

# 모바일 프로그래밍

## 최종 보고서



담당교수	최정열 교수님
학과	컴퓨터 공학과
학년	3 학년
학번	20160954, 20170838
이름	정연재, 신다영
제출일	6/16

## <목차>

<b>1. 앱 개요</b>	3
1.1 주제	3
1.2 목적	3
1.3 필요성	3
<b>2. 앱 주요 기능</b>	3
<b>3. 화면 UI 설계서(XML)</b>	3
3.1 화면 구성 및 기능	4
3.1.1 메인 화면	4
3.1.2 달력 화면	4
3.1.3 일기 작성 화면	5
3.1.4 날짜 버튼 메뉴상자	6
3.1.5 기분 버튼 메뉴상자	7
3.1.6 메뉴 화면	8
3.1.7 셋팅 화면	9
3.1.8 비밀번호 화면	10
3.2 화면 연관성	11
3.2.1 화면 시나리오	11
3.2.1.1 메인화면	11
3.2.1.2 달력화면	11
3.3 화면 시나리오	11
3.3.1 메인 화면	11
3.3.2 달력 화면	12
3.4 화면 전환	13
3.5 UI 구현 방법	18
3.5.1 content_main.xml	18
3.5.2 writeday.xml	22
3.5.3 setting.xml	22
3.5.4 bimil.xml	24
3.5.5 dlgform.xml	26
3.5.6 image.xml	26
3.5.7 day.xml	27
3.5.8 activity_main_drawer.xml	27
<b>4. 시스템 설계서(Java)</b>	28
4.1 WriteDay.java	28
4.1.1 자바 객체 설명	28

4.1.2	메서드 설명-----	29
4.1.3	버튼 클릭 이벤트 처리-----	30
4.1.3.1	wdDate 버튼-----	30
4.1.3.2	wdFeel 버튼-----	31
4.1.3.3	wdWeather 버튼-----	31
4.1.3.4	wdCategory 버튼-----	31
4.1.3.5	btnSubmit 버튼-----	32
4.1.3.6	메서드 설명-----	33
4.2	MainActivity.java-----	35
4.2.1.1	자바 변수 객체 설명-----	35
4.2.1.2	메서드 설명-----	36
4.2.1.3	CalendarAdapter.java-----	38
4.2.1.4	자바 변수 설명-----	38
4.2.1.5	메서드 설명-----	39
4.3	DayInfo.java-----	40
4.3.1	자바 변수 설명-----	40
4.3.2	메서드 설명-----	40
4.3.2.1	setting.java-----	41
4.3.2.2	자바 객체 설명-----	41
4.3.2.3	버튼 클릭시 이벤트 처리-----	41
<b>5.</b>	<b>기타-----</b>	<b>42</b>
5.1	개발 일정-----	42
5.2	업무 분담-----	43

## 1. 앱 개요

### 1.1 주제

나의 다이어리는 하루의 일과를 기록할 수 있는 앱이다. 하루의 기분과 날씨와 어떤 일이 있었는지를 기록할 수 있는 나만의 일기장과 같다. 다이어리 기록을 하면 최근 한달 동안 어떤 기분이었는지도 볼 수 있게 만들어져 있다.

### 1.2 목적

사람들은 새해마다 다이어리를 구매한다. 하지만 구매한 다이어리를 작성하는 일은 작심삼일이고, 다이어리를 매번 챙겨 나가기도 번거로울 뿐만 아니라 매번 하루를 다이어리 작성으로 끝내기는 쉽지 않다. 이런 이유로 어디서나 접근하기 간편하고 실용적인 다이어리 앱을 만드는 것이 우리 프로젝트의 목적이다.

### 1.3 필요성

스마트폰으로 인해 어디서나 휴대하기 간편하고, 짧은 시간 내에 일기를 작성 할 수 있는 다이어리 앱이다. 또한 실체가 있는 다이어리를 보관하기 위해서는 주로 좌물쇠를 사용하는데, 이는 다이어리 내용을 보관하기에는 한계가 있다. 이를 극복하기 위해서 다이어리 앱은 비밀번호를 설정함으로써 다른 사람이 나의 다이어리에 접근하는 것을 제한한다.

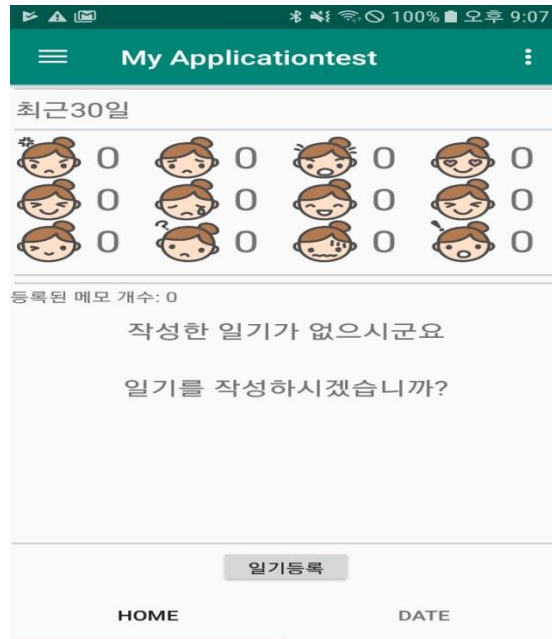
## 2. 앱 주요 기능

- 일기 입력 일기를 입력할 때 어떤 일이 있었는지에 대해서만 기록하는 것이 아니라 그날의 기분, 날씨, 사진을 넣을 수 있고 작성한 일기를 구분을 하게 되면 어느 카테고리 들어가는지를 선택해서 작성할 수 있게 된다. 그리고 일기마다 색상을 다르게 설정하여 작성할 수 있다.
- 달력으로 일기관리 하루하루 쓴 일기를 달력으로 달마다 보여주게 된다. 이렇게 보여줌으로써 쉽게 파악할 수 있다는 장점이 있다

## 3.화면 UI 설계서(XML)

### 3.1. 화면 구성 및 기능

#### 3.1.1 메인 화면



[그림 1] 메인 화면

- ListView 에 저장된 파일이 없다면 '작성한 일기가 없으시군요 일기를 작성하시겠습니까?'라는 TextView 가 VISIBLE 되고 ListView 에 저장된 파일이 있다면 TextView 는 GONE 되어진다.
- 일기등록 버튼을 누르면 일기 작성 화면으로 넘어가진다.

### 3.1.2 달력 화면



[그림 2] 달력 화면

- 두 번째 탭 DATE 를 선택하면 달력 화면이 나온다.
- 상단 바에 이전 달 버튼을 누르면 이전 달 달력 화면이 나온다.
- 상단 바에 다음 달 버튼을 누르면 다음 달 달력 화면이 나온다.
- 사용자가 원하는 날짜를 선택하면 일기 작성 화면이 나온다.

### 3.1.3 일기 작성 화면

[그림 3] 일기 작성 화면

- 일기 작성 화면에서는 사용자가 원하는 날짜와 기분, 날씨, 분류를 선택할 수 있다.
- 제목과, 일기를 작성 할 수 있다.
- 사진을 추가 할 수 있다.

#### 3.1.4 날짜 버튼 대화상자

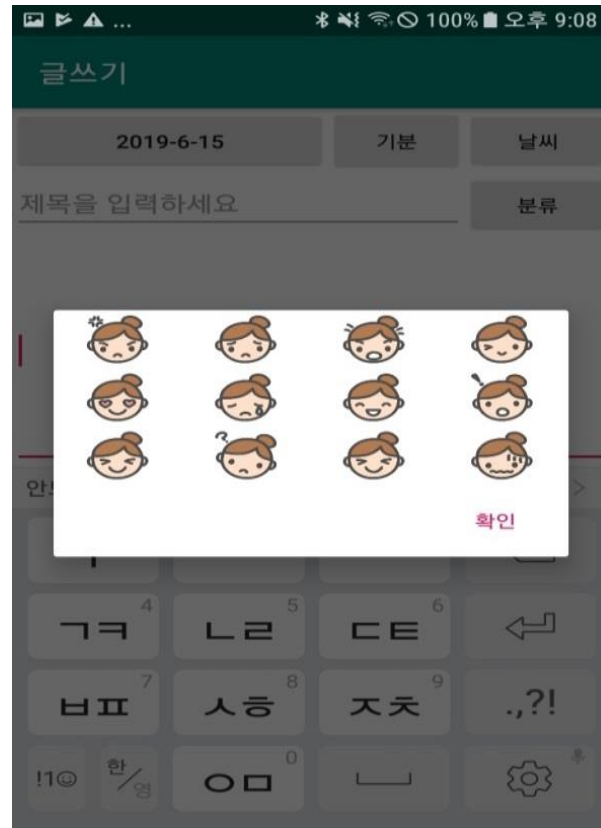


[그림 4] 날짜 버튼 대화상자

- 날짜 버튼을 누르면 calendarView 의 대화상자가 띄어진다.
- 취소 버튼을 누르면 아무런 동작 없이 대화상자만 닫힌다.
- 사용자가 원하는 날짜를 선택하고 확인 버튼을 누른다.
- 날짜 버튼의 text 가 사용자가 선택한 날짜로 변환된다.

### 3.1.5 기본 버튼 대화상자

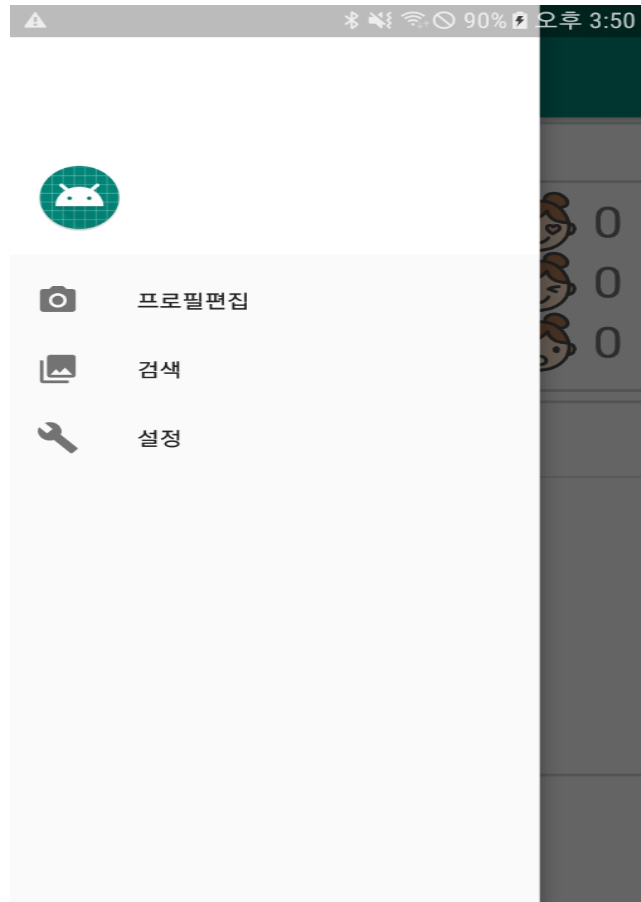




[그림 5] 기분 버튼 대화상자

- 기분 버튼을 누르면 12 가지 기분 아이콘의 대화상자가 띄어진다.
- 사용자가 기분 아이콘을 선택하고 확인 버튼을 누른다.
- 기분 버튼의 text 는 기분 아이콘의 이름으로 변환된다.

### 3.1.6 메뉴 화면



[그림 6] 메뉴 화면

- 메인 화면에서 메뉴 아이콘을 누르면 메뉴 화면 창이 뜬다.
- 메뉴 설정 item 을 누르면 셋팅 화면 창이 뜬다.

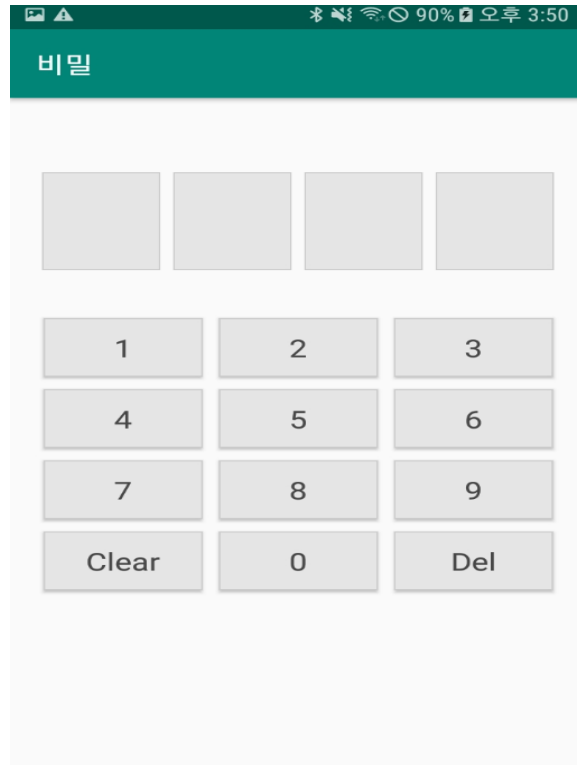
### 3.1.7 셋팅 화면



[그림 7] 설정 화면

- 비밀번호 설정 Switch 버튼을 on 하면 비밀번호가 설정된다.
- 비밀번호 변경 버튼을 누르면 비밀번호 화면 창이 뜨면서 비밀번호를 변경할 수 있다.

### 3.1.8 비밀번호 화면



[그림 8] 비밀번호 화면

## 3.2. 화면간 연관성

### 3.2.1. 화면 시나리오

#### 3.2.1.1. 메인 화면

기능 명	메인 화면에서 글 등록
액터 명	사용자
개요	메인 화면에서 글을 등록하기 위해서 사용한다.
사전 조건	첫 번째 탭의 메인 화면이 뜬다.
사후 조건	저장한 글은 메인 화면에 리스트로 보여진다.
기본 흐름	<ol style="list-style-type: none"> <li>1. 메인 화면에서 글 등록 버튼을 누른다.</li> <li>2. 글 작성 화면이 뜬다.</li> <li>3. 날짜를 선택한다.</li> <li>4. 기분, 날씨, 분류를 선택한다.</li> <li>5. 제목을 입력한다.</li> <li>6. 일기 내용을 입력한다.</li> <li>7. 저장버튼을 누른다.</li> <li>8. 메인 화면이 뜬다.</li> </ol>
대체 흐름	▶ 사진추가 버튼을 눌렀을 경우 단말기의 갤러리 화면이 띄어진다.

[그림 9] 메인 화면 시나리오

#### 3.2.1.1. 달력 화면

기능 명	달력 화면에서 글 등록
액터 명	사용자
개요	달력 화면에서 글을 등록하기 위해서 사용한다.
사전 조건	두 번째 탭의 달력 화면이 뜬다.
사후 조건	저장한 글은 메인 화면에 리스트로 보여진다.
기본 흐름	<ol style="list-style-type: none"> <li>1. 달력 화면에서 원하는 날짜 화면을 클릭한다.</li> <li>2. 기분, 날씨, 분류를 선택한다.</li> <li>3. 제목을 입력한다.</li> <li>4. 일기 내용을 입력한다.</li> <li>5. 저장버튼을 누른다.</li> <li>6. 달력 화면이 뜬다.</li> </ol>
대체 흐름	▶ 사진추가 버튼을 눌렀을 경우 단말기의 갤러리 화면이 띄어진다.

[그림 10] 달력 화면 시나리오

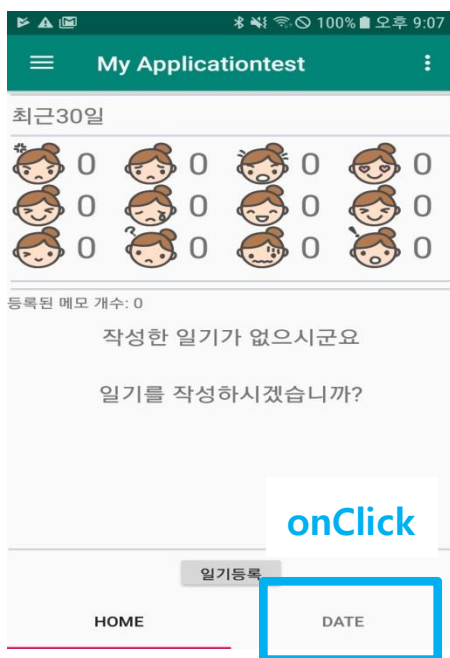
### 3.2.2. 화면 전환



Intent  
➔



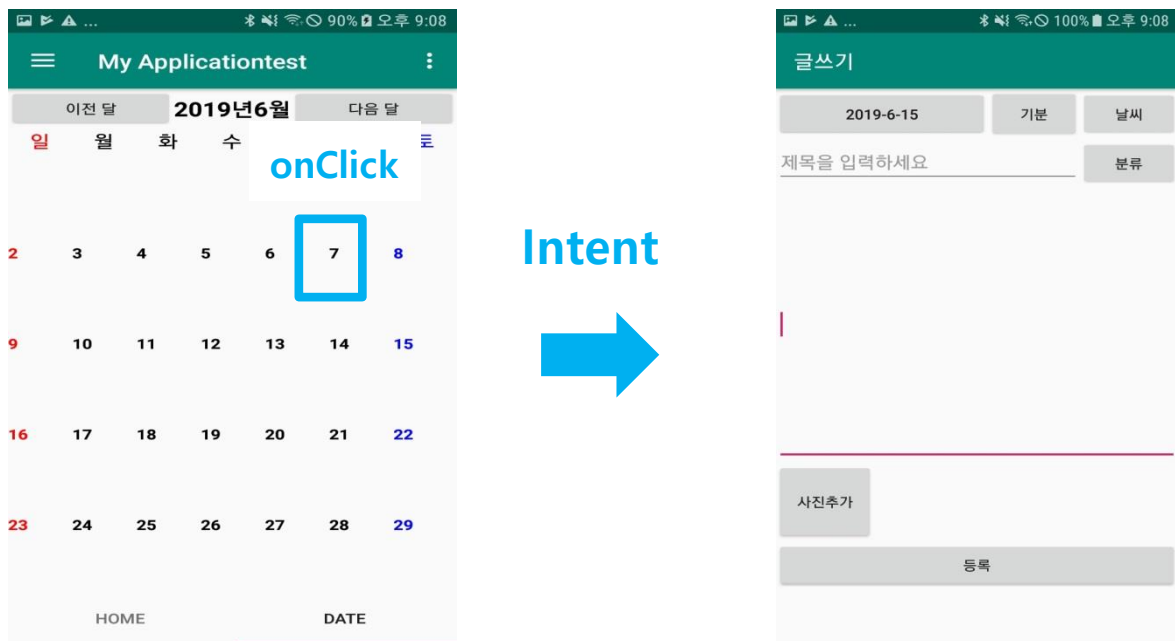
[그림 11] 메인 화면에서 일기 작성 화면 전환



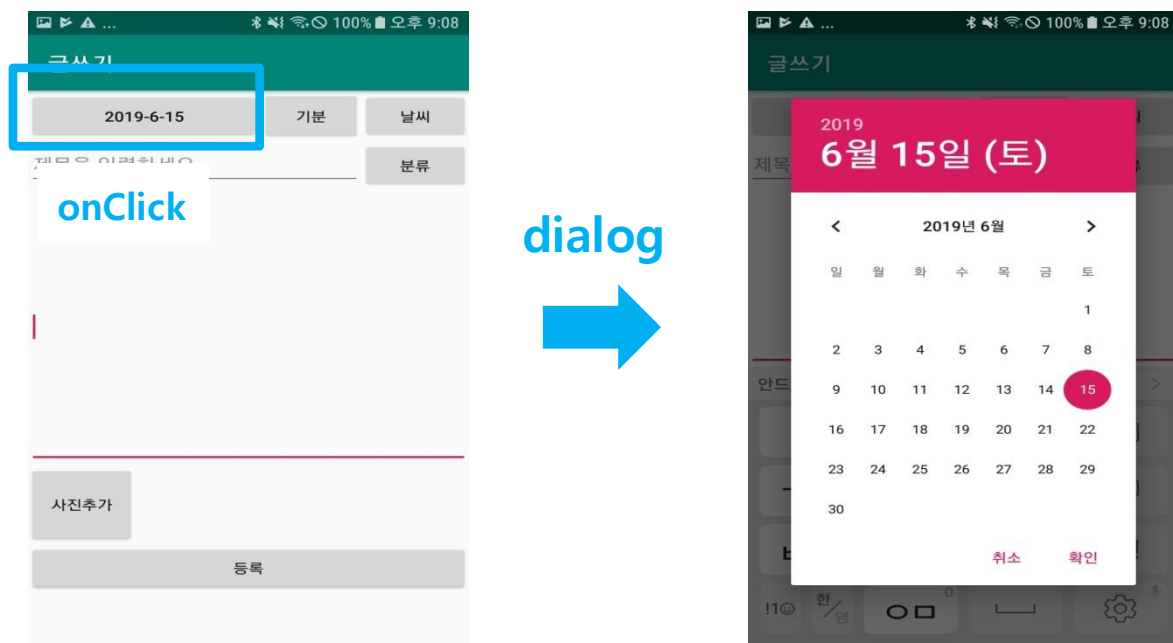
TabHost  
➔



[그림 12] 메인 화면에서 달력 화면 전환

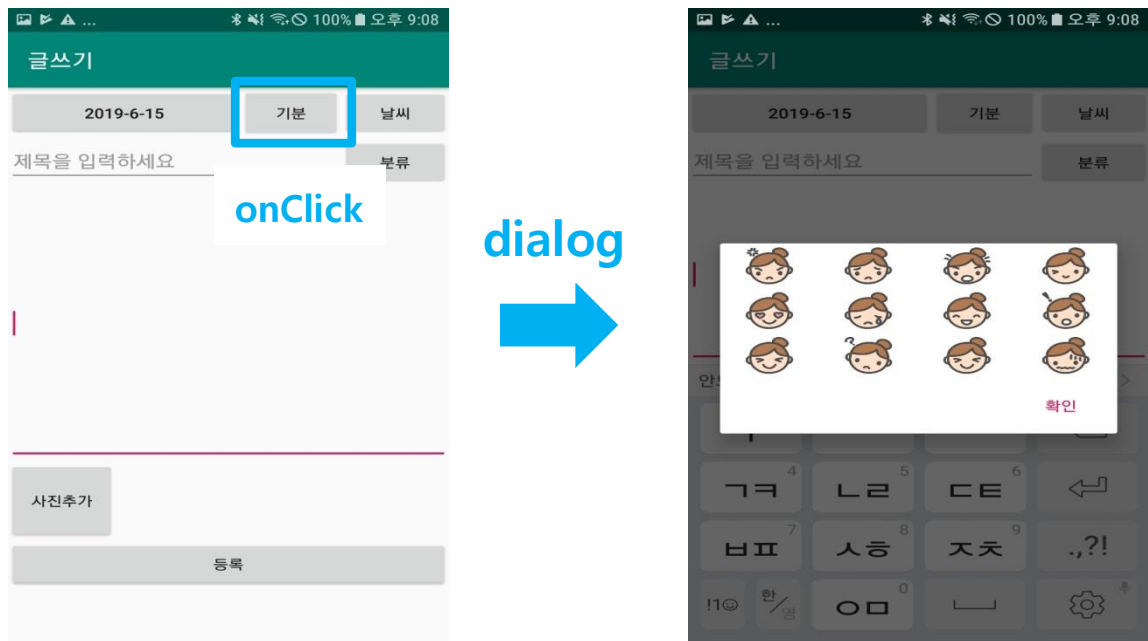


[그림 13] 달력 화면에서 일기 작성 화면 전환

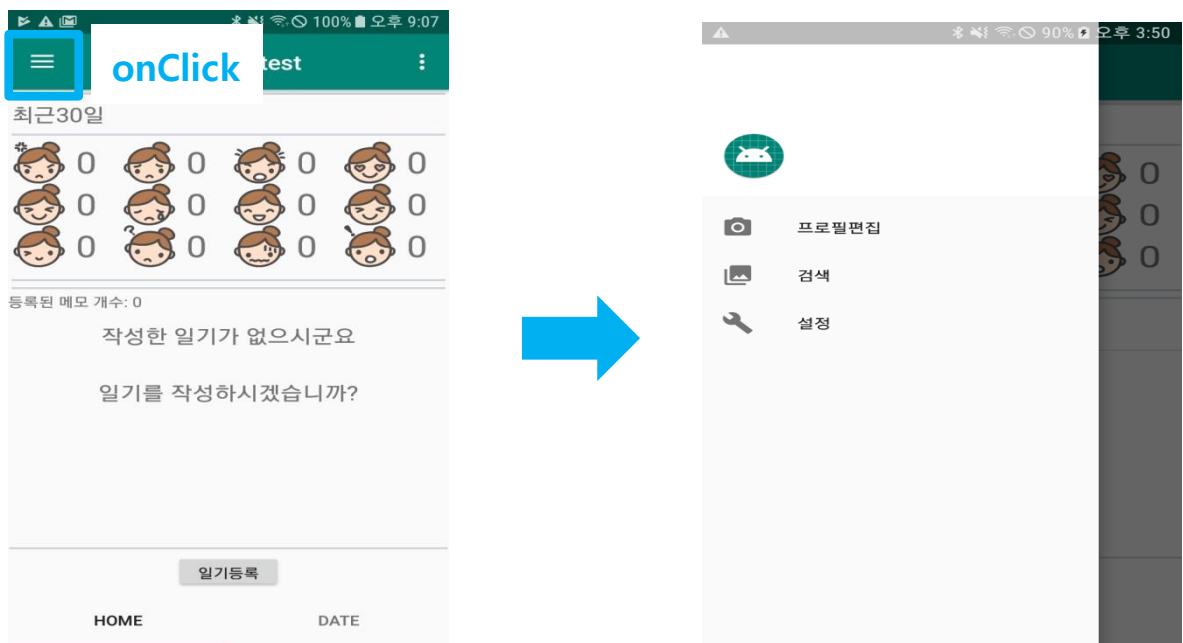


[그림 14] 날짜 버튼 클릭 시 대화 상자

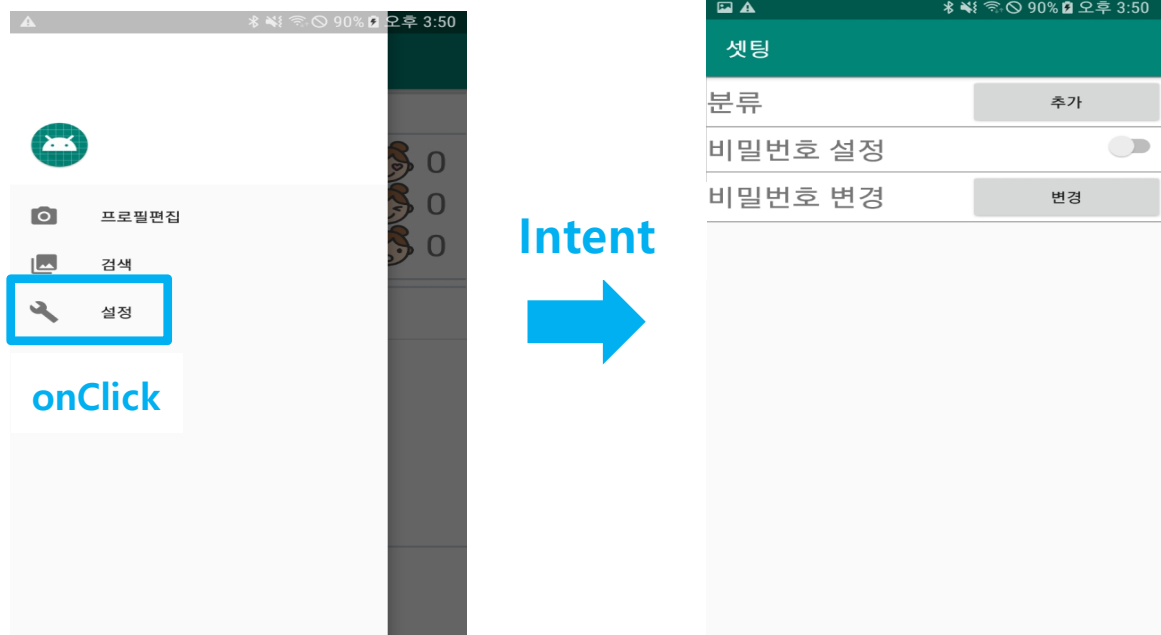




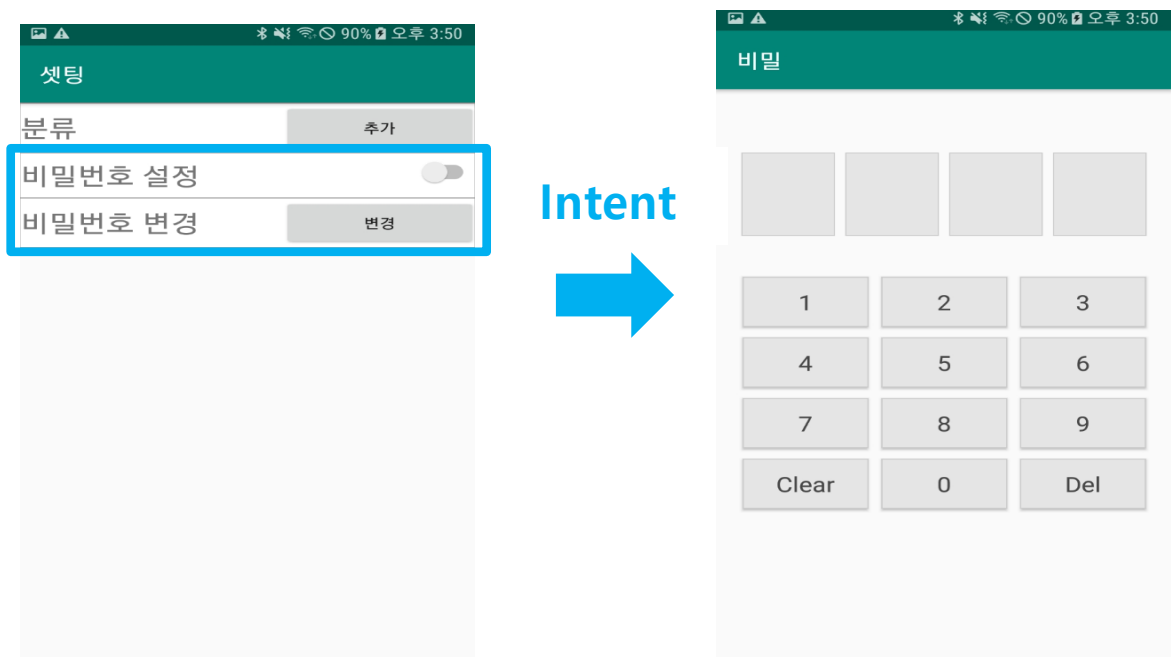
[그림 15] 기분 버튼 클릭 시 대화 상자



[그림 16] 메뉴 아이콘 클릭 시 메뉴 화면



[그림 17] 메뉴 설정 item 클릭 시 셋팅 화면



[그림 18] 비밀번호 관련 버튼 클릭 시 셋팅 화면

### 3.3. UI 구현 방법

#### 3.3.1. content\_main.xml

```
<TabHost android:layout_height="match_parent" android:layout_width="match_parent"
android:id="@+id/Host">
```

```
<LinearLayout android:orientation="vertical" android:layout_height="match_parent"
android:layout_width="match_parent">
```

```
<FrameLayout android:layout_height="match_parent"
android:layout_width="match_parent" android:id="@android:id/tabcontent"
android:layout_weight="1">
```

```
<LinearLayout android:orientation="vertical" android:layout_height="match_parent"
android:layout_width="match_parent" android:id="@+id/tabHome">
```

```
<LinearLayout android:orientation="vertical" android:layout_height="match_parent"
android:layout_width="match_parent" android:id="@+id/tabDate">
```

[그림 19] . content\_main 코드 1

- TabHost 와 FrameLayout 을 이용하여 2 개의 LinearLayout 의 id 를 각각 tabHome, tabDate 로 설정한다

```
<GridLayout android:layout_height="132dp" android:layout_width="358dp"
```

```
xmlns:android="http://schemas.android.com/apk/res/android"
```

```
android:layout_marginTop="70dp" android:layout_alignParentTop="true"
```

```
android:rowCount="4" android:columnCount="8"
```

```
android:layout_centerHorizontal="true">
```

```
<ImageView android:layout_height="42dp" android:layout_width="45dp"
```

```

android:id="@+id/img1" android:src="@drawable/bbackchim"/>

    <TextView android:layout_height="42dp" android:layout_width="45dp"
android:id="@+id/itg1" android:textSize="30dp" android:text=" 0"/>

    .
    .
    <중간 생략>
    .
    .

<FrameLayout android:layout_height="255dp" android:layout_width="match_parent">

    <ListView android:layout_height="match_parent"
android:layout_width="match_parent" android:id="@+id/listview"/>

    <TextView android:layout_height="wrap_content"
android:layout_width="wrap_content" android:id="@+id/tvCount" android:text="등록된
메모 개수: 0" android:layout_weight="1"/>

    <LinearLayout android:orientation="vertical" android:layout_height="230dp"
android:layout_width="match_parent" android:id="@+id/diaryLinear">

        <TextView android:layout_height="wrap_content"
android:layout_width="wrap_content" android:id="@+id/text1" android:textSize="20dp"
android:text="작성한 일기가 없으시군요" android:layout_margin="30dp"
android:layout_gravity="center_horizontal|center" android:layout_marginLeft="87dp"
android:layout_marginStart="87dp" android:layout_marginBottom="145dp"/>

        <TextView android:layout_height="wrap_content"
android:layout_width="wrap_content" android:id="@+id/text2" android:textSize="20dp"
android:text="일기를 작성하시겠습니까?" android:layout_margin="0dp"
android:layout_gravity="center_vertical|center_horizontal"
android:layout_marginBottom="111dp" android:layout_marginRight="82dp"
android:layout_marginEnd="82dp" android:layout_alignParentBottom="true"

```

```

android:layout_alignParentRight="true" android:layout_alignParentEnd="true"/>

</LinearLayout>

</FrameLayout>

<Button android:layout_height="wrap_content" android:layout_width="wrap_content"
android:id="@+id/btn1" android:text="일기등록" android:layout_margin="3dp"
android:layout_gravity="center_vertical|center_horizontal"
android:layout_marginLeft="149dp" android:layout_marginStart="149dp"
android:layout_marginBottom="33dp" android:onClick="onClick"/>

```

[그림 20] content\_main 코드 2

- GridLayout 을 사용하여 기분 이미지와 글 작성에서 그 이미지를 선택할 때마다 증가하는 TextView 위젯을 넣는다.
- FrameLayout 을 사용하여 작성한 파일이 없을 때는 TextView 가 화면에 띄어지고 작성한 파일이 있을 경우 ListView 가 보여지도록 한다.
- 버튼을 누르면 일기 작성 화면이 띄어지게 하기 위해서 onClick 을 설정한다.

```

<Button android:layout_height="wrap_content" android:layout_width="0dp"
android:id="@+id/btn_previous_calendar" android:layout_weight="1" android:text="이전
달"/>

<TextView android:layout_height="wrap_content" android:layout_width="wrap_content"
android:id="@+id/tv_calendar_title" android:textSize="20sp" android:text="2018 년 2 월"
android:gravity="center" android:textStyle="bold" android:textColor="#000"/>

<Button android:layout_height="wrap_content" android:layout_width="0dp"
android:id="@+id/btn_next_calendar" android:layout_weight="1" android:text="다음
달"/>

```

[그림 21] content\_main 코드 3

- Id 가 tabDate 인 LinearLayout 에서 이전 달로 넘어갈 버튼과 다음 달로 넘어갈 버튼을 설정한다.

```
<LinearLayout android:orientation="horizontal" android:layout_height="wrap_content"
android:layout_width="match_parent" android:layout_gravity="center">

<TextView android:layout_height="wrap_content" android:layout_width="0dp"
android:layout_weight="1" android:textSize="18sp" android:text="일"
android:gravity="center" android:textColor="#ff0000"/>

<TextView android:layout_height="wrap_content" android:layout_width="0dp"
android:layout_weight="1" android:textSize="18sp" android:text="월"
android:gravity="center" android:textColor="#000000"/>

.
.
<생략>
.
.

</LinearLayout>

<GridView android:layout_height="match_parent"
android:layout_width="match_parent" android:id="@+id/gv_calendar"
android:layout_gravity="center" android:verticalSpacing="0dp"
android:stretchMode="columnWidth" android:numColumns="7"
android:horizontalSpacing="0dp" android:columnWidth="90dp"/>
```

[그림 22] content\_main 코드 4

- LinearLayout 의 orientation 속성을 horizontal 로 설정하고 월,화,수,목,금,토,일을 나열한다.

- MainActivity.java 에서 직접 만든 Calendar 를 넣을 GridView 를 설정한다.

### 3.3.2. writeday.xml

```
<Button android:text="2019-05-13" android:layout_gravity="fill_horizontal"
android:layout_columnSpan="2" android:layout_row="0" android:layout_column="0"/>

<Button android:text="기분" android:layout_row="0" android:layout_column="2"/>

<Button android:text="날씨" android:layout_row="0" android:layout_column="3"/>

<EditText android:layout_gravity="fill_horizontal" android:layout_columnSpan="3"
android:layout_row="1" android:layout_column="0" android:hint="제목을 입력하세요"/>

<Button android:text="분류" android:layout_gravity="fill_horizontal"
android:layout_row="1" android:layout_column="3"/>
```

[그림 23] writeday 코드 1

- 일기 작성 화면 상단 바에 날짜, 기분, 날씨, 분류 버튼을 설정한다.

```
<EditText android:layout_height="match_parent" android:layout_width="match_parent"
android:scrollbars="vertical" android:lines="12" android:inputType="textMultiLine"/>
```

[그림 24] writeday 코드 2

- 일기의 주 내용을 적는 부분은 EditText 위젯을 설정한다
- EditText 위젯의 lines 속성은 12 로, inputType 속성은 textMultiLine 으로 설정한다.

### 3.3.3. setting.xml

```
<TableLayout android:layout_height="wrap_content"
android:layout_width="match_parent" android:stretchColumns="*"
android:background="#999999" android:layout_weight="1">
```

```

        <TableRow android:layout_height="wrap_content"
android:layout_width="match_parent" android:background="#ffffff"
android:layout_margin="1dp">

<TextView android:layout_width="wrap_content" android:textSize="25sp"
android:text="분류" android:layout_gravity="center_vertical"/>

<Button android:text="추가"/>

</TableRow>

        <TableRow android:layout_height="52dp" android:layout_width="match_parent"
android:background="#ffffff" android:layout_marginBottom="1dp"
android:layout_marginRight="1dp" android:layout_marginLeft="1dp">

<TextView android:layout_width="wrap_content" android:textSize="25sp"
android:text="비밀번호 설정" android:layout_gravity="center_vertical"
android:layout_marginBottom="5dp" android:layout_marginTop="5dp"/>

<Switch android:layout_marginBottom="5dp" android:layout_marginTop="5dp"
android:checked="false" android:id="@+id/switchd"/>

</TableRow>

        <TableRow android:layout_height="wrap_content"
android:layout_width="match_parent" android:background="#ffffff"
android:layout_marginBottom="1dp" android:layout_marginRight="1dp"
android:layout_marginLeft="1dp">

<TextView android:layout_width="wrap_content" android:textSize="25sp"
android:text="비밀번호 변경" android:layout_gravity="center_vertical"/>

```



```

<Button android:text="변경"/>

</TableRow>

</TableLayout>

```

[그림 25] setting 코드 1

- TableLayout 을 사용하여 첫 번째 TableRow 에는 분류 추가 버튼, 두 번째 TableRow 에는 비밀번호 설정하기 위한 Switch 위젯과 비밀번호 변경 버튼을 설정한다.

### 3.3.4. bimil.xml

```

<LinearLayout android:layout_height="wrap_content"
android:layout_width="match_parent" android:orientation="horizontal"
android:id="@+id/ll_passcodes">

<EditText android:id="@+id/passcode_1" style="@style/PasscodeInputIStyle">

<requestFocus/>

</EditText>

<EditText android:id="@+id/passcode_2" style="@style/PasscodeInputIStyle"/>

<EditText android:id="@+id/passcode_3" style="@style/PasscodeInputIStyle"/>

<EditText android:id="@+id/passcode_4" style="@style/PasscodeInputIStyle"/>

</LinearLayout>

```

[그림 26] bimil 코드 1

- 비밀번호 버튼을 눌렀을 때 입력되는 EditText 4 개를 LinearLayout 을 horizontal 로 설정하여 수평으로 설정한다.
- EditText 의 style 은 PasscodeInputStyle 로 설정한다.

```

<TableLayout android:layout_height="match_parent"
android:layout_width="match_parent" android:id="@+id/tl_keys"
android:paddingTop="30dp" android:stretchColumns="*" android:shrinkColumns="*">

<TableRow>

<Button android:id="@+id/button1" style="@style/PasscodeButtonStyle"
android:text="1"/>

<Button android:id="@+id/button2" style="@style/PasscodeButtonStyle"
android:text="2"/>

<Button android:id="@+id/button3" style="@style/PasscodeButtonStyle"
android:text="3"/>

</TableRow>

.
.
<중간생략>
.
.

-<LinearLayout android:layout_height="match_parent"
android:layout_width="match_parent" android:orientation="horizontal">

<Button android:layout_height="match_parent" android:layout_width="175dp"
android:text="OK"/>

<Button android:layout_height="match_parent" android:layout_width="177dp"

```

```

android:text="Cancel"/>

</LinearLayout>

```

[그림 27] bimil 코드 2

- 비밀번호 입력 버튼은 TableLayout 안에 배치한다.
- 비밀번호 입력 버튼의 style 은 PasscodeButonStyle 로 설정한다.
- 비밀번호 화면 하단에 'OK'와 'Cancel' 버튼을 배치한다.

### 3.3.5. dlgform.xml

```

<LinearLayout android:orientation="vertical" android:id="@+id/layout_root"
android:layout_height="match_parent" android:layout_width="match_parent"
xmlns:tools="http://schemas.android.com/tools"
xmlns:android="http://schemas.android.com/apk/res/android">

<GridView android:id="@+id/gridView1" android:layout_height="wrap_content"
android:layout_width="match_parent" android:numColumns="4"
android:gravity="center"/>

</LinearLayout>

```

[그림 28] dlgform 코드 1

- GridView 를 numCloumns 수를 4 로 설정한다.

### 3.3.6 image.xml

```

<?xml version="1.0" encoding="UTF-8"?>

<android.support.constraint.ConstraintLayout android:layout_height="match_parent"
android:layout_width="match_parent"
xmlns:android="http://schemas.android.com/apk/res/android">

```

```
<ImageView android:layout_height="wrap_content" android:layout_width="wrap_content"
android:id="@+id/ivFace"/>

</android.support.constraint.ConstraintLayout>
```

[그림 29] image 코드 1

- 일기 작성 화면에서 기본 버튼에 넣을 ImageView 를 설정한다.

### 3.3.7 day.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<LinearLayout android:layout_height="match_parent" android:orientation="vertical"
android:layout_width="match_parent"
xmlns:android="http://schemas.android.com/apk/res/android">

<TextView android:layout_height="wrap_content" android:layout_width="match_parent"
android:textStyle="bold" android:id="@+id/day_tv"/>

<ImageView android:layout_height="match_parent" android:layout_width="match_parent"
android:id="@+id/day_img"/>

</LinearLayout>
```

[그림 30] day 코드 1

### 3.3.8 activity\_main\_drawer.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<menu tools:showIn="navigation_view" xmlns:tools="http://schemas.android.com/tools"
xmlns:android="http://schemas.android.com/apk/res/android">

<group android:checkableBehavior="single">
```

```

<item android:title="프로필 편집" android:icon="@drawable/ic_menu_camera"
android:id="@+id/nav_camera"/>

<item android:title="검색" android:icon="@drawable/ic_menu_gallery"
android:id="@+id/nav_gallery"/>

<item android:title="설정" android:icon="@drawable/ic_menu_manage"
android:id="@+id/nav_manage"/>

</group>

</menu>

```

[그림 31] activity\_main\_drawer 코드 1

- 메뉴의 Item 을 프로필 편집, 검색, 설정 총 3 가지로 구성하였다.

## 4. 시스템 설계서(Java)

### 4.1. WriteDay 클래스

writeday.xml 과 연결되는 클래스이다. writeday 화면에서 입력한 날짜, 제목, 날씨, 분류, 일기, 사진을 SD 카드에 생성한 mydir 파일에 저장한다.

WriteDay 클래스는 AppCompatActivity 클래스를 상속 받는다.

#### 4.1.1. 자바 객체, 변수 설명

```

Button btnSubmit, wdDate, wdFeel, wdWeather, wdCategory;
EditText wdTitle, wdContents;
View dialogView;

```

[그림 32] WriteDay 코드 1

- Button btnSubmit, wdDate, wdFeel, wdWeather, wdCategory 는 writeday.xml 의 저장, 날짜, 기분, 날씨, 분류 위젯과 대응하기 위한 객체이다.
- EditText wdTitle, wdContents 는 writeday.xml 의 제목과 일기내용의 위젯과 대응하기 위한 객체이다.

- View dialogView 는 각각의 버튼을 눌렀을 때 대화상자가 뜨게 하기 위한 객체이다.
- int selectYear, selectMonth, selectDay 는 wdDate 버튼 위젯을 눌렀을 때 CalendarView 위젯이 뜨는데 그 중 사용자가 선택한 연도, 월, 일을 저장하기 위한 객체이다.

```
private Integer[] mThumbIds = {
    R.drawable.bbackchim, R.drawable.boolman, R.drawable.danghwang,
    R.drawable.kk
    , R.drawable.love, R.drawable.sad, R.drawable.sm, R.drawable.uu, R.drawable.vv,
    R.drawable.why
    , R.drawable.zzang, R.drawable.wwww
};
private Integer[] mWeathers={
    R.drawable.boengae, R.drawable.hrim, R.drawable.onlywindow, R.drawable.rain, R.drawable.rain2,
    R.drawable.rainbow, R.drawable.rainbowrain
    , R.drawable.snow, R.drawable.sun, R.drawable.three, R.drawable.window, R.drawable.window2
};
```

[그림 33] WriteDay 코드 2

- 변수 mThumbIds 에는 기분 이미지를 넣는다.
- 변수 mWeathers 에는 날씨 이미지를 넣는다.

#### 4.1.2. 메서드 설명

```
btnSubmit = findViewById(R.id.btnSubmit);
wdDate = findViewById(R.id.wdDate);
wdFeel = findViewById(R.id.wdFeel);
wdWeather = findViewById(R.id.wdWeather);
wdTitle = findViewById(R.id.wdTitle);
```

```
wdCategory = findViewById(R.id.wdCategory);  
wdContents = findViewById(R.id.wdContents);
```

[그림 34] WriteDay 코드 3

- findViewById 메서드를 사용하여 writeday.xml 의 위젯과 자바에서 만든 객체를 대응시킨다.

#### 4.1.3. 버튼 클릭 이벤트 처리

##### 4.1.3.1. wdDate 버튼

```
wdDate.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        DatePickerDialog dlg = new DatePickerDialog(WriteDay.this, new  
        DatePickerDialog.OnDateSetListener() {  
            @Override  
            public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {  
                String msg = String.format("%d-%d-%d", year, month + 1, dayOfMonth);  
                wdDate.setText(msg);  
            }  
        }, cal.get(Calendar.YEAR), cal.get(Calendar.MONTH), cal.get(Calendar.DATE));  
  
        dlg.show();  
    }  
});
```

[그림 35] WriteDay 코드 4

- wdDate 버튼을 누를 시 날짜선택 calendarView 대화상자가 띄어진다.
- 사용자가 선택한 연도, 월, 일은 각각 selectYear, selectMonth, selectDay 객체에 저장된다.
- wdDate 버튼의 text 는 선택한 연도, 월, 일로 바뀐다.

#### 4.1.3.2. wdFeel 버튼

```
wdFeel.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        showDialog(CATEGORY_ID);  
    }  
});
```

[그림 36] WriteDay 코드 5

- wdFeel 버튼을 누를 시 showDialog 메서드가 실행된다.

#### 4.1.3.3. wdWeather 버튼

```
wdWeather.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        showDialog(CATEGORY_ID2);  
    }  
});
```

[그림 37] WriteDay 코드 6

- wdWeather 버튼을 누르면 showDialog 메서드가 실행된다.

#### 4.1.3.4. wdCategory 버튼

```
wdCategory.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
  
    }  
});
```

[그림 38] WriteDay 코드 7



#### 4.1.3.5. btnSubmit 버튼

```
btnSubmit.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        //등록버튼 누르면 DB 에 저장되고 메인화면으로 돌아감  
        showDialog 함수가 실행된다.  
  
        String SDpath = Environment.getExternalStorageDirectory().getAbsolutePath();  
        File myDir=new File(SDpath+"/mydir");  
        String filename = "file.txt";  
  
        if (!myDir.exists()) {  
            myDir.mkdir();  
        }  
  
        File file = new File(myDir, filename);  
        try {  
            FileOutputStream outFs = new FileOutputStream(file);  
            String strDate = wdDate.getText().toString();  
            String strFeel = wdFeel.getText().toString();  
            String strWeather = wdWeather.getText().toString();  
            String strTitle = wdTitle.getText().toString();  
            String strCategory = wdCategory.getText().toString();  
            String strContents = wdContents.getText().toString();  
            outFs.write((strDate + strFeel + strWeather + strTitle + strCategory +  
strContents).getBytes());  
            outFs.close();  
            Toast.makeText(getApplicationContext(), "file.txt 가 생성됨",  
Toast.LENGTH_SHORT).show();  
        } catch (IOException e) {  
        }  
    }  
});
```

```

        finish();
    }
});

```

[그림 39] WriteDay 코드 8

- btnSubmit 버튼을 누르면 대화상자를 이용하여 선택했던 날짜, 기분, 날씨, 분류의 값들과 제목, 일기 내용들이 SD 카드에 생성한 mydir 파일로 저장된다.

#### 4.1.4. 메서드 설명

```

protected Dialog onCreateDialog(int id){
    switch (id){
        case CATEGORY_ID:
            AlertDialog.Builder builder;
            Context mContext=this;

            LayoutInflater
            inflater=(LayoutInflater)mContext.getSystemService(LAYOUT_INFLATER_SERVICE);
            View layout = inflater.inflate(R.layout.dlgform,(ViewGroup)
            findViewById(R.id.layout_root));

            GridView gridView = layout.findViewById(R.id.gridView1);
            gridView.setAdapter(new ImageAdapter(this, CATEGORY_ID));
            gridView.setOnItemClickListener(new AdapterView.OnItemClickListener()
            {
                public void onItemClick(AdapterView parent, View v,int position, long id)
            {
                wdFeel=findViewById(R.id.wdFeel);
                wdFeel.setImageResource(mThumbIds[position]);
            }
            });

            builder = new AlertDialog.Builder(mContext);
            builder.setView(layout);

```

```

        builder.setPositiveButton("확인", null);
        dialog = builder.create();
        break;
    case CATEGORY_ID2:
        Context mContext2=this;

        LayoutInflater
inflater2=(LayoutInflater)mContext2.getSystemService(LAYOUT_INFLATER_SERVICE);
        View layout2 = inflater2.inflate(R.layout.dlgform,(ViewGroup)
findViewById(R.id.layout_root));
        GridView gridView2 = layout2.findViewById(R.id.gridView1);
        gridView2.setAdapter(new ImageAdapter(this,CATEGORY_ID2));
        gridView2.setOnItemClickListener(new AdapterView.OnItemClickListener()
        {
            public void onItemClick(AdapterView parent, View v, int position, long
id) {

                wdWeather=findViewById(R.id.wdWeather);
                wdWeather.setImageResource(mWeathers[position]);

            }
        });

        builder = new AlertDialog.Builder(mContext2);
        builder.setView(layout2);
        builder.setPositiveButton("확인", null);
        dialog = builder.create();
        break;
    default:
        dialog = null;
    }

    return dialog;
}

```

[그림 40] WriteDay 코드 10

- onCreateDialog 클래스에는 id 값이 CATEGORY\_ID(wdFeel)이면 GridView 의 객체를 dlgform.xml 파일의 GridView 위젯과 대응시킨다.
- wdFeel 에 mThumblds 배열에 있는 이미지들을 넣는다.
- id 값이 CATEGORY\_ID(wdWeather)이면 GridView 의 객체를 digform.xml 파일의 GridView 위젯과 대응시킨다.
- GridView 에 CATEGORY\_ID 의 ImageAdapter 를 넣는다.
- wWeather 에 mWeathers 배열에 있는 이미지들을 넣는다.

## 4.2. MainActivity 클래스

### 4.2.1 자바 변수, 객체 설명

```
public static int SUNDAY = 1;

private TextView mTvCalendarTitle;
private GridView mGvCalendar; Button bLastMonth,bNextMonth,btnWrite, btnsave,
btncancel;
    Toolbar toolbar;
    TabHost tabHost;
    DrawerLayout drawer;
    NavigationView navigationView;
    ListView listView;

    .
    .
    <중간 생략>
    .
    .

LinearLayout linear2, diaryLinear;
EditText et;
```

```
TextView tvCount,text1,text2;
private ArrayList<DayInfo> mDayList;
private CalendarAdapter mCalendarAdapter;
```

[그림 41] MainActivity 코드 1

- SUNDAY 변수에 1 값을 넣는다. SUNDAY 변수는 일요일을 뜻한다.
- activity\_main.xml 파일의 위젯들을 MainActivity.java 파일과 연결하기 위해서 각각의 위젯들의 객체를 생성한다.

```
toolbar = findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    tabHost=findViewById(R.id.Host);
    tabHost.setup();

    TabHost.TabSpec
tabSpecHome=tabHost.newTabSpec("HOME").setIndicator("HOME");
    tabSpecHome.setContent(R.id.tabHome);
    tabHost.addTab(tabSpecHome);

    TabHost.TabSpec tabSpecDate=tabHost.newTabSpec("Date").setIndicator("Date");
    tabSpecDate.setContent(R.id.tabDate);
    tabHost.addTab(tabSpecDate);
```

[그림 42] MainActivity 코드 2

- TabSpec 을 이용해서 탭 위젯을 가리키는 명칭들을 HOME 과 Date 로 지정한다.

#### 4.2.2 메서드 설명

```
protected void onActivityResult(int requestCode, int resultCode,
    @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    public void checkFunction(){}
}
```

```

void init(){}

    public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {}

    public void onBackPressed() {}

@Override
    public boolean onCreateOptionsMenu(Menu menu){}

    public boolean onNavigationItemSelected(MenuItem item) {}

    protected void onResume() {}

    private void getCalendar(Calendar calendar){}

    private Calendar getLastMonth(Calendar calendar){}

    private Calendar getNextMonth(Calendar calendar){}

    public void onItemClick(AdapterView<?> parent, View v, int position, long arg3){}

    public void onClick(View v){ }

    private void initCalendarAdapter(){ }

```

[그림 43] MainActivity 코드 3

- onActivityResult() 메서드는 Intent 로 전달받은 resultCode 가 RESULT\_OK 이면 init() 메서드를 호출한다.
- checkFunction() 메서드는 SD 카드 쓰기 권한을 설명하는 메서드이다.

- init() 메서드는 저장한 일기 작성 파일을 ListView 에 가져오는 메서드이다.
- onRequestPermissionsResult() 메서드는 사용자로부터 SD 카드 쓰기 권한 허용을 받는 메서드이다.
- onCreateOptionsMenu() 메서드는 menu.main 파일을 인플레이터하는 메서드이다.
- onNavigationItemSelectedListener() 메서드는 menu.main 파일의 item 을 클릭 시 각각의 화면들을 Intend 하여 불러오는 메서드이다.
- onResume() 메서드는 이번달 Calendar 인스턴스를 생성하는 메서드이다.
- getCalendar() 메서드는 달력 화면에 보여지는 이번달의 Calendar 객체를 생성하고 달력을 셋팅한다.
- getLastMonth()은 지난 달의 Calendar 객체를 반환하는 메서드이다.
- getNextMonth() 메서드는 다음 달의 Calendar 객체를 반환하는 메서드이다.
- onItemClickListener() 메서드는 달력 화면을 클릭하면 Intent 를 사용하여 wirteday 화면으로 전환된다.
- onClick() 메서드는 달력 화면의 이전 달 Button, 다음 달 Button 을 클릭시 발생하는 이벤트 처리 메서드이다.

### 4.3 CalendarAdapter 클래스

BaseAdapter를 상속받아 구현한 Calendar Adapter이다.

```
public class CalendarAdapter extends BaseAdapter
```

[그림 44] CalendarAdapter 코드 1

#### 4.3.1 자바 변수 설명

```
private ArrayList<DayInfo> mDayList;
private Context mContext;
private int mResource;
```

```

private LayoutInflater mLlInflater;
int width, height;

.
.
<중간 생략>
.
.

public CalendarAdapter(Context context, int textResource, ArrayList<DayInfo> dayList, int
width, int height)
{
    this.mContext = context;
    this.mDayList = dayList;
    this.mResource = textResource;
    this.mLlInflater = (LayoutInflater)
mContext.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    this.width = width;
    this.height = height;
}

```

[그림 45] CalendarAdapter 코드 2

- mDayList 변수는 날짜 저장 배열이다.
- mContext는 context를, mDayList는 날짜 정보가 들어있는 dayList를 mResource은 Layout Resource를 width와 height는 각각 display의 가로 세로 길이를 저장한다.

#### 4.3.2 메서드 설명

```

public int getCount(){}

public Object getItem(int position){}

public long getItemId(int position){}

public View getView(int position, View convertView, ViewGroup parent){}

```



```
private int getCellWidthDP(){}

private int getRestCellWidthDP(){}

private int getCellHeightDP(){}

```

[그림 46] CalendarAdapter 코드 3

- getCount 메서드는 mDayList.size()를 리턴하는 메서드이다.
- getItem 메서드는 mDayList.get(position)를 리턴하는 메서드이다.
- getView 메서드는 날짜 부분의 text 색과, display의 크기를 지정하는 메서드이다.
- getCellWidthDP(), getRestCellWidthDP(), getCellHeightDP() 메서드는 날짜가 입력되어져 있는 cell의 넓이, 높이를 지정하는 메서드이다.

#### 4.4. DayInfo 클래스

하루의 날짜정보를 저장하는 클래스이다.

##### 4.4.1. 자바 변수 설명

```
private String day;
private boolean inMonth;

```

[그림 47] DayInfo 코드 1

- 달력의 날짜와 이번 달에 대한 변수이다.

##### 4.4.2. 메서드 설명

```
public String getDay()
{
    return day;
}

```

[그림 48] DayInfo 코드 2

- 날짜를 반환하는 메서드이다.

```
public void setDay(String day)
{
    this.day = day;
}
```

[그림 49] DayInfo 코드 3

- 날짜를 저장하는 메서드이다.

```
public boolean isInMonth()
{
    return inMonth;
}
```

[그림 50] DayInfo 코드 4

- 이번 달의 날짜인지 정보를 반환한다.

```
public void setInMonth(boolean inMonth)
{
    this.inMonth = inMonth;
}
```

[그림 51] DayInfo 코드 5

- 이번 달의 날짜인지 정보를 저장한다.

## 4.5. setting 클래스

### 4.5.1 자바 객체 설명

```
Switch aSwitch;
```

[그림 42] setting 코드 1

- Setting.xml 에서 비밀번호 설정 Switch 위젯과 연결하기 위한 객체를 생성한다.

### 4.5.2 버튼 클릭 시 이벤트 처리

```

aSwitch.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener()
{
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        if(aSwitch.isChecked()==true){
            Intent intent=new Intent(getApplicationContext(), Bimil.class);
            startActivity(intent);
        }
    }
}

```

[그림 43] setting 코드 2

- Switch 위젯이 체크되어 있을 경우 Intent 를 사용하여 Bimil.class 로 넘어간다.

## 5.기타

### 5.1 개발 일정

단계	내용
앱 분석 4/15	분석하려는 앱을 설정하고 U기준으로 분석을 진행한다.
1차 코딩 (메인 화면, 달력) ~4/30	5개로 나눈 화면 중 메인 화면, 달력 화면을 코딩
2차 코딩 (일기 작성) ~5/10	일기작성 화면 코딩
3차 코딩 (설정, 비밀번호) ~5/20	설정, 비밀번호 화면 코딩
4차 코딩 (추가 기능) ~6/13	추가적으로 넣을 기능들 코딩
최종 보고서 작성 ~6/16	최종 보고서 작성
비밀번호 기능 등 코딩 6/20~	최종 발표 때 구현하지 못한 기능들을 코딩한다.

[그림 44] 개발 일정

## 5.2 업무 분담

역할 분담	
정연재	팀장, UI 분석, 기능 분석, 코딩
신다영	UI 분석, 기능 분석, 코딩

[그림 45] setting 업무 분담