**모바일 프로그래밍**

**(발바닥)**

**2017.6.18**

|  |  |  |
| --- | --- | --- |
| 학과 | 학번 | 이름 |
| 컴퓨터공학과 | 201433793 | 박주연 |
| 컴퓨터공학과 | 201433875 | 이홍주 |

gachon

**목 차**

[1. 나의 앱 만들기 3](#_Toc421510929)

[1.1 애플리케이션 개요 3](#_Toc421510930)

[1.2 애플리케이션 구성 3](#_Toc421510931)

[1.3 애플리케이션 실행화면 5](#_Toc421510932)

[1.4 데이터베이스 설계 8](#_Toc421510933)

[1.5 애플리케이션 구현 9](#_Toc421510934)

[2. 나의 앱 만들기 후기 20](#_Toc421510935)

# 나의 앱 만들기

## 애플리케이션 개요

* **어플리케이션 명**

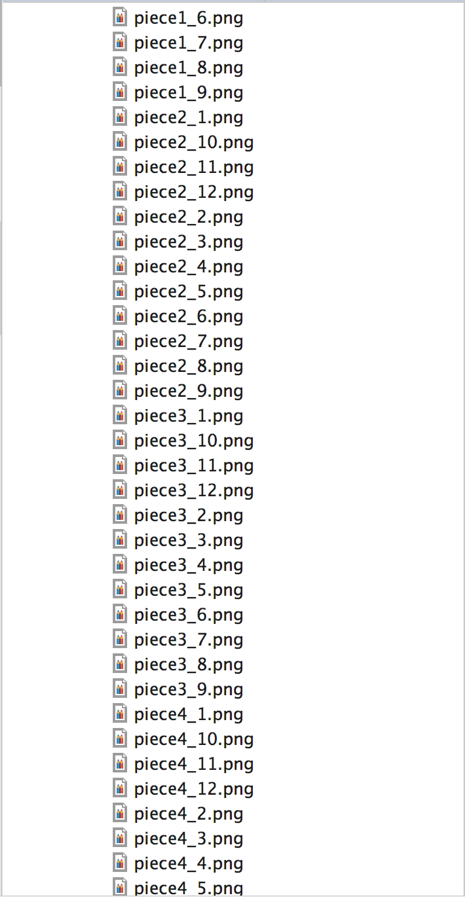
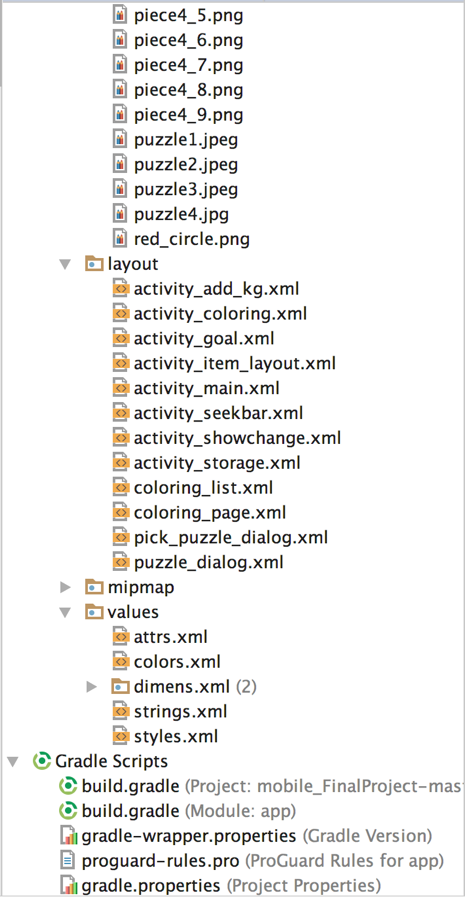
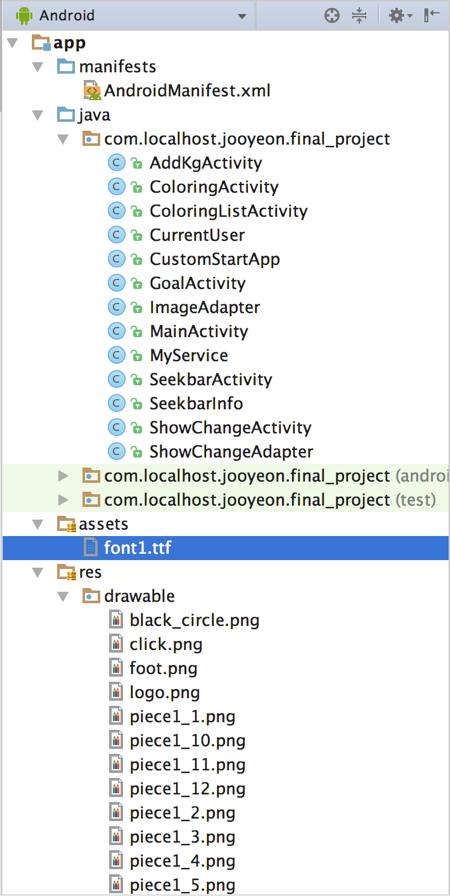
|  |
| --- |
| 발바닥 |

* **애플리케이션 요약**

|  |
| --- |
| 다이어트 어플리케이션은 다이어트를 하고 있는 사용자라면 한번쯤 설치하여 이용해보았을 정도로 보편적이고 대중적입니다. 대부분의 다이어트 어플리케이션은 사용자의 체중 변화를 보여주며, 체중 변화에 도움이 되는 운동 방법을 알려주기 위한 동영상으로 구성되어 있습니다.  ‘발바닥’은 기존의 어플리케이션과 달리 몸무게 수치가 아닌 사용자의 체형에 중점을 두어 매일매일 몸무게와 함께 사진을 찍어 저장하면 seekbar를 통해 그동안의 체형 변화를 한 눈에 볼 수 있습니다. 또한 다이어트에 도움이 되는 만보기 기능과, 만보기를 이용한 퍼즐이라는 게임적 요소를 추가시켜 지루할 수 있는 어플리케이션을 좀 더 즐겁게 이용할 수 있게 하였습니다.  주요 기능은  1. 목표 설정 : 어플리케이션 첫 실행 시 사용자의 이름, 목표 몸무게, 날짜를 설정할 수 있습니다.  2. 오늘의 몸무게 추가 : 오늘의 몸무게와 체형 사진을 찍어서 저장합니다.  3. 만보기 : 어플을 실행시킨 상태에서 핸드폰을 흔들면 만보기 횟수가 올라가고 그 횟수는 저장됩니다.  4. 몸무게 변화량 보기 : 그동안의 몸무게, 만보기 횟수 기록을 볼 수 있습니다.  5. 퍼즐 : 만보기 횟수가 100씩 올라갈 때마다 퍼즐 조각을 하나 뽑을 수 있고, 퍼즐에 퍼즐조각을 맞출 수 있습니다. |

## 애플리케이션 구성

* **프로젝트 구성도**



* **메인 화면 구성도**

******

* **실행 기능**

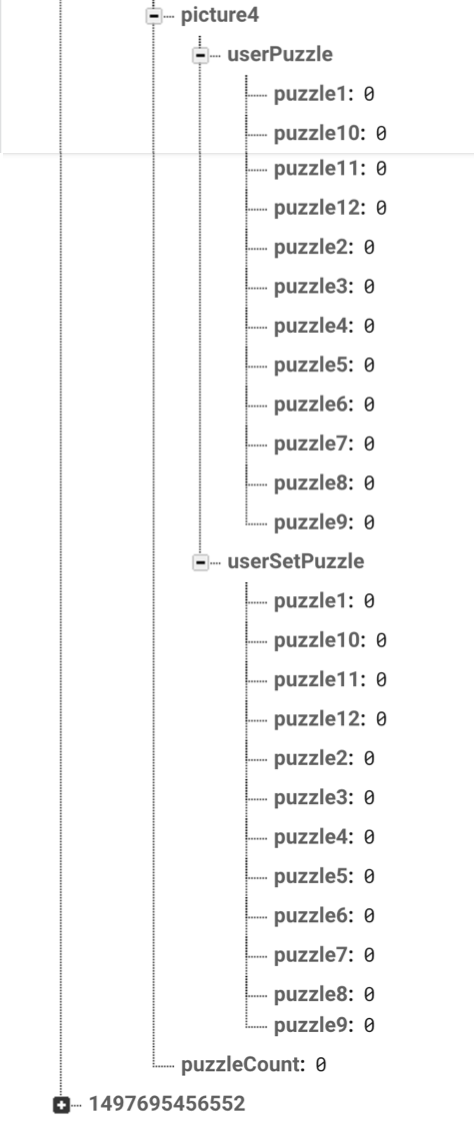
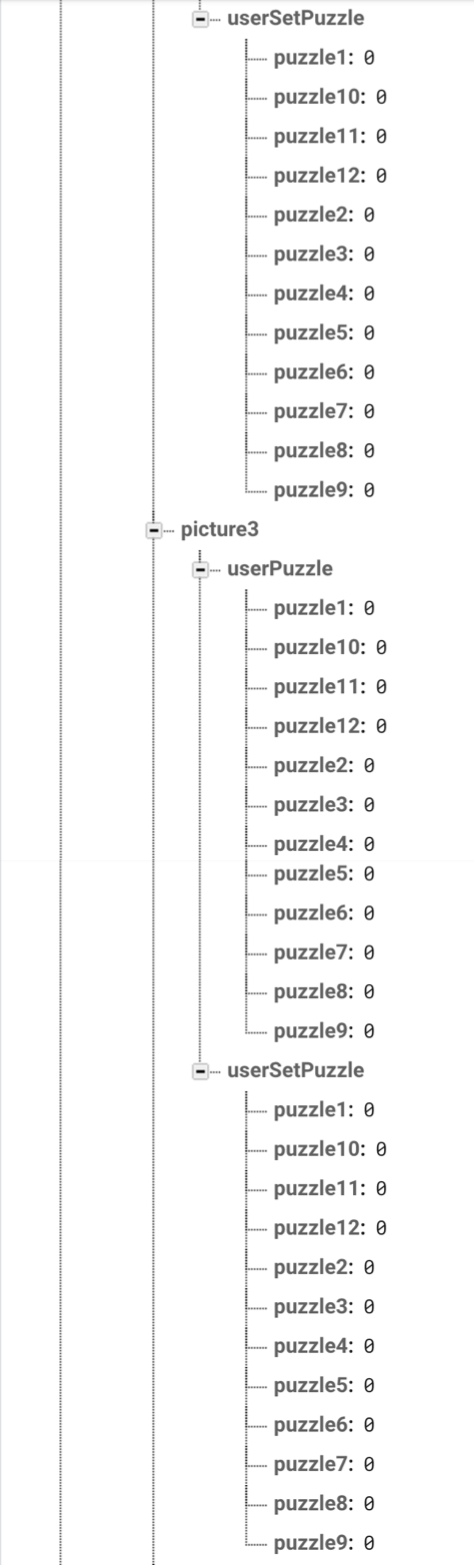
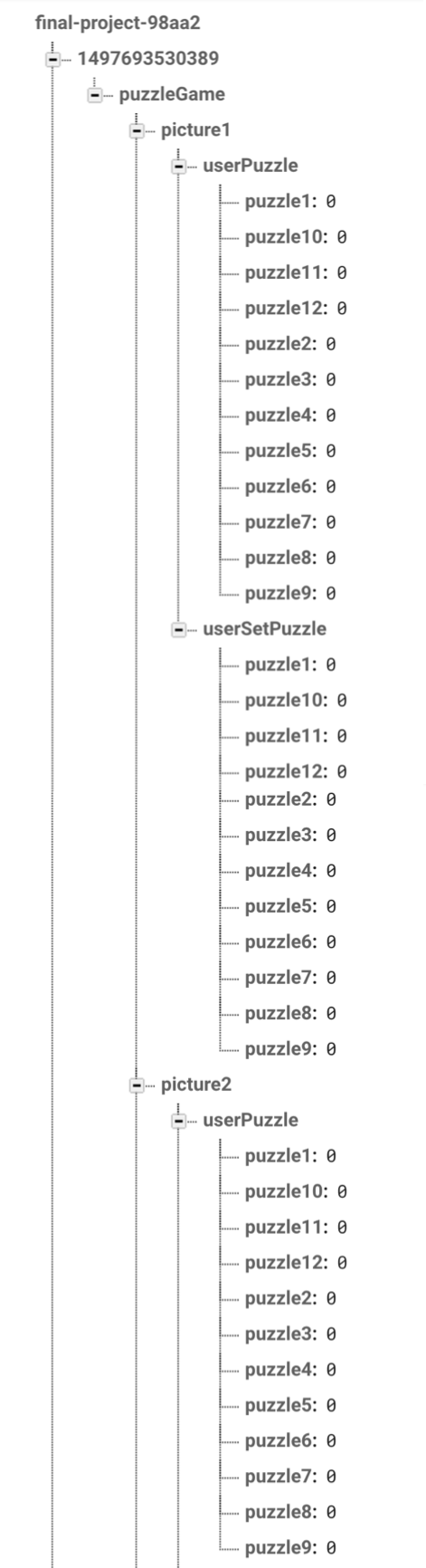
|  |  |  |
| --- | --- | --- |
| 주요기능 | 상세기능 | 비고 |
| 목표 설정  기능 | 어플리케이션 첫 실행 시 사용자의 이름, 목표 몸무게, 목표 기간을 설정한다. | 목표설정화면 |
| 만보기 | 사용자의 걸음에 따라 만보기의 횟수가 증가하고, 어플리케이션이 실행되지 않을 때에도 돌아간다. | 메인화면 |
| 오늘의 체형  사진찍기 | 오늘의 체형 사진을 찍으면 파이어베이스 storage에 저장되고, 그 uri는 파이어베이스 database에 저장된다. | Add Kg화면 |
| 오늘의 몸무게  저장하기 | 오늘의 몸무게를 저장하면 파이어베이스 database에 저장된다. | Add Kg화면 |
| 체형 변화  확인 | Seekbar를 통해 처음 시작부터 현재까지의 날짜, 사진, 몸무게를 파이어베이스 database에 불러와서 연속적으로 볼 수 있다. | Show Seekbar 화면 |
| 몸무게 수치 기록 확인 | 날짜별로 몸무게와 만보기 횟수를 파이어베이스 database에 불러와서 그동안의 기록을 볼 수 있다. | Show Change 화면 |
| 퍼즐조각 뽑기 | 만보기 횟수가 100회씩 올라갈 때마다 퍼즐조각을 무작위로 뽑을 수 있고, 이는 계속 저장된다. | Puzzle 화면 |
| 퍼즐조각  맞추기 | 사용자가 가지고 있는 퍼즐조각을 이용해서 원하는 도안의 퍼즐을 맞출 수 있다. | Puzzle 화면 |

## 애플리케이션 실행화면

|  |  |
| --- | --- |
| **화면1** | |
|  | |
| **목표 설정 화면** | |
| **화면2** | **화면3** |
| ***=*** |  |
| **만보기 및 메뉴 메인화면** | **Add Kg화면** |
| **화면4** | |
|  | |
| **체형 사진 촬영 화면** | |
| **화면5** | **화면6** |
|  |  |
| **오늘의 사진과 몸무게 저장 화면** | **체형 변화 확인 화면** |
| **화면7** | **화면8** |
|  |  |
| **처음 시작부터 현재까지의 기록 확인 화면** | **퍼즐 초기 화면** |
| **화면9** | |
|  | |
| **만보기 횟수가 100번 퍼즐 뽑기를 눌렀을 경우** | **만보기 횟수가 100번 넘고 퍼즐 뽑기를 눌렀을 경우** |
| **화면 10** | |
|  | |
| **사용자가 가지고 있는 퍼즐 조각을 이용해 퍼즐을 맞추는 화면** | |

## 데이터베이스 설계

* **파이어베이스 프로젝트 명 : Final Project**



## 애플리케이션 구현

1. **화면1 (목표 설정 화면)**

* **레이아웃 구현**

|  |  |
| --- | --- |
| 화면(UI) | 설명 |
|  | - 목표 설정 화면은 사용자의 이름, 목표 몸무게를 적을 수 있는 EditText와 목표기간을 설정할 수 있는 Dialog, Save 버튼으로 이루어져 있다.  - 목표 기간을 누를 경우 DatePicker Dialog가 뜨고, 날짜를 선택할 수 있다.  - 목표 설정 화면에서 저장하는 데이터는 모두 sharedReferences를 통해 저장된다. |
| activity\_goal.xml | |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout  android:id="@+id/activity\_goal"  xmlns:android="http://schemas.android.com/apk/res/android"  android:orientation="vertical" android:layout\_width="match\_parent"  android:layout\_height="match\_parent"**>  <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="0.5"  android:padding="16dp"**>  <**TextView  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:text="목표 설정하기"  android:textSize="20dp"  android:textStyle="bold"  android:gravity="center\_vertical"**/>  </**LinearLayout**>   *<!--이름-->* <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="1"  android:orientation="vertical"  android:background="#fec8c9"**>  <**TextView  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="1"  android:text="이름"  android:textSize="25dp"  android:gravity="center"  android:textColor="@android:color/white"  android:textStyle="bold"**/>  <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="1"  android:gravity="center"**>  <**EditText  android:id="@+id/goal\_nameEt"  android:layout\_width="150dp"  android:layout\_height="wrap\_content"  android:inputType="text"  android:textSize="20dp"  android:textAlignment="center"  android:textColor="@android:color/white"**/>  </**LinearLayout**>  </**LinearLayout**>   *<!--목표 몸무게-->* <**LinearLayout  android:id="@+id/goal\_kgLL"  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="1"  android:orientation="vertical"  android:background="#fd999a"**>  <**TextView  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="1"  android:text="목표 몸무게"  android:textSize="25dp"  android:gravity="center"  android:textColor="@android:color/white"  android:textStyle="bold"**/>  <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="1"  android:orientation="horizontal"  android:gravity="center"**>  <**EditText  android:id="@+id/goal\_kgET"  android:layout\_width="80dp"  android:layout\_height="wrap\_content"  android:textAlignment="center"  android:inputType="numberDecimal"  android:textSize="20dp"** />  <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:textSize="25dp"  android:text="kg"**/>  </**LinearLayout**>  </**LinearLayout**>   *<!--디데이설정-->* <**LinearLayout  android:id="@+id/goal\_dateLL"  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="1"  android:orientation="vertical"  android:background="#e85a71"**>  <**TextView  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="1"  android:text="목표 기간"  android:textSize="25dp"  android:gravity="center"  android:textColor="@android:color/white"  android:textStyle="bold"**/>  <**TextView  android:id="@+id/goal\_dateTv"  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="1"  android:textSize="20dp"  android:gravity="center"  android:textColor="@android:color/white"  android:text="xxxx년 x월 x일"**/>  </**LinearLayout**>   *<!--버튼-->* <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="0.5"**>  <**Button  android:id="@+id/goal\_saveBtn"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="#3e4149"  android:text="SAVE"  android:textSize="20dp"  android:textColor="@android:color/white"  android:textStyle="bold"**/>  </**LinearLayout**> </**LinearLayout**> | |

* **기능 구현**

|  |
| --- |
| **GoalActivity.java** |
| **package** hbapp.realfinalproject;  **import** android.app.AlertDialog; **import** android.content.Context; **import** android.content.DialogInterface; **import** android.content.Intent; **import** android.content.SharedPreferences; **import** android.graphics.Color; **import** android.os.Bundle; **import** android.support.annotation.Nullable; **import** android.support.v7.app.AppCompatActivity; **import** android.view.View; **import** android.view.WindowManager; **import** android.view.inputmethod.InputMethodManager; **import** android.widget.Button; **import** android.widget.EditText; **import** android.widget.LinearLayout; **import** android.widget.TextView;  **import** com.tsengvn.typekit.TypekitContextWrapper; **import** com.wdullaer.materialdatetimepicker.date.DatePickerDialog;  **import** java.util.Calendar;  **import** hbapp.realfinalproject.R;  */\*\*  \* Created by asmwj on 2017-05-23.  \*/   // 초기 화면, 목표를 설정하는 화면* **public class** GoalActivity **extends** AppCompatActivity **implements** DatePickerDialog.OnDateSetListener{   *// activity\_goal 뷰 정의* EditText **goal\_kgEt**;  EditText **goal\_nameEt**;  LinearLayout **goal\_dateLL**;  TextView **goal\_dateTv**;  Button **goal\_saveBtn**;  LinearLayout **activity\_goal**;   *// SharedPreferences 와 Editor 정의* **static** SharedPreferences *sharedPreferences*;  **static** SharedPreferences.Editor *editor*;    *// 오늘의 날짜를 가져 올 Calendar 변수와  // 날짜를 선택하는 Dialog* Calendar **calendar**;  DatePickerDialog **datePickerDialog**;  **int year**, **month**, **day**;    *// 사용자를 구분하는 user\_id* **long user\_id**;   @Override  **protected void** onCreate(@Nullable Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_goal***);   getWindow().setSoftInputMode(WindowManager.LayoutParams.***SOFT\_INPUT\_ADJUST\_PAN***);   *//SharedPreferences를 사용하기 위한 객체 가져오기  sharedPreferences* = getSharedPreferences(**"USER\_INFO"**, Context.***MODE\_PRIVATE***);  *editor* = *sharedPreferences*.edit();   *//isFirstRun 변수에 초기 false값 저장* **boolean** isFirstRun = *sharedPreferences*.getBoolean(**"isFirstRun"**,**false**);   *//isFirstRun이 true일 경우 곧바로 MainActivity 실행* **if**(isFirstRun==**true**) {  Intent newIntent = **new** Intent(GoalActivity.**this**, MainActivity.**class**);  startActivity(newIntent);  }  *//true로 변경  sharedPreferences*.edit().putBoolean(**"isFirstRun"**,**true**).apply();   *// id 값 부여* **activity\_goal** = (LinearLayout)findViewById(R.id.***activity\_goal***);  **goal\_nameEt** = (EditText)findViewById(R.id.***goal\_nameEt***);  **goal\_kgEt** = (EditText)findViewById(R.id.***goal\_kgET***);  **goal\_dateLL** = (LinearLayout)findViewById(R.id.***goal\_dateLL***);  **goal\_saveBtn** = (Button)findViewById(R.id.***goal\_saveBtn***);  **goal\_dateTv** = (TextView)findViewById(R.id.***goal\_dateTv***);   *// 키보드가 올라올 시 화면 누르면 키보드가 다시 숨겨지도록* **activity\_goal**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  hideKeyboard();  }  });   *// 오늘 날짜 가져오기* **calendar** = Calendar.*getInstance*();   **year** = **calendar**.get(Calendar.***YEAR***);  **month** = **calendar**.get(Calendar.***MONTH***);  **day** = **calendar**.get(Calendar.***DAY\_OF\_MONTH***);   *// user\_id에 현재 시간을 millisecond로 저장* **user\_id** = System.*currentTimeMillis*();   *// 목표기간을 눌렀을 경우 datepickerdialog가 나타나도록* **goal\_dateLL**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **datePickerDialog** = DatePickerDialog.newInstance(GoalActivity.**this**, **year**, **month**, **day**);  **datePickerDialog**.setThemeDark(**false**);  **datePickerDialog**.showYearPickerFirst(**false**);  **datePickerDialog**.setAccentColor(Color.*parseColor*(**"#009688"**));  **datePickerDialog**.setTitle(**"목표 기간을 선택해주세요."**);  **datePickerDialog**.show(getFragmentManager(), **"DatePickerDialog"**);  }  });  }   *// 날짜 선택 시* @Override  **public void** onDateSet(DatePickerDialog view, **final int** year, **final int** monthOfYear, **final int** dayOfMonth) {  String date = year +**"년 "**+ Integer.*valueOf*(monthOfYear+1) + **"월 "** + dayOfMonth + **"일"**;   **goal\_saveBtn**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {   **final** String name = **goal\_nameEt**.getText().toString();  **final float** goal\_kg = Float.*valueOf*(**goal\_kgEt**.getText().toString());   *// 이름과 몸무게를 입력하지 않았을 경우* **if** (**goal\_nameEt**.getText().equals(**""**)|| **goal\_kgEt**.getText().equals(**""**)){  AlertDialog.Builder builder2 = **new** AlertDialog.Builder(GoalActivity.**this**)  .setMessage(**"이름을 입력해주세요."**)  .setPositiveButton(**"확인"**, **new** DialogInterface.OnClickListener() {  @Override  **public void** onClick(DialogInterface dialog, **int** which) {  dialog.cancel();  }  });  builder2.show();  }   *// 몸무게에 터무니 없는 값을 넣었을 경우* **else if**(goal\_kg > 200){  AlertDialog.Builder builder3 = **new** AlertDialog.Builder(GoalActivity.**this**)  .setMessage(**"몸무게 다시 정해라."**)  .setPositiveButton(**"알겠어"**, **new** DialogInterface.OnClickListener() {  @Override  **public void** onClick(DialogInterface dialog, **int** which) {  dialog.cancel();  }  });  builder3.show();  }   **else**{  *// save 버튼 누를 시에 dialog 확인 창* AlertDialog.Builder builder = **new** AlertDialog.Builder(GoalActivity.**this**)  .setMessage(year +**"년 "**+ Integer.*valueOf*(monthOfYear+1) + **"월 "** + dayOfMonth + **"일까지"** + goal\_kg + **"kg 가능하니?"**)  .setPositiveButton(**"당연하지!"**, **new** DialogInterface.OnClickListener() {  @Override  **public void** onClick(DialogInterface dialog, **int** which) {   Intent i = **new** Intent(GoalActivity.**this**, MainActivity.**class**);  startActivity(i);   *// SharedPreferences editor 에 이름, 목표몸무게, 목표날짜 데이터를 put한다.  editor*.putString(**"name"**, name);  *editor*.putFloat(**"kg"**, goal\_kg);  *editor*.putInt(**"goal\_year"**, year);  *editor*.putInt(**"goal\_monthOfYear"**, monthOfYear+1);  *editor*.putInt(**"goal\_dayOfMonth"**, dayOfMonth);  *editor*.putLong(**"user\_id"**, **user\_id**);   *editor*.commit();   finish();  }  })  .setNegativeButton(**"모르겠어"**, **new** DialogInterface.OnClickListener() {  @Override  **public void** onClick(DialogInterface dialog, **int** which) {  dialog.cancel();  }  });  builder.show();  }  }  });  **goal\_dateTv**.setText(date);  }   *// 키보드를 숨기는 메소드* **private void** hideKeyboard() {  InputMethodManager inputManager = (InputMethodManager) getSystemService(Context.***INPUT\_METHOD\_SERVICE***);  inputManager.hideSoftInputFromWindow(**this**.getCurrentFocus().getWindowToken(), InputMethodManager.***HIDE\_NOT\_ALWAYS***);  } *//hideKeyboard() 끝   // 글씨체* @Override  **protected void** attachBaseContext(Context newBase) {  **super**.attachBaseContext(TypekitContextWrapper.wrap(newBase));  } } |

1. **화면2**

* **레이아웃 구현**

|  |  |
| --- | --- |
| 화면(UI) | 설명 |
|  | - activity\_main은 발바닥 어플리케이션의 가장 초기 화면으로 4개의 텍스트뷰와 4개의 버튼으로 구성되어 있습니다.  - 가장 상단에 위치한 텍스트뷰는 사용자에 대한 환영 인사말을 담고 있습니다.  -수평으로 위치한 두 개의 텍스트뷰는 다이어트 목표 날짜까지의 D-day를 계산하여 알려줍니다.  -화면의 중앙에 위치한 텍스트뷰는 사용자의 걸음 횟수를 나타내는 것으로 실시간으로 변합니다.  -ADD KG 버튼은 매일 사용자가 자신의 체형을 찍을 수 있는 사진과 몸무게를 입력할 수 있는 화면으로 이동하는 버튼입니다.  -SHOW SEEKBAR 버튼은 지금까지 사용자가 찍은 사진과 입력한 몸무게를 시크바를 이용하여 체형의 변화량을 확인하는 기능의 화면으로 이동하는 버튼입니다.  -SHOW CHANGE 버튼은 날짜별로 그 날의 사용자가 입력한 자신의 몸무게, 그리고 어플리케이션으로 측정된 걸음 횟수를 보여주는 화면으로 이동하는 버튼입니다. |
| activity\_main.xml | |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:orientation="vertical" android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="#FFFFFF"**>  <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="2"  android:orientation="vertical"**>  <**TextView  android:id="@+id/welcomeUserTv"  android:layout\_margin="20dp"  android:layout\_weight="1"  android:textStyle="bold"  android:textSize="17dp"  android:gravity="center"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"** />  <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:background="@android:color/darker\_gray"**/>  *<!--D day-->* <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal"  android:layout\_weight="1"  android:layout\_margin="10dp"**>  <**TextView  android:gravity="center\_vertical"  android:text="목표 날짜까지"  android:textStyle="bold"  android:textSize="20dp"  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"** />  <**TextView  android:gravity="center\_vertical"  android:id="@+id/DdayTv"  android:textStyle="bold"  android:textSize="20dp"  android:textAlignment="center"  android:textColor="@android:color/holo\_red\_light"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"** />  </**LinearLayout**>  </**LinearLayout**>    <**TextView  android:textColor="#000000"  android:id="@+id/cntView"  android:text="기본 텍스트"  android:textSize="50dp"  android:textStyle="bold"  android:gravity="center"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1"**/>  *<!-- 버튼-->* <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="4"  android:orientation="vertical"**>   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:orientation="horizontal"**>   <**Button  android:id="@+id/goto\_add\_kg"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:background="#fe4365"  android:text="Add KG"  android:textColor="@android:color/white"  android:textSize="20dp"  android:textStyle="bold"** />   <**Button  android:id="@+id/goto\_show\_seekbar"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:background="#fc9d9a"  android:text="Show Seekbar"  android:textColor="@android:color/white"  android:textSize="20dp"  android:textStyle="bold"** />  </**LinearLayout**>   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_weight="1"  android:layout\_height="match\_parent"  android:orientation="horizontal"**>  <**Button  android:id="@+id/goto\_show\_change"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="#f9cdad"  android:layout\_weight="1"  android:text="Show Change"  android:textColor="@android:color/white"  android:textSize="20dp"  android:textStyle="bold"** />   <**Button  android:id="@+id/goto\_puzzle\_game"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="#febb"  android:layout\_weight="1"  android:text="Puzzle"  android:textColor="@android:color/white"  android:textSize="20dp"  android:textStyle="bold"** />  </**LinearLayout**>  </**LinearLayout**> </**LinearLayout**> | |

* **기능 구현**

|  |
| --- |
| **MainActivity.java** |
| **package** com.localhost.jooyeon.final\_project;  **import** android.app.Activity; **import** android.content.Context; **import** android.content.Intent; **import** android.hardware.Sensor; **import** android.hardware.SensorManager; **import** android.os.Bundle; **import** android.view.View; **import** android.widget.Button; **import** android.widget.TextView; **import** android.widget.Toast;  **import** com.google.firebase.database.DataSnapshot; **import** com.google.firebase.database.DatabaseError; **import** com.google.firebase.database.DatabaseReference; **import** com.google.firebase.database.FirebaseDatabase; **import** com.google.firebase.database.ValueEventListener; **import** com.tsengvn.typekit.TypekitContextWrapper;  **import** java.text.DateFormat; **import** java.text.SimpleDateFormat; **import** java.util.Calendar; **import** java.util.Date;  **import static** com.localhost.jooyeon.final\_project.GoalActivity.*sharedPreferences*;   **public class** MainActivity **extends** Activity {   String **user\_id**;  **public static int** *cnt* = 0;   **private** TextView **tView**;   **private long lastTime**;  **private float speed**;  **private float lastX**;  **private float lastY**;  **private float lastZ**;  **private float x**, **y**, **z**;  **boolean flag** = **false**;   **private static final int *SHAKE\_THRESHOLD*** = 2800;  **private static final int *DATA\_X*** = SensorManager.***DATA\_X***;  **private static final int *DATA\_Y*** = SensorManager.***DATA\_Y***;  **private static final int *DATA\_Z*** = SensorManager.***DATA\_Z***;   **private** SensorManager **sensorManager**;  **private** Sensor **accelerormeterSensor**;   Long **puzzleCount**;   *//파이어베이스 연동 및 레퍼런스* DatabaseReference **mRootRef** = FirebaseDatabase.*getInstance*().getReference();  DatabaseReference **mDateRef**;  DatabaseReference **mPuzzleCountRef**;   **private final long FINSH\_INTERVAL\_TIME** = 2000; *//2초* **private long backPressedTime** = 0;   Button **showchangeBtn**;  Button **showseekbarBtn**;  Button **addkgBtn**;   TextView **ddayTv**;  TextView **welcomeuserTv**;   String **user\_name**;  Float **goal\_kg**;  **int goal\_year**, **goal\_monthOfYear**, **goal\_dayOfMonth**;   String **todayDate**;   *// D-day 변수* **static int** *tyear*, *tmonthOfYear*, *tdayOfMonth*;  **long today**, **dday**, **goal**;  **int resultNumber** = 1;   **static** FirebaseDatabase *database*;  DatabaseReference **myRef**;   String **check**;   **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);   setContentView(R.layout.***activity\_main***);  **sensorManager** = (SensorManager) getSystemService(***SENSOR\_SERVICE***);  **accelerormeterSensor** = **sensorManager**.getDefaultSensor(Sensor.***TYPE\_ACCELEROMETER***);   Button puzzleButton = (Button)findViewById(R.id.***goto\_puzzle\_game***);  **tView** = (TextView) findViewById(R.id.***cntView***);   **tView**.setText(**""** + *cnt*);   *//퍼즐 액티비티로 이동* puzzleButton.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  Intent intent = **new** Intent(getApplicationContext(), ColoringListActivity.**class**);  startActivity(intent);  }  });   startService(**new** Intent(MainActivity.**this**,MyService.**class**));  Date date = **new** Date();  DateFormat df = **new** SimpleDateFormat(**"yyyy-MM-dd"**);  **todayDate** = df.format(date);   *sharedPreferences* = getSharedPreferences(**"USER\_INFO"**, ***MODE\_PRIVATE***);  **user\_id** = String.*valueOf*(*sharedPreferences*.getLong(**"user\_id"**, 0));   *database* = FirebaseDatabase.*getInstance*();  **myRef** = *database*.getReference(**user\_id**);  **check** = **myRef**.child(**"date"**).getKey().toString();   **mDateRef** = **mRootRef**.child(**user\_id**+**"/date"**);  **mPuzzleCountRef** = **mRootRef**.child(**user\_id**+**"/puzzleGame/puzzleCount"**);   Toast.*makeText*(**this**, **check**, Toast.***LENGTH\_SHORT***);   **showchangeBtn** = (Button)findViewById(R.id.***goto\_show\_change***);  **showseekbarBtn** = (Button)findViewById(R.id.***goto\_show\_seekbar***);  **addkgBtn** = (Button)findViewById(R.id.***goto\_add\_kg***);  **ddayTv** = (TextView)findViewById(R.id.***DdayTv***);  **welcomeuserTv** = (TextView)findViewById(R.id.***welcomeUserTv***);   **showchangeBtn**.setOnClickListener(**gotolistener**);  **showseekbarBtn**.setOnClickListener(**gotolistener**);  **addkgBtn**.setOnClickListener(**gotolistener**);    **user\_name** = *sharedPreferences*.getString(**"name"**, **""**);  **goal\_kg** = *sharedPreferences*.getFloat(**"kg"**, 1);  **goal\_year** = *sharedPreferences*.getInt(**"goal\_year"**, 0);  **goal\_monthOfYear** = *sharedPreferences*.getInt(**"goal\_monthOfYear"**, 0);  **goal\_dayOfMonth** = *sharedPreferences*.getInt(**"goal\_dayOfMonth"**, 0);   String welcome = **"반갑습니다! "**+**user\_name**+**"님"**;  **welcomeuserTv**.setText(welcome);   *//Dday 표시  // 오늘 연, 월, 일 저장* Calendar c = Calendar.*getInstance*();  *tyear* = c.get(Calendar.***YEAR***);  *tmonthOfYear* = c.get(Calendar.***MONTH***)+1;  *tdayOfMonth* = c.get(Calendar.***DAY\_OF\_MONTH***);  c.set(*tyear*, *tmonthOfYear*, *tdayOfMonth*);   *// 목표 연, 월, 일 세팅* Calendar c2 = Calendar.*getInstance*();  c2.set(**goal\_year**, **goal\_monthOfYear**, **goal\_dayOfMonth**);   **today** = c.getTimeInMillis(); *// 오늘 날짜를 밀리타임으로 바꿈* **goal** = c2.getTimeInMillis(); *// 목표 날짜를 밀리타임으로 바꿈* **dday** = (**goal**-**today**)/(24\*60\*60\*1000); *// 목표 날짜에서 오늘날짜를 뺀 것을 '일' 단위로 바꿈* **resultNumber** = (**int**)**dday** + 1;   updateDisplay();  }   *// gotolistener 정의* View.OnClickListener **gotolistener** = **new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **switch** (v.getId()){  **case** R.id.***goto\_show\_change***:  Intent showChange\_i = **new** Intent(MainActivity.**this**, ShowChangeActivity.**class**);  startActivity(showChange\_i);  **break**;   **case** R.id.***goto\_show\_seekbar***:  Intent seekbar\_i = **new** Intent(MainActivity.**this**, SeekbarActivity.**class**);  startActivity(seekbar\_i);  **break**;   **case** R.id.***goto\_add\_kg***:  Intent addkg\_i = **new** Intent(MainActivity.**this**, AddKgActivity.**class**);  startActivity(addkg\_i);  **break**;  }  }  }; *//gotolistener 끝   //디데이 날짜가 오늘날짜보다 뒤에오면 '-', 앞에오면 '+'를 붙인다* **private void** updateDisplay(){  **if**(**resultNumber**>=0){  **ddayTv**.setText(String.*format*(**"D-%d"**, **resultNumber**));  }  **else**{  **int** absR=Math.*abs*(**resultNumber**);  **ddayTv**.setText(String.*format*(**"D+%d"**, absR));  }  } *//updateDisplay() 끝   // 뒤로 가기 버튼을 눌렀을 경우* @Override  **public void** onBackPressed() {  **long** tempTime = System.*currentTimeMillis*();  **long** intervalTime = tempTime - **backPressedTime**;   **if** (0 <= intervalTime && **FINSH\_INTERVAL\_TIME** >= intervalTime) {  moveTaskToBack(**true**);  finish();  android.os.Process.*killProcess*(android.os.Process.*myPid*());  } **else** {  **backPressedTime** = tempTime;  Toast.*makeText*(getApplicationContext(), **"\'뒤로\' 버튼을 한 번 더 누르시면 종료됩니다."**,  Toast.***LENGTH\_SHORT***).show();  }  } *// onBackPressed()끝* @Override  **public void** onStart() {  **super**.onStart();   *//흔들감지 여부 확인 후 있을 경우    //오늘의 날짜 저장* Calendar today = Calendar.*getInstance*();  SimpleDateFormat sdf = **new** SimpleDateFormat(**"yyyy-MM-dd"**);  String day = sdf.format(today.getTime());   *//파이어베이스에 저장된 오늘 걸은 수를 가져옴* **mDateRef**.child(day+**"/walkingNum"**).addValueEventListener(**new** ValueEventListener() {  @Override  **public void** onDataChange(DataSnapshot dataSnapshot) {  Long walkingNum = (Long)dataSnapshot.getValue();  **if**(walkingNum!=**null**){  *cnt*=walkingNum.intValue();  **tView**.setText(Integer.*toString*(*cnt*));  }  }   @Override  **public void** onCancelled(DatabaseError databaseError) {   }  });   *//파이어베이스에 저장된 퍼즐 뽑기 여부를 확인할 횟수를 가져옴* **mPuzzleCountRef**.addValueEventListener(**new** ValueEventListener() {  @Override  **public void** onDataChange(DataSnapshot dataSnapshot) {  **puzzleCount** = (Long)dataSnapshot.getValue();  **if**(**puzzleCount**==**null**){  **mPuzzleCountRef**.setValue(0);  **for**(**int** i = 1 ; i <= 4 ; i++) {  **for**(**int** j = 1 ; j <= 12 ; j++) {  **mPuzzleCountRef**.getParent().child(**"picture"**+i+**"/userPuzzle/puzzle"**+j).setValue(0);  **mPuzzleCountRef**.getParent().child(**"picture"**+i+**"/userSetPuzzle/puzzle"**+j).setValue(0);  }  }  }  }   @Override  **public void** onCancelled(DatabaseError databaseError) {   }  });  }   @Override  **protected void** attachBaseContext(Context newBase) {  **super**.attachBaseContext(TypekitContextWrapper.*wrap*(newBase));  } } |

|  |
| --- |
| **MyService.java** |
| **package** com.localhost.jooyeon.final\_project;  **import** android.app.Service; **import** android.content.Intent; **import** android.hardware.Sensor; **import** android.hardware.SensorEvent; **import** android.hardware.SensorEventListener; **import** android.hardware.SensorManager; **import** android.os.IBinder;  **import** com.google.firebase.database.DataSnapshot; **import** com.google.firebase.database.DatabaseError; **import** com.google.firebase.database.DatabaseReference; **import** com.google.firebase.database.FirebaseDatabase; **import** com.google.firebase.database.ValueEventListener;  **import** java.text.SimpleDateFormat; **import** java.util.Calendar;  **import static** com.localhost.jooyeon.final\_project.GoalActivity.*sharedPreferences*;  **public class** MyService **extends** Service **implements** SensorEventListener {   String **user\_id**;   **int cnt**;  **private long lastTime**;  **private float speed**;  **private float lastX**;  **private float lastY**;  **private float lastZ**;  **private float x**, **y**, **z**;  **boolean flag** = **false**;   **private static final int *SHAKE\_THRESHOLD*** = 2800;  **private static final int *DATA\_X*** = SensorManager.***DATA\_X***;  **private static final int *DATA\_Y*** = SensorManager.***DATA\_Y***;  **private static final int *DATA\_Z*** = SensorManager.***DATA\_Z***;   **private** SensorManager **sensorManager**;  **private** Sensor **accelerormeterSensor**;   *//파이어베이스 연동 및 레퍼런스* DatabaseReference **mRootRef** = FirebaseDatabase.*getInstance*().getReference();  DatabaseReference **mDateRef**;  DatabaseReference **mPuzzleCountRef**;   Long **puzzleCount**;   **public** MyService() {  }   @Override  **public** IBinder onBind(Intent intent) {  *// Service 객체와 (화면단 Activity 사이에서)  // 통신(데이터를 주고받을) 때 사용하는 메서드  // 데이터를 전달할 필요가 없으면 return null;* **return null**;  }  @Override  **public void** onCreate() {  **super**.onCreate();  *// 서비스에서 가장 먼저 호출됨(최초에 한번만)* }  @Override  **public int** onStartCommand(Intent intent, **int** flags, **int** startId) {  *// 서비스가 호출될 때마다 실행* **sensorManager** = (SensorManager) getSystemService(***SENSOR\_SERVICE***);  **accelerormeterSensor** = **sensorManager**.getDefaultSensor(Sensor.***TYPE\_ACCELEROMETER***);   *sharedPreferences* = getSharedPreferences(**"USER\_INFO"**, ***MODE\_PRIVATE***);  **user\_id** = String.*valueOf*(*sharedPreferences*.getLong(**"user\_id"**, 0));   *//파이어베이스 레퍼런스 등록* **mDateRef** = **mRootRef**.child(**user\_id**+**"/date"**);  **mPuzzleCountRef** = **mRootRef**.child(**user\_id**+**"/puzzleGame/puzzleCount"**);    **if** (**accelerormeterSensor** != **null**)  **sensorManager**.registerListener(**this**, **accelerormeterSensor**,  SensorManager.***SENSOR\_DELAY\_GAME***);   *//사용자가 처음 어플을 실행하였을 경우 레퍼런스로 참조가능한 데이터가 없기 때문에 이에 대한 값을 세팅* **mPuzzleCountRef**.addValueEventListener(**new** ValueEventListener() {  @Override  **public void** onDataChange(DataSnapshot dataSnapshot) {  **puzzleCount** = (Long)dataSnapshot.getValue();  **if**(**puzzleCount**==**null**){  **mPuzzleCountRef**.setValue(0);  **for**(**int** i = 1 ; i <= 4 ; i++) {  **for**(**int** j = 1 ; j <= 12 ; j++) {  **mPuzzleCountRef**.getParent().child(**"picture"**+i+**"/userPuzzle/puzzle"**+j).setValue(0);  **mPuzzleCountRef**.getParent().child(**"picture"**+i+**"/userSetPuzzle/puzzle"**+j).setValue(0);  }  }  }  }   @Override  **public void** onCancelled(DatabaseError databaseError) {   }  });   Calendar today = Calendar.*getInstance*();  SimpleDateFormat sdf = **new** SimpleDateFormat(**"yyyy-MM-dd"**);  String day = sdf.format(today.getTime());   *//백그라운드 상태에서도 만보기 값이 증가* **mDateRef**.child(day+**"/walkingNum"**).addValueEventListener(**new** ValueEventListener() {  @Override  **public void** onDataChange(DataSnapshot dataSnapshot) {  Long walkingNum = (Long)dataSnapshot.getValue();  **if**(walkingNum!=**null**){  **cnt**=walkingNum.intValue();  }  }   @Override  **public void** onCancelled(DatabaseError databaseError) {   }  });   **return super**.onStartCommand(intent, flags, startId);  }   @Override  **public void** onDestroy() {  **super**.onDestroy();   }   @Override  **public void** onSensorChanged(SensorEvent event) {   *//x, y, z값의 변화를 이용하여 사용자의 만보기 횟수 및 퍼즐 뽑기 여부를 확인할 포인트를 증가시킴* **if** (event.**sensor**.getType() == Sensor.***TYPE\_ACCELEROMETER***) {  **long** currentTime = System.*currentTimeMillis*();  **long** gabOfTime = (currentTime - **lastTime**);  **if** (gabOfTime > 100) {  **lastTime** = currentTime;  **x** = event.**values**[***DATA\_X***];  **y** = event.**values**[***DATA\_Y***];  **z** = event.**values**[***DATA\_Z***];   **speed** = ((Math.*abs*(**x** + **y** + **z** - **lastX** - **lastY** - **lastZ**) / gabOfTime \* 10000));   **if** (**speed** > ***SHAKE\_THRESHOLD***) {  **if**(**flag**) {  Calendar today = Calendar.*getInstance*();  SimpleDateFormat sdf = **new** SimpleDateFormat(**"yyyy-MM-dd"**);  String day = sdf.format(today.getTime());  **mDateRef**.child(day+**"/walkingNum"**).setValue(++**cnt**);  **if**(**puzzleCount**!=**null**) {  **mPuzzleCountRef**.setValue((++**puzzleCount**));  }**else**{  **mPuzzleCountRef**.setValue(1);  }   **flag**=**false**;  }  }   **lastX** = event.**values**[***DATA\_X***];  **lastY** = event.**values**[***DATA\_Y***];  **lastZ** = event.**values**[***DATA\_Z***];  **flag** = **true**;  }   }   }   @Override  **public void** onAccuracyChanged(Sensor sensor, **int** accuracy) {   } } |

1. **화면 3, 화면 4, 화면 5 (Add Kg 화면)**

* **레이아웃 구현**

|  |  |
| --- | --- |
| 화면(UI) | 설명 |
|  | - Add Kg 화면은 사진을 찍거나, 찍은 사진을 띄우는 ImageView 와 현재 몸무게를 저장하는 EditText, 이를 저장하는 save 버튼으로 이루어져 있다.  - 묵시적 Intent를 이용해서 어플에 설치되어 있는 카메라를 실행시키고, 찍힌 사진의 uri값을 이용해 Glide 오픈 Api로 ImageView에 사진을 띄웠다.  - save 버튼을 누르면 사진의 uri와 몸무게가 Firebase database에 저장이 된다. |
| activity\_add\_kg.xml | |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout  android:id="@+id/add\_kgLL"  xmlns:android="http://schemas.android.com/apk/res/android"  android:orientation="vertical"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"**>  <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="2"  android:gravity="center"**>  <**TextView  android:id="@+id/add\_kgTv"  android:text="현재 너의 몸무게는..?"  android:inputType="numberDecimal"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:textSize="25dp"  android:textStyle="bold"**/>  </**LinearLayout**>   *<!--사진 찍기-->* <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="7.5"  android:gravity="center"  android:orientation="vertical"**>  <**ImageView  android:id="@+id/add\_kgIv"  android:src="@drawable/click"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"**/>  </**LinearLayout**>   *<!--현재 몸무게-->* <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="2"  android:orientation="horizontal"  android:gravity="center"**>  <**EditText  android:id="@+id/add\_kgEt"  android:layout\_width="80dp"  android:layout\_height="wrap\_content"  android:inputType="numberDecimal"  android:textAlignment="center"  android:textSize="20dp"** />  <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:textSize="25dp"  android:text="kg"**/>  </**LinearLayout**>   *<!--버튼-->* <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="1"  android:gravity="center"**>  <**Button  android:id="@+id/add\_kgBtn"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:text="SAVE"  android:textSize="25dp"  android:background="#fc9d9a"  android:textColor="@android:color/white"**/>  </**LinearLayout**> </**LinearLayout**> | |

* **기능 구현**

|  |
| --- |
| **AddKgActivity.java** |
| **package** hbapp.realfinalproject;  **import** android.content.Context; **import** android.content.Intent; **import** android.graphics.Bitmap; **import** android.os.Bundle; **import** android.provider.MediaStore; **import** android.support.annotation.Nullable; **import** android.support.v7.app.AppCompatActivity; **import** android.view.View; **import** android.view.inputmethod.InputMethodManager; **import** android.widget.Button; **import** android.widget.EditText; **import** android.widget.ImageView; **import** android.widget.LinearLayout; **import** android.widget.Toast;  **import** com.bumptech.glide.Glide; **import** com.google.android.gms.tasks.OnSuccessListener; **import** com.google.firebase.database.DatabaseReference; **import** com.google.firebase.database.FirebaseDatabase; **import** com.google.firebase.storage.FirebaseStorage; **import** com.google.firebase.storage.StorageReference; **import** com.google.firebase.storage.UploadTask; **import** com.tsengvn.typekit.TypekitContextWrapper;  **import** java.io.ByteArrayOutputStream; **import** java.text.DateFormat; **import** java.text.SimpleDateFormat; **import** java.util.Date;  **import** hbapp.realfinalproject.R;  **import static** com.localhost.jooyeon.final\_project.GoalActivity.sharedPreferences; **import static** com.localhost.jooyeon.final\_project.MainActivity.database; **import static** hbapp.realfinalproject.GoalActivity.*sharedPreferences*; **import static** hbapp.realfinalproject.MainActivity.*database*;  **public class** AddKgActivity **extends** AppCompatActivity {   *// 뷰 정의* ImageView **add\_kgIv**;  Button **add\_kgBtn**;  EditText **add\_kgEt**;  LinearLayout **add\_kgLL**;   *// 사용자가 입력하는 몸무게* String **addKg**;   *// format* String **todayDate**;  String **todayDate\_photo**;   DatabaseReference **myRef**;   *// 사진 저장할 storage* FirebaseStorage **firebaseStorage**;  StorageReference **storageReference**;   @Override  **protected void** onCreate(@Nullable Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_add\_kg***);   *// 사용자를 구별하는 user\_id 값 가져오기  sharedPreferences* = getSharedPreferences(**"USER\_INFO"**, ***MODE\_PRIVATE***);  String user\_id = String.*valueOf*(*sharedPreferences*.getLong(**"user\_id"**, 0));   *database* = FirebaseDatabase.getInstance();  **myRef** = *database*.getReference(user\_id);   *// 오늘 날짜를 todayDate와 todayDate\_photo에 String형으로 저장* Date date = **new** Date();  DateFormat df = **new** SimpleDateFormat(**"yyyy-MM-dd"**);  DateFormat df\_photo = **new** SimpleDateFormat(**"yyyyMMddhm"**);   **todayDate** = df.format(date);  **todayDate\_photo** = df\_photo.format(date);   *// 찍은 사진을 firebase storage에 저장* **firebaseStorage** = FirebaseStorage.getInstance();  **storageReference** = **firebaseStorage**.getReference(**"camera"**).child(**todayDate\_photo** +**".jpg"**);   *// id 값 부여* **add\_kgLL** = (LinearLayout)findViewById(R.id.***add\_kgLL***);  **add\_kgBtn** = (Button)findViewById(R.id.***add\_kgBtn***);  **add\_kgIv** = (ImageView)findViewById(R.id.***add\_kgIv***);  **add\_kgEt** = (EditText)findViewById(R.id.***add\_kgEt***);   *// 이미지 뷰를 클릭하였을 경우* **add\_kgIv**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  *// 카메라 어플이 켜지도록* Intent i = **new** Intent(MediaStore.***ACTION\_IMAGE\_CAPTURE***);  startActivityForResult(i, 1);  }  });   *// save 버튼을 눌렀을 경우* **add\_kgBtn**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  *// edittext 값을 firebase database 에 저장* **addKg** = **add\_kgEt**.getText().toString();  **myRef**.child(**"date"**).child(**todayDate**).child(**"kg"**).setValue(**addKg**);   finish();  }  });   *// 전체 레이아웃을 클릭할 때 키보드가 숨겨지도록* **add\_kgLL**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  hideKeyboard();  }  });  } *// onCreate() 끝* @Override  **protected void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  **super**.onActivityResult(requestCode, resultCode, data);  **if**(requestCode == 1 && resultCode == ***RESULT\_OK***) {  *// 찍은 사진의 uri 를 firebase database에 저장* **myRef**.child(**"date"**).child(**todayDate**).child(**"uri"**).setValue(String.*valueOf*(data.getData()));   *// Glide 오픈 API를 이용해서 imageview에 사진 띄우기* Glide.with(**this**)  .load(data.getData())  .centerCrop()  .override(600, 800)  .into(**add\_kgIv**);   **add\_kgIv**.setDrawingCacheEnabled(**true**);  **add\_kgIv**.buildDrawingCache();  Bitmap bitmap = **add\_kgIv**.getDrawingCache();  ByteArrayOutputStream baos = **new** ByteArrayOutputStream();  bitmap.compress(Bitmap.CompressFormat.***JPEG***, 100, baos);  **byte**[] datas = baos.toByteArray();  UploadTask uploadTask = (UploadTask) **storageReference**.putBytes(datas)  .addOnSuccessListener(**new** OnSuccessListener<UploadTask.TaskSnapshot>() {  @Override  **public void** onSuccess(UploadTask.TaskSnapshot taskSnapshot) {  Toast.*makeText*(AddKgActivity.**this**, **"사진 저장 완료"**, Toast.***LENGTH\_SHORT***).show();  }  });  }  } *// onActivityResult() 끝   // 키보드를 숨기는 메소드* **private void** hideKeyboard() {  InputMethodManager inputManager = (InputMethodManager) getSystemService(Context.***INPUT\_METHOD\_SERVICE***);  inputManager.hideSoftInputFromWindow(**this**.getCurrentFocus().getWindowToken(), InputMethodManager.***HIDE\_NOT\_ALWAYS***);  } *//hideKeyboard() 끝   // 글씨체* @Override  **protected void** attachBaseContext(Context newBase) {  **super**.attachBaseContext(TypekitContextWrapper.wrap(newBase));  } } |

1. **화면 6 (Show Seekbar 화면)**

* **레이아웃 구현**

|  |  |
| --- | --- |
| 화면(UI) | 설명 |
|  | - 날짜와 몸무게를 표시하는 textview 와 그 날의 사진을 띄우는 Imageview, 그리고 하단의 seekbar로 이루어져 있다.  - seekbar를 움직이면 firebase에 저장된 값들을 불러와서 날짜별로 체형 사진과 몸무게를 볼 수 있고, 이를 이용해서 체형의 변화를 보다 쉽게 파악할 수 있다. |
| activity\_seekbar.xml | |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout  xmlns:android="http://schemas.android.com/apk/res/android"  android:orientation="vertical"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"**>  *<!-- 메인 화면 -->* <**LinearLayout  android:padding="16dp"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"**>   *<!-- 날짜 -->* <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="1"  android:gravity="center"**>  <**TextView  android:id="@+id/seekbar\_date"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:textSize="25dp"  android:textStyle="bold"  android:textAlignment="center"  android:textColor="@android:color/holo\_blue\_dark"  android:text="오늘 날짜"**/>  </**LinearLayout**>  *<!--날짜 끝-->   <!-- 이미지와 몸무게 -->* <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="8.5"  android:orientation="vertical"**>  *<!-- 이미지 -->* <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="9"  android:gravity="center"**>  <**ImageView  android:id="@+id/seekbar\_iv"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"**/>  </**LinearLayout**>  *<!-- 이미지 끝 -->   <!-- 몸무게 -->* <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="1"  android:gravity="center"**>  <**TextView  android:id="@+id/seekbar\_kg"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:textSize="25dp"  android:textStyle="bold"  android:textAlignment="center"  android:text="48"  android:textColor="@android:color/holo\_red\_light"**/>  <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:textSize="25dp"  android:textStyle="bold"  android:text="kg"**/>  </**LinearLayout**>  *<!--몸무게 끝-->* </**LinearLayout**>  *<!-- 이미지와 몸무게 끝-->   <!--seek 바-->* <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="0.5"**>  <**android.support.v7.widget.AppCompatSeekBar  android:id="@+id/seekbar"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:max="0"**/>  </**LinearLayout**>  *<!--seek 바 끝-->* </**LinearLayout**>  *<!-- 메인화면 끝-->* </**LinearLayout**> | |

* **기능 구현**

|  |
| --- |
| **SeekbarActivity.java** |
| **package** hbapp.realfinalproject;  **import** android.content.Context; **import** android.net.Uri; **import** android.os.Bundle; **import** android.support.annotation.Nullable; **import** android.support.v7.app.AppCompatActivity; **import** android.widget.ImageView; **import** android.widget.SeekBar; **import** android.widget.TextView;  **import** com.bumptech.glide.Glide; **import** com.google.firebase.database.ChildEventListener; **import** com.google.firebase.database.DataSnapshot; **import** com.google.firebase.database.DatabaseError; **import** com.google.firebase.database.DatabaseReference; **import** com.google.firebase.database.FirebaseDatabase; **import** com.tsengvn.typekit.TypekitContextWrapper;  **import** java.text.DateFormat; **import** java.text.SimpleDateFormat; **import** java.util.ArrayList; **import** java.util.Date;  **import static** com.localhost.jooyeon.final\_project.GoalActivity.sharedPreferences; **import static** com.localhost.jooyeon.final\_project.MainActivity.database; **import static** hbapp.realfinalproject.GoalActivity.*sharedPreferences*; **import static** hbapp.realfinalproject.MainActivity.*database*;  */\*\*  \* Created by asmwj on 2017-06-17.  \*/* **public class** SeekbarActivity **extends** AppCompatActivity {   *// 뷰 정의* ImageView **seekbar\_iv**;  SeekBar **seekBar**;  TextView **seekbar\_tv**;  TextView **seekbar\_date**;   *// seekbar에 필요한 정보를 저장시킬 ArrayList* ArrayList<SeekbarInfo> **seekbarInfos**;   *// seekbar에 필요한 정보를 저장시킬 변수* String **today\_kg**;  String **today\_img**;  String **today\_date**;   DatabaseReference **myRef**;   *// 카운트 변수* **int i** = 0;   *// 오늘 날짜* String **todayDate**;   @Override  **protected void** onCreate(@Nullable Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_seekbar***);   *// 사용자를 구별하는 user\_id 값 가져오기  sharedPreferences* = getSharedPreferences(**"USER\_INFO"**, ***MODE\_PRIVATE***);  String user\_id = String.*valueOf*(*sharedPreferences*.getLong(**"user\_id"**, 0));   *database* = FirebaseDatabase.getInstance();  **myRef** = *database*.getReference(user\_id);   *// 인스턴스 생성* **seekbarInfos** = **new** ArrayList<SeekbarInfo>();   *// 오늘 날짜를 가져오는 date 변수* Date date = **new** Date();  DateFormat df = **new** SimpleDateFormat(**"yyyy-MM-dd"**);  **todayDate** = df.format(date);   *// id값 부여* **seekBar** = (SeekBar)findViewById(R.id.***seekbar***);  **seekbar\_iv** = (ImageView)findViewById(R.id.***seekbar\_iv***);  **seekbar\_tv** = (TextView)findViewById(R.id.***seekbar\_kg***);  **seekbar\_date** = (TextView)findViewById(R.id.***seekbar\_date***);   *// firebase "date"키 아래의 값들을 가져오기 위한 Listener* **myRef**.child(**"date"**).addChildEventListener(**new** ChildEventListener() {  *// datasnapshot = 2017xxxx, 2017xxxx, ...* @Override  **public void** onChildAdded(DataSnapshot dataSnapshot, String s) {  *// "date" 키값을 today\_date 변수에 저장* **today\_date** = dataSnapshot.getKey().toString();  *// date 아래에 있는 kg, uri를 childDataSnapshot으로* **for**(DataSnapshot childDataSnapshot:dataSnapshot.getChildren()) {  *// 만약 키 값이 "kg"일 경우* **if** (childDataSnapshot.getKey().equals(**"kg"**)) {  *// "kg"의 값을 today\_kg 변수에 저장* **today\_kg** = childDataSnapshot.getValue().toString();  }  *// 만약 키 값이 "img"일 경우* **if**(childDataSnapshot.getKey().equals(**"uri"**)){  *// "uri"의 값을 today\_img 변수에 저장* **today\_img** = childDataSnapshot.getValue().toString();  }  } *// for문 끝   // seekbarInfo 인스턴스에 데이터 추가* **seekbarInfos**.add(**new** SeekbarInfo(**today\_date**, **today\_img**, **today\_kg**));   *// 카운트 값을 seekbar의 max값으로* **seekBar**.setMax(**i**);   **i**++; *// i값 증가   // seekbar listener* **seekBar**.setOnSeekBarChangeListener(**new** SeekBar.OnSeekBarChangeListener() {  *// progress가 바뀔때마다* @Override  **public void** onProgressChanged(SeekBar seekBar, **int** progress, **boolean** fromUser) {  *// progress의 시작 = 0  // index 의 시작 = 0 으로 같으므로 seekbarInfos.get(progress)를 해주었다.   // seekbarInfos의 kg값을 get해서 seekbar\_tv에 text를 set해주었다.* **seekbar\_tv**.setText(**seekbarInfos**.get(progress).getKg());   *// seekbarInfos의 date값을 get해서 seekbar\_date에 text를 set해주었다.* **seekbar\_date**.setText(**seekbarInfos**.get(progress).getDate());   *//get 한 uri가 null이 아닐 경우에만* **if**(**seekbarInfos**.get(progress).getUri() != **null**){  *// Glide 오픈 API를 이용하여 seekbar\_iv에 이미지 띄운다.* Glide.with(SeekbarActivity.**this**)  .load(Uri.*parse*(**seekbarInfos**.get(progress).getUri()))  .centerCrop()  .override(600, 800)  .into(**seekbar\_iv**);  }  }  *// Seekbar가 처음 시작할 경우* @Override  **public void** onStartTrackingTouch(SeekBar seekBar) {  *// 인덱스가 0인 값을 가져와서 set해준다.* **seekbar\_tv**.setText(**seekbarInfos**.get(0).getKg());  **seekbar\_date**.setText(**seekbarInfos**.get(0).getDate());  **if**(**seekbarInfos**.get(0).getUri() != **null**){  Glide.with(SeekbarActivity.**this**)  .load(Uri.*parse*(**seekbarInfos**.get(0).getUri()))  .centerCrop()  .override(600, 800)  .into(**seekbar\_iv**);  }  }   @Override  **public void** onStopTrackingTouch(SeekBar seekBar) {}  });  }    @Override  **public void** onChildChanged(DataSnapshot dataSnapshot, String s) {}   @Override  **public void** onChildRemoved(DataSnapshot dataSnapshot) {}   @Override  **public void** onChildMoved(DataSnapshot dataSnapshot, String s) {}   @Override  **public void** onCancelled(DatabaseError databaseError) {}  });  }   *// 글씨체* @Override  **protected void** attachBaseContext(Context newBase) {  **super**.attachBaseContext(TypekitContextWrapper.wrap(newBase));  } } |
| **SeekbarInfo.java** |
| **package** hbapp.realfinalproject;  */\*\*  \* Created by asmwj on 2017-06-17.  \*/* **public class** SeekbarInfo {  String **date**;  String **uri**;  String **kg**;   SeekbarInfo(String date, String uri, String kg){  **this**.**date** = date;  **this**.**uri** = uri;  **this**.**kg** = kg;  }   **public** String getDate() {  **return date**;  }   **public void** setDate(String date) {  **this**.**date** = date;  }   **public** String getUri() {  **return uri**;  }   **public void** setUri(String uri) {  **this**.**uri** = uri;  }   **public** String getKg() {  **return kg**;  }   **public void** setKg(String kg) {  **this**.**kg** = kg;  } } |

1. **화면 7 (Show Change 화면)**

* **레이아웃 구현**

|  |  |
| --- | --- |
| 화면(UI) | 설명 |
|  | - 표처럼 보이기 위해 상단에 TextView를 배치하고, 목표 몸무게까지 얼마나 남았는지 보이기 위해 하단에도 TextView를 두었다.  - Recycler View 와 Adapter를 이용해서 미리 만들어 놓은 item 레이아웃을 데이터의 양만큼 RecyclerView에 배치할 수 있게 하였다.  - Item 레이아웃에는 오늘의 날짜, 오늘의 몸무게, 오늘의 만보기 횟수가 나타나도록 하였다.  - Firebase에서 실시간으로 데이터를 불러오기 때문에 데이터의 수정이 일어날 경우 즉시 바뀐다. |
| activity\_showchange.xml | |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:orientation="vertical" android:layout\_width="match\_parent"  android:layout\_height="match\_parent"**>  *<!--표 속성-->* <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="0.5"  android:orientation="horizontal"**>  *<!--날짜-->* <**TextView  android:layout\_width="0dp"  android:layout\_height="match\_parent"  android:layout\_weight="4"  android:text="날짜"  android:textStyle="bold"  android:textColor="@android:color/white"  android:gravity="center"  android:background="@android:color/holo\_red\_light"**/>  *<!--몸무게-->* <**TextView  android:layout\_width="0dp"  android:layout\_height="match\_parent"  android:layout\_weight="2"  android:text="몸무게"  android:textStyle="bold"  android:textColor="@android:color/white"  android:gravity="center"  android:background="@android:color/holo\_orange\_light"**/>  *<!--만보기 횟수-->* <**TextView  android:layout\_width="0dp"  android:layout\_height="match\_parent"  android:layout\_weight="4"  android:text="만보기 횟수"  android:textStyle="bold"  android:textColor="@android:color/white"  android:gravity="center"  android:background="@android:color/holo\_green\_light"**/>  </**LinearLayout**>   <**android.support.v7.widget.RecyclerView  android:id="@+id/showChange\_RV"  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="9"  android:divider="@android:color/darker\_gray"  android:dividerHeight="1dp"**/>  <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:background="@android:color/darker\_gray"**/>  <**TextView  android:id="@+id/showChange\_tv"  android:padding="3dp"  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="0.5"  android:textAlignment="center"  android:gravity="center"  android:textSize="16dp"  android:textStyle="bold"  android:textColor="@android:color/holo\_blue\_light"  android:text="목표 몸무게까지 -0.0kg!!"** /> </**LinearLayout**> | |
| activity\_item\_layout.xml | |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal"**>  <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="60dp"  android:orientation="horizontal"  android:gravity="center"**>   *<!--날짜-->* <**TextView  android:id="@+id/showChange\_date"  android:gravity="center\_vertical"  android:layout\_width="0dp"  android:layout\_height="match\_parent"  android:layout\_weight="4"  android:textAlignment="center"  android:textSize="20dp"  android:text="20170614"  android:textColor="@android:color/black"**/>   *<!--몸무게 + kg-->* <**LinearLayout  android:layout\_width="0dp"  android:layout\_weight="2"  android:layout\_height="match\_parent"  android:orientation="horizontal"  android:gravity="center"**>  <**TextView  android:id="@+id/showChange\_kg"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:textAlignment="center"  android:textStyle="bold"  android:textSize="20dp"  android:text="48"  android:textColor="@android:color/holo\_red\_light"**/>  <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:textSize="20dp"  android:textColor="@android:color/black"  android:text="kg"**/>  </**LinearLayout**>  *<!--몸무게 + kg 끝-->   <!--만보기-->* <**LinearLayout  android:layout\_width="0dp"  android:layout\_weight="4"  android:layout\_height="match\_parent"  android:orientation="horizontal"  android:gravity="center"**>  <**TextView  android:id="@+id/showChange\_walkCnt"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:textAlignment="center"  android:textSize="20dp"  android:text="10000"  android:textColor="@android:color/holo\_blue\_dark"**/>  <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:textSize="20dp"  android:textColor="@android:color/black"  android:text="회"**/>  </**LinearLayout**>  *<!--만보기 횟수 끝-->* </**LinearLayout**> </**LinearLayout**> | |

* **기능 구현**

|  |
| --- |
| **ShowChangeActivity.java** |
| **package** hbapp.realfinalproject;  **import** android.content.Context; **import** android.os.Bundle; **import** android.support.annotation.Nullable; **import** android.support.v7.app.AppCompatActivity; **import** android.support.v7.widget.LinearLayoutManager; **import** android.support.v7.widget.RecyclerView; **import** android.util.Log; **import** android.widget.TextView;  **import** com.google.firebase.database.ChildEventListener; **import** com.google.firebase.database.DataSnapshot; **import** com.google.firebase.database.DatabaseError; **import** com.google.firebase.database.DatabaseReference; **import** com.google.firebase.database.FirebaseDatabase; **import** com.tsengvn.typekit.TypekitContextWrapper;  **import** java.text.DateFormat; **import** java.text.SimpleDateFormat; **import** java.util.ArrayList; **import** java.util.Date;  **import static** hbapp.realfinalproject.GoalActivity.*sharedPreferences*; **import static** hbapp.realfinalproject.MainActivity.*database*;  */\*\*  \* Created by asmwj on 2017-06-16.  \*/* **public class** ShowChangeActivity **extends** AppCompatActivity {   *// recyclerview 와 하단의 textview 변수* RecyclerView **recyclerView**;  TextView **restKgTv**;   *// firebase 에서 가져 올 데이터 값 변수* String **get\_kg**;  String **get\_date**;  Long **get\_walkingNum**=0L;   *// 목표 kg까지 남은 kg 계산하기 위한 변수* Float **goal\_kg**, **current\_kg**, **result\_kg**;  *// 오늘 날짜를 저장하기 위한 변수* String **todayDate**;   LinearLayoutManager **mLayoutManager**;  DatabaseReference **myRef**;   *// adapter 변수* ShowChangeAdapter **adapter**;   *// 현재 사용자의 정보를 저장하기 위한 ArrayList* ArrayList<CurrentUser> **currentUserArrayList**;   @Override  **protected void** onCreate(@Nullable Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.activity\_showchange);   *// 사용자를 구별하는 user\_id  sharedPreferences* = getSharedPreferences(**"USER\_INFO"**, ***MODE\_PRIVATE***);  String user\_id = String.*valueOf*(*sharedPreferences*.getLong(**"user\_id"**, 0));   *// sharedPreferences 에 저장된 목표 몸무게 가져오기* **goal\_kg** = *sharedPreferences*.getFloat(**"kg"**, 0);   *// 인스턴스 생성* **currentUserArrayList** = **new** ArrayList<CurrentUser>();  **currentUserArrayList**.clear();   *// 뷰 정의* **restKgTv** = (TextView)findViewById(R.id.showChange\_tv);  **recyclerView** = (RecyclerView)findViewById(R.id.showChange\_RV);   *// LinearLayoutManager을 recyclerview에 set* **mLayoutManager** = **new** LinearLayoutManager(**this**);  **recyclerView**.setLayoutManager(**mLayoutManager**);   *// adpter 생성* **adapter** = **new** ShowChangeAdapter(**this**, R.layout.activity\_item\_layout, **currentUserArrayList**);  *// adapter 갱신을 위한 메소드* **adapter**.notifyDataSetChanged();   *// recyclerview에 adapter set* **recyclerView**.setAdapter(**adapter**);   *database* = FirebaseDatabase.getInstance();  **myRef** = *database*.getReference(user\_id);   *// 오늘 날짜를 가져오는 date 변수* Date date = **new** Date();  DateFormat df = **new** SimpleDateFormat(**"yyyy-MM-dd"**);  **todayDate** = df.format(date);    *// 오늘 날짜의 몸무게를 가져오기 위한 listener* **myRef**.child(**"date"**).child(**todayDate**).addChildEventListener(**new** ChildEventListener() {  @Override  **public void** onChildAdded(DataSnapshot dataSnapshot, String s) {  *// dataSnapashow의 키가 "kg"일 경우* **if**(dataSnapshot.getKey().equals(**"kg"**)) {  **current\_kg** = Float.valueOf(dataSnapshot.getValue().toString());   *// 오늘 몸무게 - 목표 몸무게* **result\_kg** = **current\_kg**-**goal\_kg**;   *// 결과 몸무게가 0이상일 경우* **if**(**result\_kg** > 0){  String result = String.*format*(**"목표 몸무게 까지 - %3.1fkg !!"**, **result\_kg**);  **restKgTv**.setText(result);  }  *// 목표 몸무게를 달성했을 경우* **else if**(**result\_kg** == 0){  String result = String.*format*(**"축하드립니다!! 목표 달성!!"**);  **restKgTv**.setText(result);  }   *//살이 쪘을 경우* **else if** (**result\_kg** < 0){  **float** absR=Math.*abs*(**result\_kg**);  String result = String.*format*(**"목표 몸무게까지 +%3.1fkg!!"**, absR);  **restKgTv**.setText(result);  }  }  }   @Override  **public void** onChildChanged(DataSnapshot dataSnapshot, String s) {}   @Override  **public void** onChildRemoved(DataSnapshot dataSnapshot) {}   @Override  **public void** onChildMoved(DataSnapshot dataSnapshot, String s) {}   @Override  **public void** onCancelled(DatabaseError databaseError) {}  });   *// 오늘 몸무게를 추가시킬 때 리스트도 추가되도록 하는 listener* **myRef**.child(**"date"**).addChildEventListener(**new** ChildEventListener() {  *// 여기서 datasnapshot은 2017-xx-xx 를 가리킴* @Override  **public void** onChildAdded(DataSnapshot dataSnapshot, String s) {  Log.*d*(**"ㅁㄴㅇㄹㅇㄴㅁㄹ"**,**"ㅁㄴㅇㄻㄴㄹ"**);  **get\_date** = dataSnapshot.getKey().toString();  *// kg, walkcnt 각각을 childDataSnapshot으로* **for**(DataSnapshot childDataSnapshot:dataSnapshot.getChildren()) {  **if** (childDataSnapshot.getKey().equals(**"kg"**)) {  **get\_kg** = childDataSnapshot.getValue().toString();  }  **if** (childDataSnapshot.getKey().equals(**"walkingNum"**)) {  **get\_walkingNum** = (Long)childDataSnapshot.getValue();   }  }  **currentUserArrayList**.add(**new** com.localhost.jooyeon.final\_project.CurrentUser(**get\_date**, **get\_kg**, **get\_walkingNum**.toString()));  **adapter**.notifyDataSetChanged();   Log.*d*(**"추가"**,**"추가"**);  }   @Override  **public void** onChildChanged(DataSnapshot dataSnapshot, String s) {}   @Override  **public void** onChildRemoved(DataSnapshot dataSnapshot) {}   @Override  **public void** onChildMoved(DataSnapshot dataSnapshot, String s) {}   @Override  **public void** onCancelled(DatabaseError databaseError) {}  });    }  @Override  **protected void** attachBaseContext(Context newBase) {  **super**.attachBaseContext(TypekitContextWrapper.wrap(newBase));  } } |
| **ShowChangeAdapter.java** |
| **package** hbapp.realfinalproject;  **import** android.content.Context; **import** android.support.v7.widget.RecyclerView; **import** android.view.LayoutInflater; **import** android.view.View; **import** android.view.ViewGroup; **import** android.widget.TextView;  **import** java.util.ArrayList;  */\*\*  \* Created by asmwj on 2017-06-16.  \*/* **public class** ShowChangeAdapter **extends** RecyclerView.Adapter<ShowChangeAdapter.ViewHolder> {   Context context;  **int layout**;  ArrayList<CurrentUser> **currentUsers**;  LayoutInflater **inflater**;  View **view**;   *//ShowChangeAdapter 생성자 정의* **public** ShowChangeAdapter(Context c, **int** layout, ArrayList<CurrentUser> data){  **this**.context = c;  **this**.**layout** = layout;  **this**.**currentUsers** = data;  **inflater** = (LayoutInflater)context.getSystemService(Context.***LAYOUT\_INFLATER\_SERVICE***);  }   *// RecyclerView에 넣을 item의 뷰들 정의* **public class** ViewHolder **extends** RecyclerView.ViewHolder {  TextView **dateTv**;  TextView **kgTv**;  TextView **cntTv**;   **public** ViewHolder(View itemView) {  **super**(itemView);  **dateTv** = (TextView)**view**.findViewById(R.id.***showChange\_date***);  **kgTv** = (TextView)**view**.findViewById(R.id.***showChange\_kg***);  **cntTv** = (TextView)**view**.findViewById(R.id.***showChange\_walkCnt***);  }  }   *// viewHolder 객체를 받기 위해* @Override  **public** ShowChangeAdapter.ViewHolder onCreateViewHolder(ViewGroup parent, **int** viewType) {  **view** = LayoutInflater.*from*(parent.getContext()).inflate(**this**.**layout**,parent,**false**);  ShowChangeAdapter.ViewHolder viewHolder = **new** ShowChangeAdapter.ViewHolder(**view**);  **return** viewHolder;  }   *// position 마다의 currentUsers의 데이터를 각각 textview에 set* @Override  **public void** onBindViewHolder(ViewHolder holder, **int** position) {  CurrentUser c = **currentUsers**.get(position);  holder.**dateTv**.setText(c.**date**);  holder.**kgTv**.setText(c.**current\_kg**);  holder.**cntTv**.setText(c.**walk\_cnt**);  }   @Override  **public int** getItemCount() {  **return currentUsers**.size();  } } |
| **CurrentUser.java** |
| **package** hbapp.finalfirebase;  */\*\*  \* Created by asmwj on 2017-06-16.  \*/* **public class** CurrentUser {  String **current\_kg**;  String **date**;  String **walk\_cnt**;   **public** CurrentUser(String date, String kg, String walk\_cnt){  **this**.**current\_kg** = kg;  **this**.**date** = date;  **this**.**walk\_cnt** = walk\_cnt;  }   **public** String getCurrent\_kg() {  **return current\_kg**;  }   **public void** setCurrent\_kg(String current\_kg) {  **this**.**current\_kg** = current\_kg;  }   **public** String getDate() {  **return date**;  }   **public void** setDate(String date) {  **this**.**date** = date;  }   **public** String getWalk\_cnt() {  **return walk\_cnt**;  }   **public void** setWalk\_cnt(String walk\_cnt) {  **this**.**walk\_cnt** = walk\_cnt;  } } |

1. **화면8, 화면9**

* **레이아웃 구현**

|  |  |
| --- | --- |
| 화면(UI) | 설명 |
|  | - 퍼즐게임 초기 화면은 갤러리와 이미지뷰, 그리고 버튼으로 구성되어 있습니다.  -상단에 위치한 갤러리는 퍼즐게임을 위해 제공되는 도안들의 목록을 나타냅니다.  -중앙에 위치한 이미지뷰는 사용자가 갤러리에서 선택한 퍼즐 도안의 완성본을 보여줍니다.  -하단에 위치한 퍼즐 뽑기 버튼은 사용자가 만보기 기능으로 측정된 걸음수로 누적되는 포인트를 이용하여 퍼즐을 뽑는 버튼입니다. |
| coloring\_list.xml | |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:orientation="vertical"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="#ffffff"**>   <**Gallery  android:layout\_gravity="center\_vertical"  android:id="@+id/gallery"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="6.4"**/>   <**ImageView  android:layout\_margin="10dp"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="2"  android:scaleType="fitXY"  android:id="@+id/galleryView"**/>  <**Button  android:id="@+id/pickPuzzle"  android:background="#fcc1bf"  android:textStyle="bold"  android:textColor="#FFFFFF"  android:textSize="20dp"  android:text="퍼즐 뽑기"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="6.9"**/>  </**LinearLayout**> | |

|  |  |
| --- | --- |
| 화면(UI) | 설명 |
|  | - 퍼즐 뽑기의 화면은 대화상자로 구성되었습니다.  -사용자의 퍼즐 뽑기 포인트가 100 이하일 경우 현재 퍼즐 뽑기 기능을 사용할 수 없음을 의미하는 문구와 함께 현재 누적된 포인트가 토스트 메세지를 통해 나타납니다.  -사용자의 퍼즐 뽑기 포인트가 100 이상일 경우 사용자가 획득하지 못한 퍼즐들 중 랜덤으로 하나의 퍼즐을 제공하는데 이때 제공된 퍼즐 조각을 대화상자를 통해 보여집니다. |
| pick\_puzzle\_dialog.xml | |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:orientation="vertical" android:layout\_width="match\_parent"  android:layout\_height="match\_parent"**>   <**ImageView  android:layout\_marginTop="20dp"  android:layout\_marginBottom="20dp"  android:layout\_gravity="center"  android:layout\_width="150dp"  android:layout\_height="150dp"  android:id="@+id/pickPuzzleImageView"**/>  </**LinearLayout**> | |

* **기능 구현**

|  |
| --- |
| **ColoringListActivity.java** |
| **package** com.localhost.jooyeon.final\_project;  */\*\*  \* Created by yeona on 2017. 5. 27..  \*/* **import** android.app.Activity; **import** android.content.Context; **import** android.content.Intent; **import** android.os.Bundle; **import** android.support.v7.app.AlertDialog; **import** android.view.View; **import** android.widget.AdapterView; **import** android.widget.Button; **import** android.widget.Gallery; **import** android.widget.ImageView; **import** android.widget.Toast;  **import** com.google.firebase.database.DataSnapshot; **import** com.google.firebase.database.DatabaseError; **import** com.google.firebase.database.DatabaseReference; **import** com.google.firebase.database.FirebaseDatabase; **import** com.google.firebase.database.ValueEventListener; **import** com.tsengvn.typekit.TypekitContextWrapper;  **import** java.util.HashMap; **import** java.util.Random;  **import static** com.localhost.jooyeon.final\_project.GoalActivity.*sharedPreferences*;  **public class** ColoringListActivity **extends** Activity {  */\*\* Called when the activity is first created. \*/* String **user\_id**;  ImageView [] **imageView**;  ImageView **galleryView**;  **int position**=0;  Button **pickPuzzle**;  Long **puzzleCount**;  **boolean isStart** = **false**;   **int**[][] **imageResId** = {  {R.drawable.***piece1\_1***,R.drawable.***piece1\_2***,R.drawable.***piece1\_3***,R.drawable.***piece1\_4***,R.drawable.***piece1\_5***,R.drawable.***piece1\_6***,  R.drawable.***piece1\_7***,R.drawable.***piece1\_8***,R.drawable.***piece1\_9***,R.drawable.***piece1\_10***,R.drawable.***piece1\_11***,R.drawable.***piece1\_12***},   {R.drawable.***piece2\_1***,R.drawable.***piece2\_2***,R.drawable.***piece2\_3***,R.drawable.***piece2\_4***,R.drawable.***piece2\_5***,R.drawable.***piece2\_6***,  R.drawable.***piece2\_7***,R.drawable.***piece2\_8***,R.drawable.***piece2\_9***,R.drawable.***piece2\_10***,R.drawable.***piece2\_11***,R.drawable.***piece2\_12***},   {R.drawable.***piece3\_1***,R.drawable.***piece3\_2***,R.drawable.***piece3\_3***,R.drawable.***piece3\_4***,R.drawable.***piece3\_5***,R.drawable.***piece3\_6***,  R.drawable.***piece3\_7***,R.drawable.***piece3\_8***,R.drawable.***piece3\_9***,R.drawable.***piece3\_10***,R.drawable.***piece3\_11***,R.drawable.***piece3\_12***},   {R.drawable.***piece4\_1***,R.drawable.***piece4\_2***,R.drawable.***piece4\_3***,R.drawable.***piece4\_4***,R.drawable.***piece4\_5***,R.drawable.***piece4\_6***,  R.drawable.***piece4\_7***,R.drawable.***piece4\_8***,R.drawable.***piece4\_9***,R.drawable.***piece4\_10***,R.drawable.***piece4\_11***,R.drawable.***piece4\_12***}  };   **int puzzleArray**[][]=**new int**[4][12];   *//파이어베이스 연동 및 레퍼런스* DatabaseReference **mRootRef** = FirebaseDatabase.*getInstance*().getReference();  DatabaseReference **mpuzzleCountRef**;  DatabaseReference **mpuzzleGameRef**;   @Override  **protected void** onStart() {  **super**.onStart();  **mpuzzleCountRef**.addValueEventListener(**new** ValueEventListener() {  @Override  *//퍼즐 뽑기 여부 확인을 위한 값 초기화* **public void** onDataChange(DataSnapshot dataSnapshot) {  **puzzleCount** = (Long)dataSnapshot.getValue();  }   @Override  **public void** onCancelled(DatabaseError databaseError) {   }  });  **mpuzzleGameRef**.addValueEventListener(**new** ValueEventListener() {  @Override  *//파이어베이스에 저장된 퍼즐별 사용자가 갖고 있는 퍼즐 종류를 가져옴* **public void** onDataChange(DataSnapshot dataSnapshot) {  HashMap puzzleGameMap = (HashMap)dataSnapshot.getValue();   HashMap picture1Map = (HashMap)puzzleGameMap.get(**"picture1"**);  HashMap picture2Map = (HashMap)puzzleGameMap.get(**"picture2"**);  HashMap picture3Map = (HashMap)puzzleGameMap.get(**"picture3"**);  HashMap picture4Map = (HashMap)puzzleGameMap.get(**"picture4"**);   HashMap userPuzzle1Map = (HashMap)picture1Map.get(**"userPuzzle"**);  HashMap userPuzzle2Map = (HashMap)picture2Map.get(**"userPuzzle"**);  HashMap userPuzzle3Map = (HashMap)picture3Map.get(**"userPuzzle"**);  HashMap userPuzzle4Map = (HashMap)picture4Map.get(**"userPuzzle"**);    **for**(**int** i=0;i<12;i++){  Long puzzle = ((Long)userPuzzle1Map.get(**"puzzle"**+Integer.*toString*(i+1)));  **if**(puzzle==**null**)  **return**;  **puzzleArray**[0][i] = puzzle.intValue();  }  **for**(**int** i=0;i<12;i++){  Long puzzle = ((Long)userPuzzle2Map.get(**"puzzle"**+Integer.*toString*(i+1)));  **if**(puzzle==**null**)  **return**;  **puzzleArray**[1][i] = ((Long)userPuzzle2Map.get(**"puzzle"**+Integer.*toString*(i+1))).intValue();  }  **for**(**int** i=0;i<12;i++){  Long puzzle = ((Long)userPuzzle3Map.get(**"puzzle"**+Integer.*toString*(i+1)));  **if**(puzzle==**null**)  **return**;  **puzzleArray**[2][i] = ((Long)userPuzzle3Map.get(**"puzzle"**+Integer.*toString*(i+1))).intValue();  }  **for**(**int** i=0;i<12;i++){  Long puzzle = ((Long)userPuzzle4Map.get(**"puzzle"**+Integer.*toString*(i+1)));  **if**(puzzle==**null**)  **return**;  **puzzleArray**[3][i] = ((Long)userPuzzle4Map.get(**"puzzle"**+Integer.*toString*(i+1))).intValue();  }  }   @Override  **public void** onCancelled(DatabaseError databaseError) {   }  });   }   @Override  **public void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***coloring\_list***);   *sharedPreferences* = getSharedPreferences(**"USER\_INFO"**, ***MODE\_PRIVATE***);  **user\_id** = String.*valueOf*(*sharedPreferences*.getLong(**"user\_id"**, 0));    **mpuzzleCountRef** = **mRootRef**.child(**user\_id**+**"/puzzleGame/puzzleCount"**);  **mpuzzleGameRef** = **mRootRef**.child(**user\_id**+**"/puzzleGame"**);   **pickPuzzle** = (Button)findViewById(R.id.***pickPuzzle***);  **galleryView** = (ImageView)findViewById(R.id.***galleryView***);  Gallery g = (Gallery) findViewById(R.id.***gallery***);  **final** ImageAdapter imageAdapter=**new** ImageAdapter(**this**);  g.setAdapter(imageAdapter);  **galleryView**.setImageResource(imageAdapter.getImageId(0));   *//갤러리에서 선택한 퍼즐에 대한 이미지 출력* g.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  **public void** onItemClick(AdapterView parent, View v, **int** position, **long** id) {  **galleryView**.setImageResource(imageAdapter.getImageId(position));  ColoringListActivity.**this**.**position**=position;  }  });   *//선택한 퍼즐로 시작* **galleryView**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  Intent intent = **new** Intent(ColoringListActivity.**this**, ColoringActivity.**class**);  intent.putExtra(**"position"**, **position**);   startActivity(intent);  }  });   *//퍼즐 뽑기 버튼 클릭 시 일정 횟수 이상 시 퍼즐을 뽑고, 안되면 못뽑음* **pickPuzzle**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **if**(**puzzleCount**>=100){  Toast.*makeText*(getApplicationContext(), **puzzleCount**.toString(), Toast.***LENGTH\_SHORT***).show();  **mpuzzleCountRef**.setValue(**puzzleCount**-100);   Random random = **new** Random(System.*currentTimeMillis*());   AlertDialog.Builder pickDialog = **new** AlertDialog.Builder(ColoringListActivity.**this**);  pickDialog.setTitle(**"축하합니다!!"**);   *//사용자가 갖고 있지 않은 퍼즐을 알아내여 그 중 랜덤으로 퍼즐을 제공* **while**(**true**){  **int** pictureNum = Math.*abs*(random.nextInt())%4;  **int** puzzleNum = Math.*abs*(random.nextInt())%12;   **if**(**puzzleArray**[pictureNum][puzzleNum]==0){  **mpuzzleGameRef**.child(**"picture"**+Integer.*toString*(pictureNum+1)).child(**"userPuzzle"**).child(**"puzzle"**+Integer.*toString*(puzzleNum+1)).setValue(1);  View view = View.*inflate*(ColoringListActivity.**this**,R.layout.***pick\_puzzle\_dialog***,**null**);  ImageView pickPuzzleImageView = (ImageView)view.findViewById(R.id.***pickPuzzleImageView***);  pickPuzzleImageView.setImageResource(**imageResId**[pictureNum][puzzleNum]);  pickPuzzleImageView.setScaleType(ImageView.ScaleType.***FIT\_XY***);  pickDialog.setView(view);  pickDialog.show();  **break**;  }  }  }  **else**{  Toast.*makeText*(getApplicationContext(), **"100번 이상 걸어야 뽑을 수 있습니다! (현재 누적 횟수: "**+Long.*toString*(**puzzleCount**)+**")"**, Toast.***LENGTH\_SHORT***).show();  }   }  });  }  @Override  **protected void** attachBaseContext(Context newBase) {  **super**.attachBaseContext(TypekitContextWrapper.*wrap*(newBase));  }  } |

|  |
| --- |
| **ImageAdapter.java** |
| **package** com.localhost.jooyeon.final\_project;  **import** android.content.Context; **import** android.content.res.TypedArray; **import** android.view.View; **import** android.view.ViewGroup; **import** android.widget.BaseAdapter; **import** android.widget.Gallery; **import** android.widget.ImageView;  */\*\*  \* Created by yeona on 2017. 5. 27..  \*/* **public class** ImageAdapter **extends** BaseAdapter {  **int mGalleryItemBackground**;  **private** Context **mContext**;  **private** Integer[] **mImageIds** = {  R.drawable.***puzzle1***,  R.drawable.***puzzle2***,  R.drawable.***puzzle3***,  R.drawable.***puzzle4*** };   *//퍼즐 리스트에 띄울 갤러리에 대한 설정 등록* **public** ImageAdapter(Context c) {  **mContext** = c;  TypedArray a = c.obtainStyledAttributes(R.styleable.***GalleryTheme***);  **mGalleryItemBackground** = a.getResourceId(R.styleable.***GalleryTheme\_android\_galleryItemBackground***, 0);  a.recycle();  }   *//갤러리에 포함시킬 이미지에 대한 정보들* **public int** getImageId(**int** position){  **return mImageIds**[position];  }  **public int** getCount() {  **return mImageIds**.**length**;  }  **public** Object getItem(**int** position) {  **return** position;  }  **public long** getItemId(**int** position) {  **return** position;  }   *//갤러리에 이미지 등록* **public** View getView(**int** position, View convertView, ViewGroup parent) {  ImageView i = **new** ImageView(**mContext**);  i.setImageResource(**mImageIds**[position]);  i.setLayoutParams(**new** Gallery.LayoutParams(250, 200));  i.setScaleType(ImageView.ScaleType.***FIT\_XY***);  i.setBackgroundResource(**mGalleryItemBackground** );  **return** i;  } } |

1. **화면10**

* **레이아웃 구현**

|  |  |
| --- | --- |
| 화면(UI) | 설명 |
|  | - 퍼즐 게임의 실행 화면입니다.  - 12개의 이미지 버튼과 1개의 버튼으로 구성되어 있습니다.  - 12개의 이미지 버튼은 각각 퍼즐 조각을 입력하고 입력된 값을 나타내는 버튼입니다.  - 하단에 위치한 버튼은 도안을 선택하는 리스트 목록으로 돌아가는 버튼입니다. |
| coloring\_page.xml | |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:orientation="vertical" android:layout\_width="match\_parent"  android:layout\_height="match\_parent"**>    <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:orientation="vertical"**>  <**LinearLayout  android:orientation="horizontal"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1"**>  <**ImageButton  android:background="#ffebe8"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:scaleType="fitXY"  android:src="@drawable/foot"  android:id="@+id/puzzleButton1"**/>  <**ImageButton  android:background="#FFFFFF"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:src="@drawable/foot"  android:scaleType="fitXY"  android:id="@+id/puzzleButton2"**/>  <**ImageButton  android:background="#ffebe8"  android:src="@drawable/foot"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:scaleType="fitXY"  android:id="@+id/puzzleButton3"**/>  </**LinearLayout**>  <**LinearLayout  android:orientation="horizontal"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1"**>  <**ImageButton  android:background="#FFFFFF"  android:src="@drawable/foot"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:scaleType="fitXY"  android:id="@+id/puzzleButton4"**/>  <**ImageButton  android:background="#ffebe8"  android:src="@drawable/foot"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:scaleType="fitXY"  android:id="@+id/puzzleButton5"**/>  <**ImageButton  android:background="#FFFFFF"  android:src="@drawable/foot"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:scaleType="fitXY"  android:id="@+id/puzzleButton6"** />  </**LinearLayout**>  <**LinearLayout  android:orientation="horizontal"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1"**>  <**ImageButton  android:background="#ffebe8"  android:src="@drawable/foot"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:scaleType="fitXY"  android:id="@+id/puzzleButton7"**/>  <**ImageButton  android:background="#FFFFFF"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:scaleType="fitXY"  android:id="@+id/puzzleButton8"  android:src="@drawable/foot"**/>  <**ImageButton  android:background="#ffebe8"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:scaleType="fitXY"  android:id="@+id/puzzleButton9"  android:src="@drawable/foot"**/>  </**LinearLayout**>  <**LinearLayout  android:orientation="horizontal"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1"**>  <**ImageButton  android:background="#FFFFFF"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:scaleType="fitXY"  android:id="@+id/puzzleButton10"  android:src="@drawable/foot"**/>  <**ImageButton  android:background="#ffebe8"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:scaleType="fitXY"  android:id="@+id/puzzleButton11"  android:src="@drawable/foot"**/>  <**ImageButton  android:background="#FFFFFF"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:scaleType="fitXY"  android:id="@+id/puzzleButton12"  android:src="@drawable/foot"**/>  </**LinearLayout**>  </**LinearLayout**>   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="7"**>   <**Button  android:background="#fc9d9a"  android:textStyle="bold"  android:textColor="#FFFFFF"  android:textSize="15dp"  android:id="@+id/backButton"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:text="목록 화면으로\n돌아가기"**/>  </**LinearLayout**>  </**LinearLayout**> | |

|  |  |
| --- | --- |
| 화면(UI) | 설명 |
|  | - 퍼즐게임 실행화면에 위치한 버튼을 클릭할 경우에 나타나는 대화상자로 리니어 레이아웃 안에 이미지뷰로 구성되어 있습니다.  -대화상자에 나타나는 이미지는 퍼즐 뽑기 기능을 이용하여 사용자가 획득한 퍼즐들 중 현재 진행중인 퍼즐 도안에 해당하는 조각들만이 대화상자 안에 위치한 리니어 레이아웃에 동적으로 생성되어 보여집니다. |
| puzzle\_dialog.xml | |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:orientation="vertical" android:layout\_width="match\_parent"  android:layout\_height="200dp"**>   <**HorizontalScrollView  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"** >   <**LinearLayout  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:orientation="horizontal"  android:id="@+id/dialog\_linear"** >  <**ImageView  android:layout\_width="130dp"  android:scaleType="centerInside"  android:layout\_height="150dp"  android:src="@drawable/foot"  android:id="@+id/foot\_image"**/>   </**LinearLayout**>  </**HorizontalScrollView**> </**LinearLayout**> | |

* **기능 구현**

|  |
| --- |
| **ColoringActivity.java** |
| **package** com.localhost.jooyeon.final\_project;  **import** android.content.Context; **import** android.content.Intent; **import** android.os.Bundle; **import** android.support.v7.app.AlertDialog; **import** android.support.v7.app.AppCompatActivity; **import** android.view.View; **import** android.widget.Button; **import** android.widget.ImageButton; **import** android.widget.ImageView; **import** android.widget.LinearLayout;  **import** com.google.firebase.database.DataSnapshot; **import** com.google.firebase.database.DatabaseError; **import** com.google.firebase.database.DatabaseReference; **import** com.google.firebase.database.FirebaseDatabase; **import** com.google.firebase.database.ValueEventListener; **import** com.tsengvn.typekit.TypekitContextWrapper;  **import** java.util.HashMap;  **import static** com.localhost.jooyeon.final\_project.GoalActivity.*sharedPreferences*;  **public class** ColoringActivity **extends** AppCompatActivity {   *//파이어베이스 연동 및 레퍼런스* DatabaseReference **mRootRef** = FirebaseDatabase.*getInstance*().getReference();  DatabaseReference **mpuzzleGameRef**;  DatabaseReference **mUserPuzzle**;  DatabaseReference **mUserSetPuzzle**;   **int**[] **arr** = **new int**[12];  HashMap **userPuzzleMap** = **null**;  **int**[][] **imageResId** = {  {R.drawable.***piece1\_1***,R.drawable.***piece1\_2***,R.drawable.***piece1\_3***,R.drawable.***piece1\_4***,R.drawable.***piece1\_5***,R.drawable.***piece1\_6***,  R.drawable.***piece1\_7***,R.drawable.***piece1\_8***,R.drawable.***piece1\_9***,R.drawable.***piece1\_10***,R.drawable.***piece1\_11***,R.drawable.***piece1\_12***},   {R.drawable.***piece2\_1***,R.drawable.***piece2\_2***,R.drawable.***piece2\_3***,R.drawable.***piece2\_4***,R.drawable.***piece2\_5***,R.drawable.***piece2\_6***,  R.drawable.piece2\_7,R.drawable.***piece2\_8***,R.drawable.***piece2\_9***,R.drawable.***piece2\_10***,R.drawable.***piece2\_11***,R.drawable.***piece2\_12***},   {R.drawable.***piece3\_1***,R.drawable.***piece3\_2***,R.drawable.***piece3\_3***,R.drawable.***piece3\_4***,R.drawable.***piece3\_5***,R.drawable.***piece3\_6***,  R.drawable.***piece3\_7***,R.drawable.***piece3\_8***,R.drawable.***piece3\_9***,R.drawable.***piece3\_10***,R.drawable.***piece3\_11***,R.drawable.***piece3\_12***},   {R.drawable.***piece4\_1***,R.drawable.***piece4\_2***,R.drawable.***piece4\_3***,R.drawable.***piece4\_4***,R.drawable.***piece4\_5***,R.drawable.***piece4\_6***,  R.drawable.***piece4\_7***,R.drawable.***piece4\_8***,R.drawable.***piece4\_9***,R.drawable.***piece4\_10***,R.drawable.***piece4\_11***,R.drawable.***piece4\_12***}  };  @Override  **protected void** onStart() {  **super**.onStart();   *//사용자가 갖고 있는 퍼즐의 종류 파악* **mUserPuzzle**.addValueEventListener(**new** ValueEventListener() {  @Override  **public void** onDataChange(DataSnapshot dataSnapshot) {  HashMap map = (HashMap)dataSnapshot.getValue();  **userPuzzleMap** = map;   **for**(**int** i = 0 ; i < 12 ; i++)  {  Long userPuzzle = (Long)map.get(**"puzzle"**+Integer.*toString*(i+1));  **if**(userPuzzle==1){  **arr**[i] = 1;  }  }   }  @Override  **public void** onCancelled(DatabaseError databaseError) {   }  });   *//이전에 설정한 퍼즐 배치를 가져옴* **mUserSetPuzzle**.addValueEventListener(**new** ValueEventListener() {  @Override  **public void** onDataChange(DataSnapshot dataSnapshot) {  HashMap map = (HashMap)dataSnapshot.getValue();  **userPuzzleMap** = map;   **for**(**int** i = 0 ; i < 12 ; i++)  {  Long userPuzzle = (Long)map.get(**"puzzle"**+Integer.*toString*(i+1));  **if**(userPuzzle!=0){  **imgArr**[i].setImageResource(userPuzzle.intValue());  }  }   }   @Override  **public void** onCancelled(DatabaseError databaseError) {   }  });  }   **private** Integer[] **mImageIds** = {  R.drawable.***puzzle1***,  R.drawable.***puzzle2***,  R.drawable.***puzzle3***,  R.drawable.***puzzle4*** };  Integer **position**;  ImageView [] **imgArr** = **new** ImageView[12];  String **user\_id**;  @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);   *sharedPreferences* = getSharedPreferences(**"USER\_INFO"**, ***MODE\_PRIVATE***);  **user\_id** = String.*valueOf*(*sharedPreferences*.getLong(**"user\_id"**, 0));    setContentView(R.layout.***coloring\_page***);  Intent intent = getIntent();  **position** = intent.getIntExtra(**"position"**,0)+1;  **mpuzzleGameRef** = **mRootRef**.child(**user\_id**+**"/puzzleGame/picture"**+**position**.toString());   Button backButton = (Button)findViewById(R.id.***backButton***);  **final** ImageButton puzzleButton1 = (ImageButton)findViewById(R.id.***puzzleButton1***);  **final** ImageButton puzzleButton2 = (ImageButton)findViewById(R.id.***puzzleButton2***);  **final** ImageButton puzzleButton3 = (ImageButton)findViewById(R.id.***puzzleButton3***);  **final** ImageButton puzzleButton4 = (ImageButton)findViewById(R.id.***puzzleButton4***);  **final** ImageButton puzzleButton5 = (ImageButton)findViewById(R.id.***puzzleButton5***);  **final** ImageButton puzzleButton6 = (ImageButton)findViewById(R.id.***puzzleButton6***);  **final** ImageButton puzzleButton7 = (ImageButton)findViewById(R.id.***puzzleButton7***);  **final** ImageButton puzzleButton8 = (ImageButton)findViewById(R.id.***puzzleButton8***);  **final** ImageButton puzzleButton9 = (ImageButton)findViewById(R.id.***puzzleButton9***);  **final** ImageButton puzzleButton10 = (ImageButton)findViewById(R.id.***puzzleButton10***);  **final** ImageButton puzzleButton11 = (ImageButton)findViewById(R.id.***puzzleButton11***);  **final** ImageButton puzzleButton12 = (ImageButton)findViewById(R.id.***puzzleButton12***);  **position**--;  **imgArr**[0] = puzzleButton1;**imgArr**[1] = puzzleButton2;**imgArr**[2] = puzzleButton3;  **imgArr**[3] = puzzleButton4;**imgArr**[4] = puzzleButton5;**imgArr**[5] = puzzleButton6;  **imgArr**[6] = puzzleButton7;**imgArr**[7] = puzzleButton8;**imgArr**[8] = puzzleButton9;  **imgArr**[9] = puzzleButton10;**imgArr**[10] = puzzleButton11;**imgArr**[11] = puzzleButton12;   **mUserPuzzle** = **mpuzzleGameRef**.child(**"userPuzzle"**);  **mUserSetPuzzle** = **mpuzzleGameRef**.child(**"userSetPuzzle"**);  *//퍼즐 클릭 시 대화상자에 사용자가 갖고 있는 퍼즐 종류를 출력 후 선택한 퍼즐을 퍼즐버튼1에 위치시키고 그 값을 파이어베이스에 저장* puzzleButton1.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  AlertDialog.Builder ab = **new** AlertDialog.Builder(ColoringActivity.**this**);  View view = View.*inflate*(ColoringActivity.**this**,R.layout.***puzzle\_dialog***,**null**);  **final** AlertDialog ad = ab.create();  LinearLayout dialog\_linear = (LinearLayout)view.findViewById(R.id.***dialog\_linear***);  **for**(**int** i = 0 ; i < 12 ; i++){  **if**(**arr**[i]==1){  **final** ImageView imageView = **new** ImageView(ColoringActivity.**this**);  imageView.setImageResource(**imageResId**[**position**][i]);  imageView.setScaleType(ImageView.ScaleType.***FIT\_XY***);  LinearLayout.LayoutParams lp = **new** LinearLayout.LayoutParams(300,300);  lp.setMargins(5,0,0,5);  imageView.setLayoutParams(lp);  dialog\_linear.addView(imageView);  **final int** index = i;  imageView.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  puzzleButton1.setImageResource(**imageResId**[**position**][index]);  ad.dismiss();  }  });  }  }   ad.setView(view);  ad.show();  }  });   *//퍼즐 클릭 시 대화상자에 사용자가 갖고 있는 퍼즐 종류를 출력 후 선택한 퍼즐을 퍼즐버튼2에 위치시키고 그 값을 파이어베이스에 저장* puzzleButton2.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  AlertDialog.Builder ab = **new** AlertDialog.Builder(ColoringActivity.**this**);  View view = View.*inflate*(ColoringActivity.**this**,R.layout.***puzzle\_dialog***,**null**);  **final** AlertDialog ad = ab.create();  LinearLayout dialog\_linear = (LinearLayout)view.findViewById(R.id.***dialog\_linear***);  **for**(**int** i = 0 ; i < 12 ; i++){  **if**(**arr**[i]==1){  **final** ImageView imageView = **new** ImageView(ColoringActivity.**this**);  imageView.setImageResource(**imageResId**[**position**][i]);  imageView.setScaleType(ImageView.ScaleType.***FIT\_XY***);  LinearLayout.LayoutParams lp = **new** LinearLayout.LayoutParams(300,300);  lp.setMargins(5,0,0,5);  imageView.setLayoutParams(lp);  dialog\_linear.addView(imageView);  **final int** index = i;  imageView.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **mpuzzleGameRef**.child(**"userSetPuzzle/puzzle2"**).setValue(**imageResId**[**position**][index]);  puzzleButton2.setImageResource(**imageResId**[**position**][index]);   ad.dismiss();  }  });  }  }  ImageView foot\_image = (ImageView)view.findViewById(R.id.***foot\_image***);   foot\_image.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **mpuzzleGameRef**.child(**"userSetPuzzle/puzzle2"**).setValue(R.drawable.***foot***);  puzzleButton2.setImageResource(R.drawable.***foot***);  ad.dismiss();  }  });    ad.setView(view);  ad.show();  }  });   *//퍼즐 클릭 시 대화상자에 사용자가 갖고 있는 퍼즐 종류를 출력 후 선택한 퍼즐을 퍼즐버튼3에 위치시키고 그 값을 파이어베이스에 저장* puzzleButton3.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  AlertDialog.Builder ab = **new** AlertDialog.Builder(ColoringActivity.**this**);  View view = View.*inflate*(ColoringActivity.**this**,R.layout.***puzzle\_dialog***,**null**);  **final** AlertDialog ad = ab.create();  LinearLayout dialog\_linear = (LinearLayout)view.findViewById(R.id.***dialog\_linear***);  **for**(**int** i = 0 ; i < 12 ; i++){  **if**(**arr**[i]==1){  **final** ImageView imageView = **new** ImageView(ColoringActivity.**this**);  imageView.setImageResource(**imageResId**[**position**][i]);  imageView.setScaleType(ImageView.ScaleType.***FIT\_XY***);  LinearLayout.LayoutParams lp = **new** LinearLayout.LayoutParams(300,300);  lp.setMargins(5,0,0,5);  imageView.setLayoutParams(lp);  dialog\_linear.addView(imageView);  **final int** index = i;  imageView.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **mpuzzleGameRef**.child(**"userSetPuzzle/puzzle3"**).setValue(**imageResId**[**position**][index]);  puzzleButton3.setImageResource(**imageResId**[**position**][index]);  ad.dismiss();  }  });  }  }  ImageView foot\_image = (ImageView)view.findViewById(R.id.***foot\_image***);   foot\_image.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **mpuzzleGameRef**.child(**"userSetPuzzle/puzzle3"**).setValue(R.drawable.***foot***);  puzzleButton3.setImageResource(R.drawable.***foot***);  ad.dismiss();  }  });   ad.setView(view);  ad.show();  }  });   *//퍼즐 클릭 시 대화상자에 사용자가 갖고 있는 퍼즐 종류를 출력 후 선택한 퍼즐을 퍼즐버튼4에 위치시키고 그 값을 파이어베이스에 저장* puzzleButton4.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  AlertDialog.Builder ab = **new** AlertDialog.Builder(ColoringActivity.**this**);  View view = View.*inflate*(ColoringActivity.**this**,R.layout.***puzzle\_dialog***,**null**);  **final** AlertDialog ad = ab.create();  LinearLayout dialog\_linear = (LinearLayout)view.findViewById(R.id.***dialog\_linear***);  **for**(**int** i = 0 ; i < 12 ; i++){  **if**(**arr**[i]==1){  **final** ImageView imageView = **new** ImageView(ColoringActivity.**this**);  imageView.setImageResource(**imageResId**[**position**][i]);  imageView.setScaleType(ImageView.ScaleType.***FIT\_XY***);  LinearLayout.LayoutParams lp = **new** LinearLayout.LayoutParams(300,300);  lp.setMargins(5,0,0,5);  imageView.setLayoutParams(lp);  dialog\_linear.addView(imageView);  **final int** index = i;  imageView.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **mpuzzleGameRef**.child(**"userSetPuzzle/puzzle4"**).setValue(**imageResId**[**position**][index]);  puzzleButton4.setImageResource(**imageResId**[**position**][index]);  ad.dismiss();  }  });  }  }  ImageView foot\_image = (ImageView)view.findViewById(R.id.***foot\_image***);   foot\_image.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **mpuzzleGameRef**.child(**"userSetPuzzle/puzzle4"**).setValue(R.drawable.***foot***);  puzzleButton4.setImageResource(R.drawable.***foot***);  ad.dismiss();  }  });   ad.setView(view);  ad.show();  }  });   *//퍼즐 클릭 시 대화상자에 사용자가 갖고 있는 퍼즐 종류를 출력 후 선택한 퍼즐을 퍼즐버튼5에 위치시키고 그 값을 파이어베이스에 저장* puzzleButton5.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  AlertDialog.Builder ab = **new** AlertDialog.Builder(ColoringActivity.**this**);  View view = View.*inflate*(ColoringActivity.**this**,R.layout.***puzzle\_dialog***,**null**);  **final** AlertDialog ad = ab.create();  LinearLayout dialog\_linear = (LinearLayout)view.findViewById(R.id.***dialog\_linear***);  **for**(**int** i = 0 ; i < 12 ; i++){  **if**(**arr**[i]==1){  **final** ImageView imageView = **new** ImageView(ColoringActivity.**this**);  imageView.setImageResource(**imageResId**[**position**][i]);  imageView.setScaleType(ImageView.ScaleType.***FIT\_XY***);  LinearLayout.LayoutParams lp = **new** LinearLayout.LayoutParams(300,300);  lp.setMargins(5,0,0,5);  imageView.setLayoutParams(lp);  dialog\_linear.addView(imageView);  **final int** index = i;  imageView.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **mpuzzleGameRef**.child(**"userSetPuzzle/puzzle5"**).setValue(**imageResId**[**position**][index]);  puzzleButton5.setImageResource(**imageResId**[**position**][index]);  ad.dismiss();  }  });  }  }  ImageView foot\_image = (ImageView)view.findViewById(R.id.***foot\_image***);   foot\_image.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **mpuzzleGameRef**.child(**"userSetPuzzle/puzzle5"**).setValue(R.drawable.***foot***);  puzzleButton5.setImageResource(R.drawable.***foot***);  ad.dismiss();  }  });   ad.setView(view);  ad.show();  }  });   *//퍼즐 클릭 시 대화상자에 사용자가 갖고 있는 퍼즐 종류를 출력 후 선택한 퍼즐을 퍼즐버튼6에 위치시키고 그 값을 파이어베이스에 저장* puzzleButton6.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  AlertDialog.Builder ab = **new** AlertDialog.Builder(ColoringActivity.**this**);  View view = View.*inflate*(ColoringActivity.**this**,R.layout.***puzzle\_dialog***,**null**);  **final** AlertDialog ad = ab.create();  LinearLayout dialog\_linear = (LinearLayout)view.findViewById(R.id.***dialog\_linear***);  **for**(**int** i = 0 ; i < 12 ; i++){  **if**(**arr**[i]==1){  **final** ImageView imageView = **new** ImageView(ColoringActivity.**this**);  imageView.setImageResource(**imageResId**[**position**][i]);  imageView.setScaleType(ImageView.ScaleType.***FIT\_XY***);  LinearLayout.LayoutParams lp = **new** LinearLayout.LayoutParams(300,300);  lp.setMargins(5,0,0,5);  imageView.setLayoutParams(lp);  dialog\_linear.addView(imageView);  **final int** index = i;  imageView.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **mpuzzleGameRef**.child(**"userSetPuzzle/puzzle6"**).setValue(**imageResId**[**position**][index]);  puzzleButton6.setImageResource(**imageResId**[**position**][index]);  ad.dismiss();  }  });  }  }  ImageView foot\_image = (ImageView)view.findViewById(R.id.***foot\_image***);   foot\_image.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **mpuzzleGameRef**.child(**"userSetPuzzle/puzzle6"**).setValue(R.drawable.***foot***);  puzzleButton6.setImageResource(R.drawable.***foot***);  ad.dismiss();  }  });   ad.setView(view);  ad.show();  }  });   *//퍼즐 클릭 시 대화상자에 사용자가 갖고 있는 퍼즐 종류를 출력 후 선택한 퍼즐을 퍼즐버튼7에 위치시키고 그 값을 파이어베이스에 저장* puzzleButton7.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  AlertDialog.Builder ab = **new** AlertDialog.Builder(ColoringActivity.**this**);  View view = View.*inflate*(ColoringActivity.**this**,R.layout.***puzzle\_dialog***,**null**);  **final** AlertDialog ad = ab.create();  LinearLayout dialog\_linear = (LinearLayout)view.findViewById(R.id.***dialog\_linear***);  **for**(**int** i = 0 ; i < 12 ; i++){  **if**(**arr**[i]==1){  **final** ImageView imageView = **new** ImageView(ColoringActivity.**this**);  imageView.setImageResource(**imageResId**[**position**][i]);  imageView.setScaleType(ImageView.ScaleType.***FIT\_XY***);  LinearLayout.LayoutParams lp = **new** LinearLayout.LayoutParams(300,300);  lp.setMargins(5,0,0,5);  imageView.setLayoutParams(lp);  dialog\_linear.addView(imageView);  **final int** index = i;  imageView.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **mpuzzleGameRef**.child(**"userSetPuzzle/puzzle7"**).setValue(**imageResId**[**position**][index]);  puzzleButton7.setImageResource(**imageResId**[**position**][index]);  ad.dismiss();  }  });  }  }  ImageView foot\_image = (ImageView)view.findViewById(R.id.***foot\_image***);   foot\_image.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **mpuzzleGameRef**.child(**"userSetPuzzle/puzzle7"**).setValue(R.drawable.***foot***);  puzzleButton7.setImageResource(R.drawable.***foot***);  ad.dismiss();  }  });   ad.setView(view);  ad.show();  }  });   *//퍼즐 클릭 시 대화상자에 사용자가 갖고 있는 퍼즐 종류를 출력 후 선택한 퍼즐을 퍼즐버튼8에 위치시키고 그 값을 파이어베이스에 저장* puzzleButton8.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  AlertDialog.Builder ab = **new** AlertDialog.Builder(ColoringActivity.**this**);  View view = View.*inflate*(ColoringActivity.**this**,R.layout.***puzzle\_dialog***,**null**);  **final** AlertDialog ad = ab.create();  LinearLayout dialog\_linear = (LinearLayout)view.findViewById(R.id.***dialog\_linear***);  **for**(**int** i = 0 ; i < 12 ; i++){  **if**(**arr**[i]==1){  **final** ImageView imageView = **new** ImageView(ColoringActivity.**this**);  imageView.setImageResource(**imageResId**[**position**][i]);  imageView.setScaleType(ImageView.ScaleType.***FIT\_XY***);  LinearLayout.LayoutParams lp = **new** LinearLayout.LayoutParams(300,300);  lp.setMargins(5,0,0,5);  imageView.setLayoutParams(lp);  dialog\_linear.addView(imageView);  **final int** index = i;  imageView.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **mpuzzleGameRef**.child(**"userSetPuzzle/puzzle8"**).setValue(**imageResId**[**position**][index]);  puzzleButton8.setImageResource(**imageResId**[**position**][index]);  ad.dismiss();  }  });  }  }  ImageView foot\_image = (ImageView)view.findViewById(R.id.***foot\_image***);   foot\_image.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **mpuzzleGameRef**.child(**"userSetPuzzle/puzzle8"**).setValue(R.drawable.***foot***);  puzzleButton8.setImageResource(R.drawable.***foot***);  ad.dismiss();  }  });   ad.setView(view);  ad.show();  }  });   *//퍼즐 클릭 시 대화상자에 사용자가 갖고 있는 퍼즐 종류를 출력 후 선택한 퍼즐을 퍼즐버튼9에 위치시키고 그 값을 파이어베이스에 저장* puzzleButton9.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  AlertDialog.Builder ab = **new** AlertDialog.Builder(ColoringActivity.**this**);  View view = View.*inflate*(ColoringActivity.**this**,R.layout.***puzzle\_dialog***,**null**);  **final** AlertDialog ad = ab.create();  LinearLayout dialog\_linear = (LinearLayout)view.findViewById(R.id.***dialog\_linear***);  **for**(**int** i = 0 ; i < 12 ; i++){  **if**(**arr**[i]==1){  **final** ImageView imageView = **new** ImageView(ColoringActivity.**this**);  imageView.setImageResource(**imageResId**[**position**][i]);  imageView.setScaleType(ImageView.ScaleType.***FIT\_XY***);  LinearLayout.LayoutParams lp = **new** LinearLayout.LayoutParams(300,300);  lp.setMargins(5,0,0,5);  imageView.setLayoutParams(lp);  dialog\_linear.addView(imageView);  **final int** index = i;  imageView.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **mpuzzleGameRef**.child(**"userSetPuzzle/puzzle9"**).setValue(**imageResId**[**position**][index]);  puzzleButton9.setImageResource(**imageResId**[**position**][index]);  ad.dismiss();  }  });  }  }  ImageView foot\_image = (ImageView)view.findViewById(R.id.***foot\_image***);   foot\_image.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **mpuzzleGameRef**.child(**"userSetPuzzle/puzzle9"**).setValue(R.drawable.***foot***);  puzzleButton9.setImageResource(R.drawable.***foot***);  ad.dismiss();  }  });   ad.setView(view);  ad.show();  }  });   *//퍼즐 클릭 시 대화상자에 사용자가 갖고 있는 퍼즐 종류를 출력 후 선택한 퍼즐을 퍼즐버튼10에 위치시키고 그 값을 파이어베이스에 저장* puzzleButton10.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  AlertDialog.Builder ab = **new** AlertDialog.Builder(ColoringActivity.**this**);  View view = View.*inflate*(ColoringActivity.**this**,R.layout.***puzzle\_dialog***,**null**);  **final** AlertDialog ad = ab.create();  LinearLayout dialog\_linear = (LinearLayout)view.findViewById(R.id.***dialog\_linear***);  **for**(**int** i = 0 ; i < 12 ; i++){  **if**(**arr**[i]==1){  **final** ImageView imageView = **new** ImageView(ColoringActivity.**this**);  imageView.setImageResource(**imageResId**[**position**][i]);  imageView.setScaleType(ImageView.ScaleType.***FIT\_XY***);  LinearLayout.LayoutParams lp = **new** LinearLayout.LayoutParams(300,300);  lp.setMargins(5,0,0,5);  imageView.setLayoutParams(lp);  dialog\_linear.addView(imageView);  **final int** index = i;  imageView.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **mpuzzleGameRef**.child(**"userSetPuzzle/puzzle10"**).setValue(**imageResId**[**position**][index]);  puzzleButton10.setImageResource(**imageResId**[**position**][index]);  ad.dismiss();  }  });  }  }  ImageView foot\_image = (ImageView)view.findViewById(R.id.***foot\_image***);   foot\_image.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **mpuzzleGameRef**.child(**"userSetPuzzle/puzzle10"**).setValue(R.drawable.***foot***);  puzzleButton10.setImageResource(R.drawable.***foot***);  ad.dismiss();  }  });   ad.setView(view);  ad.show();  }  });   *//퍼즐 클릭 시 대화상자에 사용자가 갖고 있는 퍼즐 종류를 출력 후 선택한 퍼즐을 퍼즐버튼11에 위치시키고 그 값을 파이어베이스에 저장* puzzleButton11.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  AlertDialog.Builder ab = **new** AlertDialog.Builder(ColoringActivity.**this**);  View view = View.*inflate*(ColoringActivity.**this**,R.layout.***puzzle\_dialog***,**null**);  **final** AlertDialog ad = ab.create();  LinearLayout dialog\_linear = (LinearLayout)view.findViewById(R.id.***dialog\_linear***);  **for**(**int** i = 0 ; i < 12 ; i++){  **if**(**arr**[i]==1){  **final** ImageView imageView = **new** ImageView(ColoringActivity.**this**);  imageView.setImageResource(**imageResId**[**position**][i]);  imageView.setScaleType(ImageView.ScaleType.***FIT\_XY***);  LinearLayout.LayoutParams lp = **new** LinearLayout.LayoutParams(300,300);  lp.setMargins(5,0,0,5);  imageView.setLayoutParams(lp);  dialog\_linear.addView(imageView);  **final int** index = i;  imageView.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **mpuzzleGameRef**.child(**"userSetPuzzle/puzzle11"**).setValue(**imageResId**[**position**][index]);  puzzleButton11.setImageResource(**imageResId**[**position**][index]);  ad.dismiss();  }  });  }  }  ImageView foot\_image = (ImageView)view.findViewById(R.id.***foot\_image***);   foot\_image.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **mpuzzleGameRef**.child(**"userSetPuzzle/puzzle11"**).setValue(R.drawable.***foot***);  puzzleButton11.setImageResource(R.drawable.***foot***);  ad.dismiss();  }  });   ad.setView(view);  ad.show();  }  });   *//퍼즐 클릭 시 대화상자에 사용자가 갖고 있는 퍼즐 종류를 출력 후 선택한 퍼즐을 퍼즐버튼12에 위치시키고 그 값을 파이어베이스에 저장* puzzleButton12.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  AlertDialog.Builder ab = **new** AlertDialog.Builder(ColoringActivity.**this**);  View view = View.*inflate*(ColoringActivity.**this**,R.layout.***puzzle\_dialog***,**null**);  **final** AlertDialog ad = ab.create();  LinearLayout dialog\_linear = (LinearLayout)view.findViewById(R.id.***dialog\_linear***);  **for**(**int** i = 0 ; i < 12 ; i++){  **if**(**arr**[i]==1){  **final** ImageView imageView = **new** ImageView(ColoringActivity.**this**);  imageView.setImageResource(**imageResId**[**position**][i]);  imageView.setScaleType(ImageView.ScaleType.***FIT\_XY***);  LinearLayout.LayoutParams lp = **new** LinearLayout.LayoutParams(300,300);  lp.setMargins(5,0,0,5);  imageView.setLayoutParams(lp);  dialog\_linear.addView(imageView);  **final int** index = i;  imageView.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **mpuzzleGameRef**.child(**"userSetPuzzle/puzzle12"**).setValue(**imageResId**[**position**][index]);  puzzleButton12.setImageResource(**imageResId**[**position**][index]);  ad.dismiss();  }  });  }  }  ImageView foot\_image = (ImageView)view.findViewById(R.id.***foot\_image***);   foot\_image.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **mpuzzleGameRef**.child(**"userSetPuzzle/puzzle12"**).setValue(R.drawable.***foot***);  puzzleButton12.setImageResource(R.drawable.***foot***);  ad.dismiss();  }  });   ad.setView(view);  ad.show();  }  });   *//뒤로가기 버튼 누를 경우 뒤로가기 기능 구현* backButton.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  onBackPressed();  }  });  }  @Override  **protected void** attachBaseContext(Context newBase) {  **super**.attachBaseContext(TypekitContextWrapper.*wrap*(newBase));  } } |

# 나의 앱 만들기 후기

* **박주연, 이홍주**

|  |  |
| --- | --- |
|  | * **나의 소감 (박주연)**   처음 제작을 시작하였을 때는 하루 빨리 완성해서 이것을 이용하여 실제로 내가 사용함으로써 다이어트에 성공하고 싶었지만 제출 하루 전날 완성한 관계로 사용할 시간이 없었다는 점이 가장 아쉽습니다. 하지만 서비스와 파이어베이스 등 평소에 써보지 못했던 기술을 사용해보면서 많은 것을 배웠습니다.   * **나의 소감(이홍주)**   방학 때 안드로이드 스터디를 했기 때문에 수업을 잘 따라갈 것이라 생각했지만 내 자신을 너무 과대평가 했었던 것 같았다. 생각보다 아는 것도 많았지만 그만큼 모르는 것도 많았기 때문에 더욱 공부가 필요하다는 것을 느끼게 되었다.  처음으로 내가 원하는 어플을 제작한다는 것에 즐거움을 느꼈고, 원하는 기능을 다 구현하지 못한 점이 아쉬웠다. 방학 중에 이 어플리케이션을 가지고 더욱 기능을 추가하면서 안드로이드 실력을 쌓을 것이다. |
| * **모바일 프로그래밍 강의를 통해 얻은 것 (박주연)**   모바일 프로그래밍을 하면서 서비스, 위젯 생성 등 생소하지만 어플리케이션 제작을 위해서 필요한 기능을 배웠습니다. 개념과 실습을 동시에 배울 수 있는 수업이었습니다. 뿐만 아니라 라즈베리파이와 모바일의 연동 수업을 준비하면서 보통같으면 접하지 못하였을 IoT까지 접하는 시간이 되었습니다. 라즈베리파이에 대한 수업을 진행하면서 실습실 준비에 대한 중요함과 나에게 제공된 시간 등을 고려해야 한다는 점도 배우며, 개발에 대한 경험과 강의에 대한 경험을 느낄 수 있었습니다.   * **모바일 프로그래밍 강의를 통해 얻은 것(이홍주)**   팀원 주연이와 함께 역할을 나누고, 제가 맡은 부분을 끝까지 해내야 한다는 것에서 책임감을 느끼게 되었습니다. 혼자가 아닌 둘이서 하다 보니 서로가 해낸 부분을 서로가 알려주고, 모르는 부분이 있으면 알려주고 하는 과정에서 실력을 더 쌓고, 갈등없이 원만하게 할 수 있었습니다.  모바일 프로그래밍 강의를 통해서 저의 안드로이드 지식을 더욱 향상시킬 수 있었고, UI를 이제 능숙하게 짤 수 있게 되었습니다. Java 언어에 대한 이해도도 높아졌고, 앞으로 Java의 디자인 패턴을 공부해서 어플리케이션 개발 실력을 더욱 쌓을 것입니다. | |