkbox"/> | 10 Content | varchar(30) | utf8mb4_0900_ai_ci | | 아니오 | 없음 |

- 타 이용자의 데이터를 받아올 때 사용하는 DB 로 구성은 Main 과 같으나 다른 사용자 데이터를 저장한다는 점에서 다릅니다.

Noticeboard

| # | 이름 | 종류 | 데이터정렬방식 | 보기 | Null | 기본값 | 설명 | 추가 |
|---|----------|--------------|--------------------|----|------|-----|----------------|----|
| 1 | list | int | | | 아니오 | 없음 | AUTO_INCREMENT | |
| 2 | Content | varchar(200) | utf8mb4_0900_ai_ci | | 아니오 | 없음 | | |
| 3 | timeline | varchar(30) | utf8mb4_0900_ai_ci | | 아니오 | 없음 | | |
| 4 | writer | varchar(30) | utf8mb4_0900_ai_ci | | 아니오 | 없음 | | |
| 5 | Title | varchar(10) | utf8mb4_0900_ai_ci | | 아니오 | 없음 | | |

- 게시판을 위한 DB 로 기본 키인 list, 내용을 저장하는 Content, 작성 일자와 시간을 자동으로 저장하는 timeline, 게시자인 writer, 제목인 Title 을 저장하고 있습니다.

SUB2

| # | 이름 | 종류 | 데이터정렬방식 | 보기 | Null | 기본값 |
|---|---------|-------------|--------------------|----|------|-----|
| 1 | ID | varchar(30) | utf8mb4_0900_ai_ci | | 아니오 | 없음 |
| 2 | Year | int | | | 아니오 | 없음 |
| 3 | Month | int | | | 아니오 | 없음 |
| 4 | Day | int | | | 아니오 | 없음 |
| 5 | Content | varchar(30) | utf8mb4_0900_ai_ci | | 아니오 | 없음 |

- 일정을 저장하기 전 임시 DB 로 사용자, 내용, 날짜를 임시로 저장하고 해당 사항의 중복, 오류 등 문제가 없으면 Main 으로 넘어가는 DB 입니다.

Users

| # | 이름 | 종류 | 데이터정렬방식 | 보기 | Null | 기본값 |
|---|--------|-------------|--------------------|----|------|-----|
| 1 | IDlist | int | | | 아니오 | 없음 |
| 2 | ID | varchar(10) | utf8mb4_0900_ai_ci | | 아니오 | 없음 |
| 3 | PW | varchar(20) | utf8mb4_0900_ai_ci | | 아니오 | 없음 |
| 4 | Name | varchar(10) | utf8mb4_0900_ai_ci | | 아니오 | 없음 |
| 5 | Age | varchar(10) | utf8mb4_0900_ai_ci | | 아니오 | 없음 |

- 사용자 정보를 저장하는 DB 로 Login, Register 시 사용합니다.

D) 소스 코드 구조(GitHub 링크 포함)

코드 GitHub 주소: <https://github.com/Chaessen/our-schedule>

PHPrequest.java

- 서버와 클라이언트 사이 데이터 교환이 주 기술이기 때문에 PHP를 통한 데이터 교환 알고리즘이 주가 되었습니다.
- 클라이언트에서 PHP로 보낼 변수들과 DataStream을 설정을 합니다
- 또한 그에 필요한 옵션, 인코딩 방식, PHP주소, request방식등을 설정을 합니다
- 서버에 있는 PHP파일을 실행시킵니다.
- PHP 파일에는 phpMyAdmin에서 값들을 받거나 올리기 위한 쿼리문들이 작성되어 있습니다.
- 쿼리 문 결과를 리턴하여 다시 클라이언트, 즉 안드로이드 스튜디오로 받아오게 됩니다.
- 이 후 그 결과를 바탕으로 프로그램이 활성화됩니다.

LoginActivity.java

- Login ID, PW를 받는 변수, 로그인, 회원가입 등록 버튼을 가리키는 변수 설정

| 변수 명 | 변수 내용 |
|--------------|----------------|
| login_ID | 로그인 아이디를 받는 변수 |
| login_PW | 로그인 비밀번호 받는 변수 |
| btn_login | 로그인 버튼 |
| btn_register | 회원가입 버튼 |

- login_ID.setOnKeyListener
ID 입력을 확인 후 PW가 입력되었는지 확인합니다.
- login_PW.setOnKeyListener
LOGIN 버튼이 클릭되는지 대기합니다.
- btn_login.setOnKeyListener
PhPLogin(ID,PW)을 통해 Calander_Login.php를 불러와서 아이디, 비밀번호 값을 데이터베이스에 삽입합니다.
PhPgetname(ID)을 통해 Calender_Getname.php를 불러와 아이디에 맞는 Name을 받고 String name 변수에 값을 넣은 뒤 MainActivity로 이동합니다.
- btn_register.setOnKeyListener
RegisterActivity로 넘어갑니다.

RegisterActivity.java

- reg_ID, reg_PW, reg_Name, reg_Age 변수 설정

| 변수 명 | 변수 내용 |
|--------------|-----------------|
| reg_ID | 아이디를 받는 변수 |
| reg_PW | 비밀번호 받는 변수 |
| reg_Name | 닉네임을 받는 변수 |
| reg_Age | 나이를 받는 변수 |
| Btn_register | 회원가입을 하는 버튼 |
| Btn_check | 유저 네임 중복확인하는 버튼 |

- btn_check.setOnClickListener()
PhPgetname(ID)을 통해 findid.php를 불러와서 아이디에 맞는 Name이 있는 지 중복 체크를 합니다.
- Btn_register.setOnClickListener()
PhPregister(ID, PW, Name, Age)을 통해 Calander_Register.php를 불러와서 아이디, 비밀번호, 닉네임, 나이를 서버에 저장하고 LoginActivity로 넘어갑니다.

GetothercalActivity.java

| 변수 명 | 변수 내용 |
|----------------|---------------|
| Btn_getdata | 데이터 받는 버튼 |
| Get_ID | ID 를 받는 변수 |
| isBtnOpen | Open 변수 확인 버튼 |
| OpenBtnList | + 버튼 |
| MovetoBoard | 게시판 이동 버튼 |
| MovetoPersonal | 개인 일정 이동 버튼 |
| MovetoOpen | 공유 일정 이동 버튼 |
| Btn_open | 버튼 활성화 변수 |
| Btn_close | 버튼 비활성화 변수 |

- getIntent()를 이용해 로그인한 유저의 name을 받아옵니다.
- Btn_getdata.setOnClickListener
PhPGettotherCal(ID받는 변수, 유저네임)을 통해 GetotherCal.php를 불러와서 받아온 ID에 있는 일정들을 유저 네임의 일정들로 새로 서버 데이터베이스에 추가합니다.
Intent를 이용해 MainActivity로 이동합니다.

MainActivity.java

| 변수 명 | 변수 내용 |
|----------------|-------------------------|
| Str | string 값을 받는 변수 |
| calendarView | MaterialCalendarView 변수 |
| cha_Btn | 수정하는 버튼 |
| Del_Btn | 삭제하는 버튼 |
| Save_Btn | 저장하는 버튼 |
| Start_time | 시작시간 저장하는 버튼 |
| End_time | 끝나는 시간 저장하는 버튼 |
| Time_set | 시간 저장하는 버튼 |
| Reset_btn | 창 닫는 버튼 |
| diaryTextView | 내용을 화면에 나타내는 변수 |
| textView2 | 텍스트 값 받는 변수 |
| textView3 | 텍스트 값 받는 변수 2 |
| Time_start | 시작 시간 받는 변수 |
| Time_end | 끝나는 시간 받는 변수 |
| Year | 연도 받는 변수 |
| Month | 월 받는 변수 |
| Day | 일 받는 변수 |
| Change | 수정 여부 받는 변수 |
| Btn_open | 버튼 활성화 변수 |
| Btn_close | 버튼 비활성화 변수 |
| isBtnOpen | Open 변수 확인 버튼 |
| OpenBtnList | + 버튼 |
| MovetoBoard | 게시판 이동 버튼 |
| MovetoPersonal | 개인 일정 이동 버튼 |
| MovetoOpen | 공유 일정 이동 버튼 |
| alarmHour | 시간 받는 변수 |
| alarmMinute | 분 받는 변수 |

- getIntent()를 이용해 로그인한 유저의 네임을 받아옵니다.
- PhPCalenderGetDay(유저이름)을 통해 Calander_Pickdot.php를 불러와서 서버에 저장되어있는 일정들을 불러옵니다.

일정이 있는 날에는 addDecorators함수에 EventDecorator()함수를 호출해 이용해 빨간색 점을 찍습니다.

- 오늘 날짜에는 addDecorators함수에 EventDecorator()함수를 호출해 파란색 점을 찍습니다.
- addDecorators함수에 SundayDecorator(),SaturdayDecorator()함수를 호출해서 토요일과 일요일에 색상을 추가합니다.
- calendarView.setOnDateChangeListener
onDateSelected() 캘린더뷰에서 날짜가 선택되었을 때 해당 연도, 월, 일을 각 year,month,day 변수로 받습니다.
Reset,diaryTextView,save버튼만 VISIBLE상태로 전환합니다.
Checkday()함수를 호출합니다.
- Reset_btn.setOnClickListener
버튼 창,일정을 알려주는 창들 모두 INVISIBLE상태로 전환합니다.
- save_btn.setOnClickListener
일정 내용 칸 string값을 str변수에 받습니다.
PhPCalenderUpload(유저이름, 연도, 월, 일,str,change)을 통해 Calander_Main.php를 불러와서 변수들 값들을 change값에 따라 서버 데이터베이스에 저장합니다.
Change 값을 0으로 설정합니다.
일정이 추가된 날짜에 addDecorators함수 안 EventDecorator()함수를 호출해 이용해 빨간색 점을 찍습니다.
리셋버튼,시간 설정 레이아웃만을 VISIBLE상태로 전환합니다.
- End_time.setOnClickListener()
시간을 설정할 수 있는 TimePickerDialog를 불러옵니다
시간을 설정하면 시간, 분을 저장하게 되고 PhPCalenderEndTimeupload(유저이름, 연도, 월, 일, str, 설정한 시간, 설정한 분)을 통해 Calander_UploadEndtime.php를 불러와서 유저이름, 연도, 월, 일, str에 맞는 row에 시간과 분을 서버 데이터베이스에 저장합니다.
PhPCalenderEndTimeGet(유저이름, 연도, 월, 일, str, 설정한 시간, 설정한 분)을 통해 Calander_GetEndtime.php를 불러와서 시간과 분을 서버에서 불러와서 화면에 나타냅니다.

- start_time.setOnClickListener()
시간을 설정할 수 있는 TimePickerDialog를 불러옵니다
시간을 설정하면 시간, 분을 저장하게 되고 PhPCalenderStartTimeupload(유저이름, 연도, 월, 일, str, 설정한 시간, 설정한 분)을 통해 Calander_UploadStarttime.php를 불러와서 유저이름, 연도, 월, 일, str에 맞는 row에 시간과 분을 서버 데이터베이스에 저장합니다.
PhPCalenderStartTimeGet(유저이름, 연도, 월, 일, str, 설정한 시간, 설정한 분)을 통해 Calander_GetStarttime.php를 불러와서 시간과 분을 서버에서 불러와서 화면에 나타냅니다.
- Time_set.setOnClickListener()
수정, 삭제, 리셋 버튼만 VISIBLE상태로 전환합니다.
- Checkday(연도, 월, 일)
calendarView.setOnDateChangeListener에서 받은 연도, 월, 일을 받아옵니다.
PhPCalenderGet(유저이름, 연도, 월, 일)을 통해 Calander_MainGetDate.php를 불러와서 일정 내용을 불러옵니다
- Cha_btn.setOnClickListener()
PhPCalenderCha(유저이름, 연도, 월, 일, str)를 통해 Calander_Maincha.php를 불러와서 str을 수정해서 서버 데이터베이스에 저장합니다.
리셋버튼. 내용 나타내는 란, 저장 버튼만 VISIBLE 상태로 전환합니다
change값을 1로 설정합니다.
- Del_btn.setOnClickListener()
PhPCalenderDel(유저이름, 연도, 월, 일, str)를 통해 Calander_Maindel.php를 불러와서 변수들에 맞는 서버 데이터베이스에 있는 row값을 삭제합니다.
날짜에 있던 점을 삭제합니다.

NoticeActivity.java

| 변수 명 | 변수 내용 |
|---------------------------|-------------|
| Notice_Textview_title; | 제목 받는 변수 |
| Notice_Textview_timeline; | 일자 받는 변수 |
| Notice_Textview_content; | 내용 받는 변수 |
| Notice_Textview_username; | 유저 네임 받는 변수 |

- setStringExtra를 이용해 내용, 제목, 일자, 유저 네임 값들을 받아와서 게시물 텍스트 뷰를 통해 화면에 나타냅니다.

Noticeboard.java

| s | 변수 내용 |
|--------------------|-------------------------|
| Listview | 리스트 뷰 형식을 받는 변수 |
| myListAdapter | MyListAdapter 형식을 받는 변수 |
| Btn_Upload | 업로드 창으로 넘어가는 버튼 |
| List_itemArrayList | 게시판 형식 리스트 |
| isBtnOpen | Open 변수 확인 버튼 |
| OpenBtnList | + 버튼 |
| MovetoBoard | 게시판 이동 버튼 |
| MovetoPersonal | 개인 일정 이동 버튼 |
| MovetoOpen | 공유 일정 이동 버튼 |
| Btn_open | 버튼 활성화 변수 |
| Btn_close | 버튼 비활성화 변수 |

- getIntent()를 이용해 로그인한 유저의 이름을 받아옵니다.
- PhPNoticeGet()을 통해 Calender_NoticeGet.php를 불러와서 Notice테이블에 있는 모든 값들을 서버 데이터베이스에서 불러옵니다.
- List_itmeArrayList.add()를 통해 이미지, 이름, 제목, 일자, 내용들을 list_item 형식으로 추가합니다.

UploadNoticeboard.java

| 변수 명 | 변수 내용 |
|-------------|--------------------------------|
| now; | 현재 시간을 나타내는 변수 |
| mDate; | now 값을 일정 형식으로 받는 변수 |
| simpleDate; | yy-MM-dd hh:mm:ss 형식 일정 받는 변수 |
| getTime; | mDate 를 simepleDate 형식으로 받는 변수 |
| Title | 제목을 받는 변수 |
| Content | 내용을 받는 변수 |
| Btn_Upload | 게시물 내용을 업로드 하는 버튼 |

- getIntent()를 이용해 로그인한 유저의 이름을 받아옵니다.
- Btn_Upload.setClickListener
 PhPNoticeUpload(내용, 현재시간, 유저네임, 제목)을 통해 Calender_Notice.php를 불러와서 해당 값들을 서버 데이터베이스 Notice테이블에 저장합니다.
 입력한 값들을 intent.putExtra를 이용해 Noticeboard로 값을 넘겨줍니다.s

PHPRequest

- 안드로이드 스튜디오(클라이언트)에서 각 상황에 맞는 변수들을 아파치 서버로 보내기 위해 패키지들을 선언해 놓은 액티비티입니다.
- PHP 파일 중 query문 추가 설명
 1. Calender_Login.php
클라이언트에서 받아온 ID값과 PW값을 만족하는 ID값을 users테이블에서 찾습니다.
 2. Register.php
클라이언트에서 받아온 ID,PW,username,userAge값들을 users테이블에 삽입합니다.
 3. Calender_Getname.php
클라이언트에서 받아온 ID에 맞는 Name을 users테이블에서 찾습니다.
 4. Calender_Maincha.php
클라이언트에서 받아온 userID,Year,Month,Day,Content를 SUB2테이블에 삽입합니다.
 5. Calender_Maindel.php
클라이언트에서 받아온 userID와 Year, Month, Day, Content와 같은 row값을 Main테이블에서 삭제합니다.
 6. CalenderMainGetDate.php
클라이언트에서 받아온 userID와 Year, Month,Day,Content과 동일한 row에서의 Content, start_time, start_time_min, end_time, end_time_min값들을 Main테이블에서 찾습니다.

7. Calender_UploadEndtime.php

클라이언트에서 받아온 userID와 Year, Month, Day, Content과 Main테이블에서 동일한 row에 end_time, end_time_min값들을 업데이트 합니다.

8. Calender_UploadStarttime.php

클라이언트에서 받아온 userID와 Year, Month, Day, Content과 동일한 Main테이블에서 row에 start_time, start_time_min값들을 업데이트 합니다.

9. Calender_GetEndtime.php

클라이언트에서 받아온 userID와 Year, Month, Day, Content과 Main테이블에서 동일한 row에 end_time, end_time_min값들을 찾습니다.

10. Calender_GetStarttime.php

클라이언트에서 받아온 userID와 Year, Month, Day, Content과 동일한 Main테이블에서 row에 start_time, start_time_min값들을 찾습니다.

11. Calender_Main.php

If 구문: 클라이언트에서 받아온 userID와 Year, Month, Day, Content과 Main테이블에 삽입합니다.

Else 구문: 값들을 Main 테이블에서 조건에 맞는 row 값을 업데이트합니다.

\$query 구문: SUB2 테이블에 해당 조건에 맞는 row 값을 삭제합니다.

12. Calender_NoticeGet.php

\$query = "select * from Noticeboard order by timeline desc";

: Timeline의 역순에 따라 Noticeboard 테이블로부터 모든 값들을 찾습니다.

13. Calender_Notice.php

클라이언트에서 받아온 게시물의 내용, 타임라인, 작성자, 제목 값들을 Noticeboard 테이블에 삽입합니다.

14. Calender_Pickdot.php

Main테이블에서 userID와 동일한 row값들의 연도, 월, 일을 찾습니다.

15. Findid.php

Main테이블에서 userID와 동일한 row값의 이름을 찾습니다.

16. GetotherCal.php

- \$query = "INSERT INTO SUB (select * from Main where ID = '\$getID')";

: Main 테이블에서 user의 ID를 가진 값들을 SUB테이블로 삽입합니다.

- \$query = "UPDATE SUB SET ID = '\$userID', list = NULL";

: SUB 테이블 list값을 NULL로 설정하고 ID값들을 userID로 업데이트합니다.

- \$query = "INSERT INTO Main (SELECT * FROM SUB WHERE ID = '\$userID')";

: SUB 테이블에서 userID와 같은 ID값을 가진 row값들을 Main 테이블에 설정합니다.

- \$query = "TRUNCATE SUB";

: SUB 테이블을 초기화합니다.

E) 빌드 및 테스트 방법

1. Log 기능 활용

- Android Studio 코드 기능 중 하나인 log 를 활용해 각 상황 별 nickname 을 부여하고, 원하는 출력 문구를 지정한 뒤, 버튼 클릭 또는 특정 상황에 따른 변수 및 문구를 출력해 코드 내부 구현 중인 상황을 파악할 수 있게 했습니다.

```
2021-06-13 11:39:08.685 3502-3502/com.example.open_calander_junkyu_java3 D/NetworkSecurityConfig: No Network Security Config specified, using platform default
2021-06-13 11:39:08.803 3502-3543/com.example.open_calander_junkyu_java3 D/EGL_emulation: eglMakeCurrent: 0xd5e1a3c0: ver 3 0 (tinfo 0xd5e0f210)
2021-06-13 11:39:08.814 3502-3543/com.example.open_calander_junkyu_java3 D/EGL_emulation: eglMakeCurrent: 0xd5e1a3c0: ver 3 0 (tinfo 0xd5e0f210)
2021-06-13 11:39:08.867 3502-3502/com.example.open_calander_junkyu_java3 W/ActivityThread: handleWindowVisibility: no activity for token android.os.BinderProxy@71109f1
2021-06-13 11:39:09.278 3502-3502/com.example.open_calander_junkyu_java3 D/pickdot: Dotarry2021
2021-06-13 11:39:09.280 3502-3502/com.example.open_calander_junkyu_java3 D/pickdot: Dotarry2021
2021-06-13 11:39:09.280 3502-3502/com.example.open_calander_junkyu_java3 D/pickdot: Dotarry2021
2021-06-13 11:39:09.281 3502-3502/com.example.open_calander_junkyu_java3 I/chatty: uid=10163(com.example.open_calander_junkyu_java3) identical 6 lines
2021-06-13 11:39:09.282 3502-3502/com.example.open_calander_junkyu_java3 D/pickdot: Dotarry2021
2021-06-13 11:39:09.305 3502-3502/com.example.open_calander_junkyu_java3 I/Choreographer: Skipped 32 frames! The application may be doing too much work on its main thread.
2021-06-13 11:39:09.475 3502-3519/com.example.open_calander_junkyu_java3 I/er_junkyu_java: NativeAlloc concurrent copying GC freed 18009(1319KB) AllocSpace objects, 6(128KB) LOS o
2021-06-13 11:39:10.146 3502-3519/com.example.open_calander_junkyu_java3 I/er_junkyu_java: Background young concurrent copying GC freed 37532(1571KB) AllocSpace objects, 0(0B) LOS
2021-06-13 11:39:10.156 3502-3543/com.example.open_calander_junkyu_java3 D/EGL_emulation: eglMakeCurrent: 0xd5e1a3c0: ver 3 0 (tinfo 0xd5e0f210)
2021-06-13 11:39:10.212 3502-3543/com.example.open_calander_junkyu_java3 D/EGL_emulation: eglMakeCurrent: 0xd5e1a3c0: ver 3 0 (tinfo 0xd5e0f210)
2021-06-13 11:39:10.212 3502-3543/com.example.open_calander_junkyu_java3 D/EGL_emulation: eglMakeCurrent: 0xd5e1a3c0: ver 3 0 (tinfo 0xd5e0f210)
2021-06-13 11:39:10.212 3502-3543/com.example.open_calander_junkyu_java3 I/Choreographer: Skipped 54 frames! The application may be doing too much work on its main thread.
```

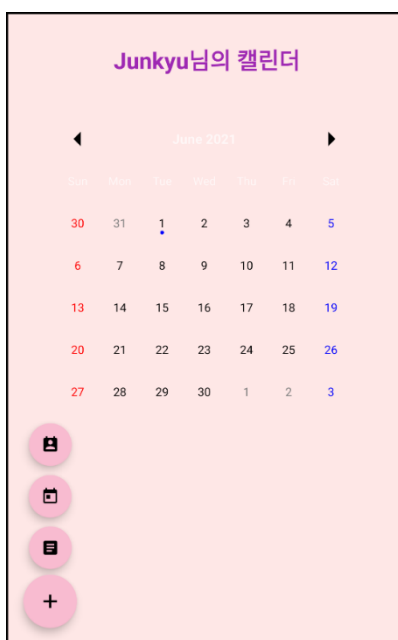
2. 상황에 따른 기능 구현 확인

- 개발자가 의도한 방향으로 코드가 작성되고 실행되는지 확인하기 위해 여러 상황을 부여하고 실행해 진행 경과를 확인했습니다.
 1. 다른 아이디의 데이터를 성공적으로 받아오는지?
 - ID 를 구분 짓고 서버의 데이터를 잘 받는지 확인하기 위해
 2. 게시판의 모든 글이 정상적으로 로드 되는지?
 - 서버와의 통신이 원활한지 확인하기 위해
 3. 중복된 아이디의 로그인을 차단하는지?
 - DBMS 상 설정한 중복 불가 튜플에 대한 설정이 잘 되었는지 확인하기 위해
- 해당 사항에 대한 검증은 아래 테스트 결과 및 분석에 정리되어 있습니다.

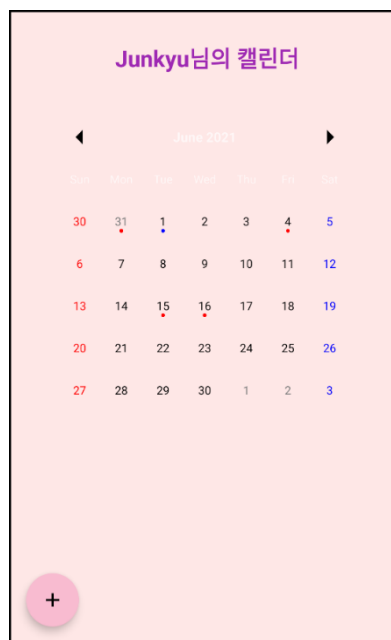
IV. 테스트 결과 및 분석

1. 여러 개의 아이디를 통해 일정 받아오기가 가능한지?

하나의 아이디를 입력해서 일정을 받아온 뒤, 다시 오픈 캘린더 버튼으로 들어가 다른 아이디를 입력하면, 그 아이디에 있는 일정 또한 받아올 수 있습니다.



일정 x



1 번아이디 일정



2 번아이디 일정

2. 사용자 1 로 게시판에 글을 작성한 뒤 사용자 2 로 들어가 확인해도 게시판 글이 남아있는지?

게시판의 글은 안드로이드 스튜디오 내부에 저장이 되는 것이 아닌, php 파일을 통해 데이터베이스로 저장이 되는 형식이기 때문에 게시판의 글은 다른 아이디로 로그인하더라도 남아있습니다.

3. 중복된 아이디로 회원가입이 불가능한지?

데이터베이스 테이블은 아이디가 메인 키입니다. 저희 서버 데이터는 메인 키가 중복될 경우에 정보 등록이 되지 않습니다. 아이디를 메인 키로 설정했기 때문에 나머지 정보들이 입력되더라도 메인 키가 중복될 경우 '중복된 아이디'라는 메시지를 출력하며 회원가입이 진행되지 않습니다.

V. 문제점 및 개선 방안

1. 이전 달력 로드 시간

로그인 이후 해당 월 및 이후 달력에 대한 로딩은 문제없이 잘 로딩되나 현재 달 이전으로 이동할 시 5 초가량의 로딩이 존재합니다.

AVD 만의 문제로 생각했으나 안드로이드 스마트폰에서도 같은 오류 발생을 확인했고 Calendar 오픈소스의 최적화 문제라고 파악 중입니다.

2. 회원가입 ID 중복 확인

회원가입 시 ID 를 중복 확인하지 않고 그대로 DB 에 입력하려는 오류가 있었습니다.

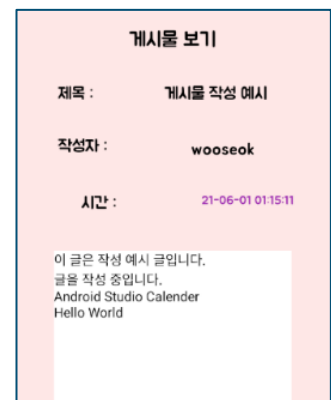
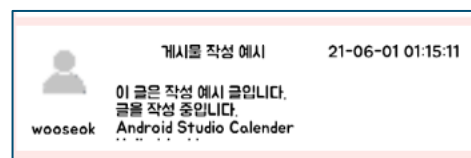
이를 해결하기 위해 Android Studio 측에서 JAVA 코드를 수정하였으나 해결되지 않았고, 연구를 거듭한 끝에 단순히 DBMS SQL 문 상에 중복 불가 옵션을 넣어 해결할 수 있었습니다.

이후 회원가입 시 ID 가 중복될 경우 오류가 return 되어 경고문과 함께 ID 를 다시 설정하도록 제작했습니다.



3. 게시판 내용 길이 출력 문제

게시판 내용이 길어지면 아래 예시 사진과 같이 잘려 출력되는 현상이 있어 이를 해결하기 위해 새로운 레이아웃을 출력해 전체 게시글을 편하게 확인할 수 있도록 만들었습니다.



4. 게시판 다중 데이터 로드 문제

게시판 데이터를 받아올 때 개발 도중 사용 중이던 기술로는 한 개의 데이터만 받아와 출력시키기에 얼마나 많은 데이터를 가져와야 할지 모르는 상태에서 while 문이나 for loop 을 사용하는 것이 제한되었습니다.

이를 해결하기 위해 JSON 개념을 도입했고, 이는 데이터를 array 형태로 받아와 원하는 만큼 데이터를 출력할 수 있었고 코드 구조도 매우 간단하게 작성할 수 있었습니다.

```

PHPRequest request = new PHPRequest( url: "http://133.186.229.67/Calendar_NoticeGet.php");
String result_date = request.PHPNoticeGet();
JSONArray Noticearray = new JSONArray(result_date);
Log.d( tag: "pickdot", mmsg: "Noticearray" + Noticearray );
for(int i=0; i<Noticearray.length();i++){
    JSONObject NoticeObject = Noticearray.getJSONObject(i);
    String not_name = NoticeObject.getString( name: "writer");
    String not_title = NoticeObject.getString( name: "Title");
    String not_date = NoticeObject.getString( name: "timeLine");
    String not_content = NoticeObject.getString( name: "Content");

    list_itemArrayList.add(
        new list_item(R.drawable.human, not_name, not_title, not_date, not_content));

    Log.d( tag: "pickdot", mmsg: "name : " + not_name);
    Log.d( tag: "pickdot", mmsg: "title : " + not_title);
    Log.d( tag: "pickdot", mmsg: "date : " + not_date);
    Log.d( tag: "pickdot", mmsg: "content : " + not_content);
}

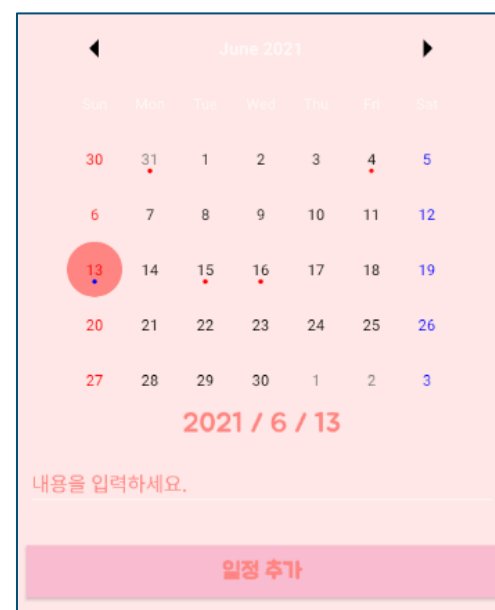
```

위 사진에서 볼 수 있듯 JSONArray 로 result_date 를 받아와 저장하고 for loop 의 getObject 중 i 번째 열을 받아와 not_변수들로 데이터를 넘겨줍니다. 이후 list_itemArrayList 에 add 해 게시판 데이터를 순차적으로 집어넣어 성공적으로 로드할 수 있었습니다.

5. 주말 색깔 표시

Calendar 오픈소스를 통해 달력을 가져온 뒤 확인해본 결과 평일과 주말의 색이 모두 검정이라 지나치게 밋밋해 보인다는 단점이 있었습니다.

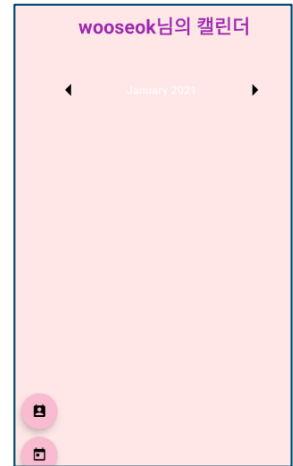
이를 해결하기 위해 Saturday Decorator, Sunday Decorator 코드를 만들어 각각 파랑, 빨강 색상을 입히는데 성공했습니다.



VI. 버그 리포트

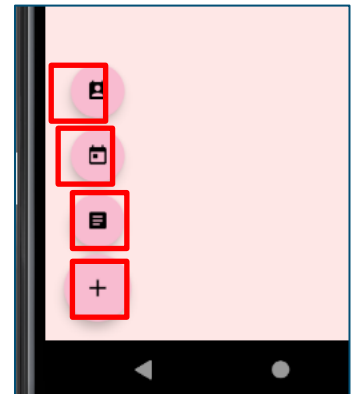
1. 이전 달력 로딩 속도가 5 초가량 발생하는 문제

- 로그인 후 이전 달로 이동하면 로딩이 발생하고, 다른 달로 이동한 후 돌아오면 로드 완료되는 현상이 있습니다.
- 약 5 초정도 대기하면 정상적으로 로드 가능합니다. 개발환경의 차이에 따라 발생하는 문제로 파악했습니다.



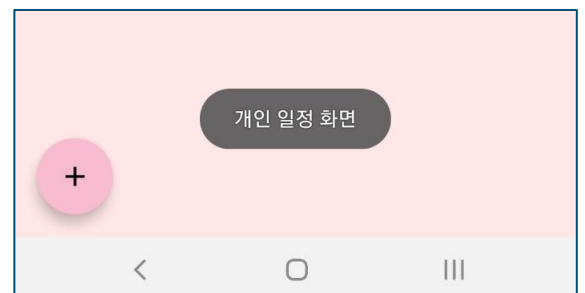
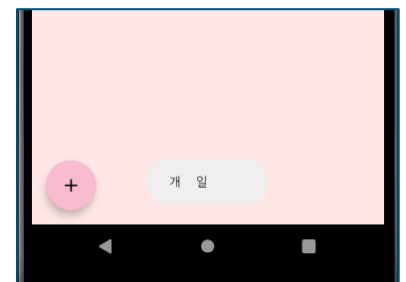
2. 캘린더 좌측하단 + 버튼의 메뉴들의 클릭 위치가 아이콘과 맞지 않는 문제

- 게시판 이동, 개인 일정 보기, 다른 사람 일정 불러오기 아이콘의 로드 버튼 위치가 보이는 버튼보다 살짝 왼쪽에 위치한 문제가 있습니다.
- 버튼 배열을 적용할 때 가로로 나열하는 기능을 적용함에 따라 발생한 문제라고 파악됩니다.



3. 알림 문구가 정상적으로 출력되지 않는 문제

- 우측의 사진과 같이 "개인 일정 화면"이라는 문구가 깨져서 출력되는 현상
- UTF-8 이 AVD 환경에서 제대로 로드되지 않아 발생하는 문제로 파악됩니다. 실제 안드로이드 휴대폰에서는 정상적으로 출력됩니다.



VII. 프로젝트 후기

처음 프로젝트를 시작할 때 직접 어플리케이션을 만들어보고자 하는 마음만으로 막연하게 오픈 캘린더 제작으로 방향을 설정했습니다. 하지만 막상 프로젝트를 시작하니 어플리케이션 UI 설정, DB 서버 관리 등 해결해야하는 양이 상당히 많았고 DBMS, Android Studio Java, PHP 등을 공부하는데 많은 시간을 쏟았습니다.

시간이 지나고 실질적으로 진행된 내용이 거의 없었으며 프로젝트 마감 날짜가 다가올수록 기간까지 못 끝낼 수도 있다는 불안감이 커져만 갔습니다. 하지만 각자 공부를 마치고 어플리케이션 제작 과정에 돌입하니 작업은 수월히 진행됐고, 각자 맡은 파트를 하나씩 결합해보니 로그인 (회원가입), 일정 공유, 게시판 등 저희가 구현하려고 했던 상당 부분이 완성된 상태였습니다.

특히 이번 프로젝트를 통해 느낀 점은 서버 사용의 사용 방법 및 유용성과 협업의 중요성이었습니다. 서버를 사용함으로 한 명의 사용자가 단순히 자신의 기기에 저장만 할 수 있었던 데이터들을 다른 사용자와 공유할 수 있게 만들었습니다. 또, 그 내용을 구현하기 위한 php 를 공부해 서버의 사용방법과 접속 방법 등을 배울 수 있었습니다. 조원들 사이의 협업도 매우 중요하다고 느껴졌습니다. 각 조원마다 역할을 분담함으로 인해 개인의 부담감을 덜 수 있었고 각자 도맡은 부분을 완료하고 서로 진행 내용을 공유하며 프로젝트의 진행 방향을 점검할 수 있었습니다.

팀 프로젝트를 진행해본 경험들이 적어 진행이 서툴렀지만 이번 오픈 캘린더 앱 제작을 통해 프로그래밍은 물론이고, 프로젝트 관리 방법 또한 배울 수 있었습니다. 이후 진행하게 될 수많은 팀 프로젝트는 이번 경험을 토대로 더욱 수월하게 해낼 수 있겠다는 자신감을 저희 팀 모두가 갖게 되었습니다.

최종발표 이후에 게시판의 디자인, 수정 삭제 등 일정 기입하는 칸을 없애는 버튼 아이디어의 중복확인 버튼을 추가적으로 구현하였습니다. 피드백 당시 받았던 비밀번호와 아이디의 보안 추가하는 작업, 본인 확인하는 작업을 시간상 해내지는 못했습니다. 하지만 앞으로 종강이후에도 해당 프로젝트를 계속 조금씩 구현해내면서 앱을 플레이스토어에 내놓는 것이 최종 목표입니다. 앞으로도 현재 팀을 유지하고 싶을 정도로 너무 뜻깊은 시간이었습니다.

VIII. 용어 해설 및 참고 문헌

참고 문헌

- <https://developer.android.com/>
안드로이드 스튜디오 개발자 문서 사이트
안드로이드 스튜디오의 기능들 기재
- <https://github.com/prolificinteractive/material-calendarview>
안드로이드 스튜디오 기본 캘린더 뷰가 아닌 커스텀 캘린더 뷰
추가 구현 가능한 기능들 분석
- http://tcpschool.com/javascript/js_intro_basic
자바스크립트의 문법 참고

용어 해설

- 플로팅 버튼 : 앱 UI 의 기본 작업을 트리거하는 원형 버튼이다.
- AVD : Android Virtual Device 의 줄임 말로, Android Emulator 에서 시뮬레이션하려는 Android 스마트폰이나 태블릿, Wear Os, Android TV, Automotive OS 기기의 특성을 정의하는 구성이다.
- 튜플 : 데이터베이스 내의 주어진 목록과 관계 있는 속성값의 모음이다. 관련 테이블에서 행한 수치 이상으로 혼합된 자료 요소를 의미한다.