Filter.java

//TimeTable.java로 값을 넘겨주는 부분

package com.example.it103\_32.cnt;

/\*\*

\* Created by USER on 2016-11-16.

\*/

public class Filter {

// 받은 교시를 int값으로 넘겨주는 부분 (실패시 -1)

public int Filter\_time(String time){

if(time.equals("1교시"))

return 0;

else if(time.equals("2교시"))

return 1;

else if(time.equals("3교시"))

return 2;

else if(time.equals("4교시"))

return 3;

else if(time.equals("5교시"))

return 4;

else if(time.equals("6교시"))

return 5;

else if(time.equals("7교시"))

return 6;

else if(time.equals("8교시"))

return 7;

else if(time.equals("9교시"))

return 8;

else if(time.equals("10교시"))

return 9;

else if(time.equals("11교시"))

return 10;

else if(time.equals("12교시"))

return 11;

else if(time.equals("13교시"))

return 12;

else if(time.equals("14교시"))

return 13;

else if(time.equals("15교시"))

return 14;

else if(time.equals("16교시"))

return 15;

else if(time.equals("17교시"))

return 16;

else if(time.equals("18교시"))

return 17;

else if(time.equals("19교시"))

return 18;

else if(time.equals("20교시"))

return 19;

else

return -1;

}

// 받은 요일을 int값으로 넘겨주는 부분 (실패시 -1)

public int Filter\_day(String day){

if(day.equals("월요일"))

return 0;

else if(day.equals("화요일"))

return 1;

else if(day.equals("수요일"))

return 2;

else if(day.equals("목요일"))

return 3;

else if(day.equals("금요일"))

return 4;

else if(day.equals("토요일"))

return 5;

else if(day.equals("일요일"))

return 6;

else

return -1;

}

}

Main.java

package com.example.it103\_32.cnt;

import android.app.Activity;

import android.content.Intent;

import android.graphics.Color;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.LinearLayout;

public class Main extends Activity { // C.N.T의 Main 화면

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

Button btnnavi = (Button)findViewById(R.id.btnNavi);

btnnavi.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

startActivity(new Intent(getApplication(),Menu.class));

// 네비게이션 버튼 기능

}

});

Button btntime = (Button)findViewById(R.id.btnTime);

btntime.setOnClickListener(new View.OnClickListener(){

public void onClick(View v)

{

startActivity(new Intent(getApplication(),TimeTable.class));

// Timetable 버튼 기능

//

}

});

}

}

Timetable.java

package com.example.it103\_32.cnt;

import android.app.\*;

import android.content.\*;

import android.content.res.AssetManager;

import android.database.\*;

import android.database.sqlite.\*;

import android.graphics.\*;

import android.graphics.drawable.\*;

import android.os.\*;

import android.view.\*;

import android.view.Menu;

import android.view.View.OnClickListener;

import android.widget.\*;

import java.io.File;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.InputStream;

public class TimeTable extends Activity implements OnClickListener{

// TimeTable 클래스

public Intent intent, get\_intent; // 보내고 받는 인텐트

private Typeface font; // 폰트

private TableRow row\_1, row\_2, row\_3, row\_4, row\_5, row\_6, row\_7, row\_8, row\_9, row\_10, row\_11, row\_12; // 교시에 대한 정보

private TextView mon, tue, wed, thu, fri; // 요일에 대한 정보

private TextView what;

private TextView time\_1, time\_2, time\_3, time\_4, time\_5, time\_6, time\_7, time\_8, time\_9, time\_10, time\_11, time\_12; // 교시와 요일에 해당하는 정보

private TextView[][] day = new TextView[12][5];

// 해당 정보를 표시하는 contents

private TextView alert\_text; // 다이얼로그의 입력 텍스트

private EditText alert\_edit; // 다이얼로그의 텍스트 바

private Database db = null; // Database 클래스 정보

private SQLiteDatabase sql\_db; // DB 정보

private Cursor cursor; // select 쿼리문 위한 cursor

private static int select\_num; // 시 설정에서 몇 교시인지를 선택한 int변수

private static String one, two, three, four, five, six, seven, eight, nine, ten, tenone, tentwo; // 각 교시에 설정된 시간을 담는 변수

private AlertDialog.Builder save\_message\_box, update\_message\_box, saveok\_message\_box;

// 다이얼로그

private Context ctx;

// Context 변수 선언 (다른 레이아웃을 현재 레이아웃에 적용하기 위해 사용)

private LayoutInflater factory;

// LayoutInflater 변수 선언 (다른 레이아웃을 현재 레이아웃에 적용하기 위해 사용)

private View dialog; // 뷰 선언 (다른 레이아웃을 현재 레이아웃에 적용하기 위해 사용)

private AlertDialog save\_ad, update\_ad, saveok\_ad;

// 다이얼로그를 사용하기 위한 변수

private Drawable what\_alpha; // 투명도 설정

private Drawable[] time\_alpha = new Drawable[12]; // 교시의 투명도

private Drawable[] day\_alpha = new Drawable[5]; // 요일의 투명도

private Drawable[][] contents\_alpha = new Drawable[12][5]; // 내용의 투명도

@Override

protected void onCreate(Bundle savedInstanceState) {

// TODO Auto-generated method stub

super.onCreate(savedInstanceState);

setContentView(R.layout.timetable);

// DB 생성 시 예외처리 구문

try {

db\_create(); // assets/database폴더에 있는 db파일을 현재 AVD에 설치

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace(); // 에러시 프린트

}

init(); // 변수 초기화

setFont(); // 텍스트 폰트 지정

save();

// TimeTableSetting 클래스 (교시 선택 후 입력된 시간)로부터 값을 받아

static 변수에 입력

show(select\_num); // 화면에 보여질 줄 수 (교시)

setting(); // 초기 TimeTable 화면을 셋팅

setColor(); // 각 요일마다 해당되는 contents의 배경화면 지정

makealpha(); // 각 TextView 투명도 지정

// contents 부분의 TextView에 클릭 이벤트 지정

for(int i = 0; i < 12; i++)

for(int j = 0; j < 5; j++)

day[i][j].setOnClickListener(this);

}

private void db\_create() throws IOException { // DB 생성

File folder = new File("/data/data/abc.prj1/databases/");

folder.mkdirs();

File outfile = new File("/data/data/abc.prj1/databases/table.db");

AssetManager am = getResources().getAssets();

InputStream is = am.open("database/table.db", AssetManager.ACCESS\_BUFFER);

long filesize = is.available();

if(outfile.length() < filesize){

byte[] data = new byte[(int)filesize];

is.read(data);

is.close();

outfile.createNewFile();

FileOutputStream fos = new FileOutputStream(outfile);

fos.write(data);

fos.close();

}

}

// 옵션 메뉴 – 초기화를 위한 옵션

@Override

public boolean onCreateOptionsMenu(Menu menu) {

menu.add(0, 1, 0, "초기화");

return true;

}

// 옵션 메뉴 호출

@Override

public boolean onOptionsItemSelected(MenuItem item){

// 초기화 옵션 호출 시 DB의 정보를 모두 삭제한다.

if(item.getItemId() == 1){

sql\_db = db.getWritableDatabase();

sql\_db.execSQL("delete from TimeTable");

db.close();

Intent intent = getIntent();

finish();

startActivity(intent);

}

return true;

}

// TimeTableSetting 클래스 (교시 선택 후 입력된 시간)로부터 값을 받아 static 변수에 입력

private void save() {

// TODO Auto-generated method stub

if(get\_intent.getIntExtra("select\_num", 0) != 0)

select\_num = get\_intent.getIntExtra("select\_num", 0);

if(get\_intent.getStringExtra("time\_1") != null)

one = get\_intent.getStringExtra("time\_1").toString();

if(get\_intent.getStringExtra("time\_2") != null)

two = get\_intent.getStringExtra("time\_2").toString();

if(get\_intent.getStringExtra("time\_3") != null)

four = get\_intent.getStringExtra("time\_4").toString();

if(get\_intent.getStringExtra("time\_5") != null)

five = get\_intent.getStringExtra("time\_5").toString();

if(get\_intent.getStringExtra("time\_6") != null)

six = get\_intent.getStringExtra("time\_6").toString();

if(get\_intent.getStringExtra("time\_7") != null)

seven = get\_intent.getStringExtra("time\_7").toString();

if(get\_intent.getStringExtra("time\_8") != null)

eight = get\_intent.getStringExtra("time\_8").toString();

if(get\_intent.getStringExtra("time\_9") != null)

nine = get\_intent.getStringExtra("time\_9").toString();

if(get\_intent.getStringExtra("time\_10") != null)

ten = get\_intent.getStringExtra("time\_10").toString();

if(get\_intent.getStringExtra("time\_11") != null)

tenone = get\_intent.getStringExtra("time\_11").toString();

if(get\_intent.getStringExtra("time\_12") != null)

tentwo = get\_intent.getStringExtra("time\_12").toString();

}

// 초기 TimeTable 화면을 셋팅

// DB 상 저장된 시간표 정보를 불러와서 표시

private void setting() {

// 어플리케이션을 처음 시작했을 때

if(get\_intent.getStringExtra("notstart") == null) {

sql\_db = db.getReadableDatabase();

cursor = sql\_db.rawQuery("select \* from TimeTable", null);

while (cursor.moveToNext()) {

int time = new Filter().Filter\_time(cursor.getString(0));

int day = new Filter().Filter\_day(cursor.getString(2));

this.day[time][day].setText(cursor.getString(3) + "\n(" +

cursor.getString(4) + ")"); // 과목 내용 표시

}

cursor.close();

db.close();

}

else{ // 인텐트로 값을 받았을 경우

sql\_db = db.getReadableDatabase();

cursor = sql\_db.rawQuery("select \* from TimeTable", null);

while(cursor.moveToNext()){

int time = new Filter().Filter\_time(cursor.getString(0));

int day = new Filter().Filter\_day(cursor.getString(2));

this.day[time][day].setText(cursor.getString(3)+"\n("+cursor.getString(4)+")"); // 과목 내용 표시

}

cursor.close();

db.close();

}

}

// 변수 초기화 및 레이아웃 설정

private void init() {

// TODO Auto-generated method stub

get\_intent = getIntent(); // 받아오는 인텐트 설정

font = Typeface.createFromAsset(getAssets(), "menufont.ttf");

db = new Database(this);

save\_message\_box = new AlertDialog.Builder(this);

update\_message\_box = new AlertDialog.Builder(this);

saveok\_message\_box = new AlertDialog.Builder(this);

ctx = getApplicationContext();

factory

= (LayoutInflater)ctx.getSystemService(LAYOUT\_INFLATER\_SERVICE);

what = (TextView)findViewById(R.id.what);

row\_1 = (TableRow)findViewById(R.id.row\_1);

row\_2 = (TableRow)findViewById(R.id.row\_2);

row\_3 = (TableRow)findViewById(R.id.row\_3);

row\_4 = (TableRow)findViewById(R.id.row\_4);

row\_5 = (TableRow)findViewById(R.id.row\_5);

row\_6 = (TableRow)findViewById(R.id.row\_6);

row\_7 = (TableRow)findViewById(R.id.row\_7);

row\_8 = (TableRow)findViewById(R.id.row\_8);

row\_9 = (TableRow)findViewById(R.id.row\_9);

row\_10 = (TableRow)findViewById(R.id.row\_10);

row\_11 = (TableRow)findViewById(R.id.row\_11);

row\_12 = (TableRow)findViewById(R.id.row\_12);

mon = (TextView)findViewById(R.id.mon);

tue = (TextView)findViewById(R.id.tue);

wed = (TextView)findViewById(R.id.wed);

thu = (TextView)findViewById(R.id.thu);

fri = (TextView)findViewById(R.id.fri);

time\_1 = (TextView)findViewById(R.id.time\_1);

time\_2 = (TextView)findViewById(R.id.time\_2);

time\_3 = (TextView)findViewById(R.id.time\_3);

time\_4 = (TextView)findViewById(R.id.time\_4);

time\_5 = (TextView)findViewById(R.id.time\_5);

time\_6 = (TextView)findViewById(R.id.time\_6);

time\_7 = (TextView)findViewById(R.id.time\_7);

time\_8 = (TextView)findViewById(R.id.time\_8);

time\_9 = (TextView)findViewById(R.id.time\_9);

time\_10 = (TextView)findViewById(R.id.time\_10);

time\_11 = (TextView)findViewById(R.id.time\_11);

time\_12 = (TextView)findViewById(R.id.time\_12);

day[0][0] = (TextView)findViewById(R.id.mon\_1);

day[0][1] = (TextView)findViewById(R.id.tue\_1);

day[0][2] = (TextView)findViewById(R.id.wed\_1);

day[0][3] = (TextView)findViewById(R.id.thu\_1);

day[0][4] = (TextView)findViewById(R.id.fri\_1);

day[1][0] = (TextView)findViewById(R.id.mon\_2);

day[1][1] = (TextView)findViewById(R.id.tue\_2);

day[1][2] = (TextView)findViewById(R.id.wed\_2);

day[1][3] = (TextView)findViewById(R.id.thu\_2);

day[1][4] = (TextView)findViewById(R.id.fri\_2);

day[2][0] = (TextView)findViewById(R.id.mon\_3);

day[2][1] = (TextView)findViewById(R.id.tue\_3);

day[2][2] = (TextView)findViewById(R.id.wed\_3);

day[2][3] = (TextView)findViewById(R.id.thu\_3);

day[2][4] = (TextView)findViewById(R.id.fri\_3);

day[3][0] = (TextView)findViewById(R.id.mon\_4);

day[3][1] = (TextView)findViewById(R.id.tue\_4);

day[3][2] = (TextView)findViewById(R.id.wed\_4);

day[3][3] = (TextView)findViewById(R.id.thu\_4);

day[3][4] = (TextView)findViewById(R.id.fri\_4);

day[4][0] = (TextView)findViewById(R.id.mon\_5);

day[4][1] = (TextView)findViewById(R.id.tue\_5);

day[4][2] = (TextView)findViewById(R.id.wed\_5);

day[4][3] = (TextView)findViewById(R.id.thu\_5);

day[4][4] = (TextView)findViewById(R.id.fri\_5);

day[5][0] = (TextView)findViewById(R.id.mon\_6);

day[5][1] = (TextView)findViewById(R.id.tue\_6);

day[5][2] = (TextView)findViewById(R.id.wed\_6);

day[5][3] = (TextView)findViewById(R.id.thu\_6);

day[5][4] = (TextView)findViewById(R.id.fri\_6);

day[6][0] = (TextView)findViewById(R.id.mon\_7);

day[6][1] = (TextView)findViewById(R.id.tue\_7);

day[6][2] = (TextView)findViewById(R.id.wed\_7);

day[6][3] = (TextView)findViewById(R.id.thu\_7);

day[6][4] = (TextView)findViewById(R.id.fri\_7);

day[7][0] = (TextView)findViewById(R.id.mon\_8);

day[7][1] = (TextView)findViewById(R.id.tue\_8);

day[7][2] = (TextView)findViewById(R.id.wed\_8);

day[7][3] = (TextView)findViewById(R.id.thu\_8);

day[7][4] = (TextView)findViewById(R.id.fri\_8);

day[8][0] = (TextView)findViewById(R.id.mon\_9);

day[8][1] = (TextView)findViewById(R.id.tue\_9);

day[8][2] = (TextView)findViewById(R.id.wed\_9);

day[8][3] = (TextView)findViewById(R.id.thu\_9);

day[8][4] = (TextView)findViewById(R.id.fri\_9);

day[9][0] = (TextView)findViewById(R.id.mon\_10);

day[9][1] = (TextView)findViewById(R.id.tue\_10);

day[9][2] = (TextView)findViewById(R.id.wed\_10);

day[9][3] = (TextView)findViewById(R.id.thu\_10);

day[9][4] = (TextView)findViewById(R.id.fri\_10);

day[10][0] = (TextView)findViewById(R.id.mon\_11);

day[10][1] = (TextView)findViewById(R.id.tue\_11);

day[10][2] = (TextView)findViewById(R.id.wed\_11);

day[10][3] = (TextView)findViewById(R.id.thu\_11);

day[10][4] = (TextView)findViewById(R.id.fri\_11);

day[11][0] = (TextView)findViewById(R.id.mon\_12);

day[11][1] = (TextView)findViewById(R.id.tue\_12);

day[11][2] = (TextView)findViewById(R.id.wed\_12);

day[11][3] = (TextView)findViewById(R.id.thu\_12);

day[11][4] = (TextView)findViewById(R.id.fri\_12);

// 투명도 선언 //

what\_alpha = what.getBackground();

time\_alpha[0] = time\_1.getBackground();

time\_alpha[1] = time\_2.getBackground();

time\_alpha[2] = time\_3.getBackground();

time\_alpha[3] = time\_4.getBackground();

time\_alpha[4] = time\_5.getBackground();

time\_alpha[5] = time\_6.getBackground();

time\_alpha[6] = time\_7.getBackground();

time\_alpha[7] = time\_8.getBackground();

time\_alpha[8] = time\_9.getBackground();

time\_alpha[9] = time\_10.getBackground();

time\_alpha[10] = time\_11.getBackground();

time\_alpha[11] = time\_12.getBackground();

day\_alpha[0] = mon.getBackground();

day\_alpha[1] = tue.getBackground();

day\_alpha[2] = wed.getBackground();

day\_alpha[3] = thu.getBackground();

day\_alpha[4] = fri.getBackground();

for(int i = 0; i < 12; i++)

for(int j = 0; j < 5; j++)

contents\_alpha[i][j] = day[i][j].getBackground();

}

// 텍스트 폰트 지정

private void setFont() {

// TODO Auto-generated method stub

mon.setTypeface(font);

tue.setTypeface(font);

wed.setTypeface(font);

thu.setTypeface(font);

fri.setTypeface(font);

time\_1.setTypeface(font);

time\_2.setTypeface(font);

time\_3.setTypeface(font);

time\_4.setTypeface(font);

time\_5.setTypeface(font);

time\_6.setTypeface(font);

time\_7.setTypeface(font);

time\_8.setTypeface(font);

time\_9.setTypeface(font);

time\_10.setTypeface(font);

time\_11.setTypeface(font);

time\_12.setTypeface(font);

for(int i = 0; i < 12; i++)

for(int j = 0; j < 5; j++)

day[i][j].setTypeface(font);

}

// 화면에 보여질 줄 수 (교시)

private void show(int a) {

// TODO Auto-generated method stub

while(a > 0){

if(a == 1){

row\_1.setVisibility(0);

time\_1.setText(one);

}

else if(a == 2){

row\_2.setVisibility(0);

time\_2.setText(two);

}

else if(a == 3){

row\_3.setVisibility(0);

time\_3.setText(three);

}

else if(a == 4){

row\_4.setVisibility(0);

time\_4.setText(four);

}

else if(a == 5){

row\_5.setVisibility(0);

time\_5.setText(five);

}

else if(a == 6){

row\_6.setVisibility(0);

time\_6.setText(six);

}

else if(a == 7){

row\_7.setVisibility(0);

time\_7.setText(seven);

}

else if(a == 8){

row\_8.setVisibility(0);

time\_8.setText(eight);

}

else if(a == 9){

row\_9.setVisibility(0);

time\_9.setText(nine);

}

else if(a == 10){

row\_10.setVisibility(0);

time\_10.setText(ten);

}

else if(a == 11){

row\_11.setVisibility(0);

time\_11.setText(tenone);

}

else if(a == 12){

row\_12.setVisibility(0);

time\_12.setText(tentwo);

}

a--;

}

}

// 해당되는 contents의 색상정보 설정

private void setColor() {

// TODO Auto-generated method stub

sql\_db = db.getReadableDatabase();

cursor = sql\_db.rawQuery("select \* from TimeTable", null);

while(cursor.moveToNext()){

int time = new Filter().Filter\_time(cursor.getString(0));

int day = new Filter().Filter\_day(cursor.getString(2));

String color ="";

color = cursor.getString(5).toString();

if(color.equals("LightPink"))

this.day[time][day].setBackgroundColor(Color.argb(100, 255, 192, 203));

else if(color.equals("LightGold"))

this.day[time][day].setBackgroundColor(Color.argb(100, 250, 250, 210));

else if(color.equals("Peachpuff"))

this.day[time][day].setBackgroundColor(Color.argb(100, 255, 218, 185));

else if(color.equals("Lavender"))

this.day[time][day].setBackgroundColor(Color.argb(100, 230, 230, 250));

else if(color.equals("Palegreen"))

this.day[time][day].setBackgroundColor(Color.argb(100, 152, 251, 152));

else if(color.equals("Violet"))

this.day[time][day].setBackgroundColor(Color.argb(100, 238, 130, 238));

else if(color.equals("SpringGreen"))

this.day[time][day].setBackgroundColor(Color.argb(100, 0, 255, 127));

else if(color.equals("LightCyan"))

this.day[time][day].setBackgroundColor(Color.argb(100, 224, 255, 255));

}

cursor.close();

db.close();

}

// 각 TextView 투명도 지정

private void makealpha(){

// TODO Auto-generated method stub

what\_alpha.setAlpha(100);

for(int i = 0; i < 12; i++)

time\_alpha[i].setAlpha(100);

for(int i = 0; i < 5; i++)

day\_alpha[i].setAlpha(100);

for(int i = 0; i < 12; i++)

for(int j = 0; j < 5; j++)

contents\_alpha[i][j].setAlpha(100);

}

// 이벤트 영역 - 각 해당 교시를 선택했을 때 발생하는 이벤트

public void onClick(View v) {

// TODO Auto-generated method stub

finish();

for(int i = 0; i < 12; i++)

for(int j = 0; j < 5; j++){

if(v == day[i][j]){

intent = new Intent(TimeTable.this, TableSetting.class);

// 시간표 설정 화면으로 전환

if(i == 0){

intent.putExtra("time\_", "1교시");

intent.putExtra("time", time\_1.getText().toString());

}

else if(i == 1){

intent.putExtra("time\_", "2교시");

intent.putExtra("time", time\_2.getText().toString());

}

else if(i == 2){

intent.putExtra("time\_", "3교시");

intent.putExtra("time", time\_3.getText().toString());

}

else if(i == 3){

intent.putExtra("time\_", "4교시");

intent.putExtra("time", time\_4.getText().toString());

}

else if(i == 4){

intent.putExtra("time\_", "5교시");

intent.putExtra("time", time\_5.getText().toString());

}

else if(i == 5){

intent.putExtra("time\_", "6교시");

intent.putExtra("time", time\_6.getText().toString());

}

else if(i == 6){

intent.putExtra("time\_", "7교시");

intent.putExtra("time", time\_7.getText().toString());

}

else if(i == 7){

intent.putExtra("time\_", "8교시");

intent.putExtra("time", time\_8.getText().toString());

}

else if(i == 8){

intent.putExtra("time\_", "9교시");

intent.putExtra("time", time\_9.getText().toString());

}

else if(i == 9){

intent.putExtra("time\_", "10교시");

intent.putExtra("time", time\_10.getText().toString());

}

else if(i == 10){

intent.putExtra("time\_", "11교시");

intent.putExtra("time", time\_11.getText().toString());

}

else if(i == 11){

intent.putExtra("time\_", "12교시");

intent.putExtra("time", time\_12.getText().toString());

}

if(j == 0)

intent.putExtra("day", "월요일");

else if(j == 1)

intent.putExtra("day", "화요일");

else if(j == 2)

intent.putExtra("day", "수요일");

else if(j == 3)

intent.putExtra("day", "목요일");

else if(j == 4)

intent.putExtra("day", "금요일");

if(day[i][j].getText() != null) {

sql\_db = db.getReadableDatabase();

cursor = sql\_db.rawQuery("select \* from TimeTable", null);

while(cursor.moveToNext()){

intent.putExtra("contents",cursor.getString(3));

intent.putExtra("rooms",cursor.getString(4));

intent.putExtra("contents","");

intent.putExtra("rooms","");

}

cursor.close();

db.close();

}

startActivity(intent);

}

}

}

}

TableSetting.java

package com.example.it103\_32.cnt;

import android.app.\*;

import android.content.\*;

import android.database.\*;

import android.database.sqlite.\*;

import android.graphics.\*;

import android.graphics.drawable.\*;

import android.os.\*;

import android.view.\*;

import android.view.View.OnClickListener;

import android.widget.\*;

public class TableSetting extends Activity implements OnClickListener{

private Intent intent, get\_intent;

private Typeface font; // 폰트

private Button table\_save, table\_delete, color\_contents, extend\_contents;

private TextView what\_day, what\_time, what\_contents,what\_room, color, extend;

private EditText what\_day\_text, what\_time\_text, what\_contents\_text, what\_room\_text;

private String what\_time\_text\_=""; // 선택한 교시를 넣을 변수

private String what\_day\_text\_=""; // 선택한 요일를 넣을 변수

private String what\_color\_text\_="";

private Database db; // Database 클래스 선언

private SQLiteDatabase sql\_db; // DB 정보

private Cursor cursor; // select 쿼리문 사용위한 선언

private Drawable back\_btn\_alpha, save\_btn\_alpha; // 투명도 선언

@Override

protected void onCreate(Bundle savedInstanceState) {

// TODO Auto-generated method stub

super.onCreate(savedInstanceState);

setContentView(R.layout.tablesetting);

init(); // 변수 초기화

setFont(); // 폰트 설정

makealpha(); // 투명도 조정

getintent(); // 받은 인텐트값 설정

// 이벤트 선언

table\_save.setOnClickListener(this);

table\_delete.setOnClickListener(this);

extend\_contents.setOnClickListener(this);

color\_contents.setOnClickListener(this);

}

// 변수 초기화

private void init() {

// TODO Auto-generated method stub

get\_intent = getIntent();

intent = new Intent(TableSetting.this, TimeTable.class);

font = Typeface.createFromAsset(getAssets(), "menufont.ttf");

db = new Database(this);

table\_save = (Button)findViewById(R.id.table\_save);

table\_delete = (Button)findViewById(R.id.table\_delete);

color\_contents = (Button)findViewById(R.id.color\_contents);

extend\_contents = (Button)findViewById(R.id.extend\_contents);

what\_day = (TextView)findViewById(R.id.what\_day);

what\_time = (TextView)findViewById(R.id.what\_time);

what\_contents = (TextView)findViewById(R.id.what\_contents);

what\_room = (TextView)findViewById(R.id.what\_room);

color = (TextView)findViewById(R.id.color);

extend = (TextView)findViewById(R.id.extend);

what\_day\_text = (EditText)findViewById(R.id.what\_day\_text);

what\_time\_text = (EditText)findViewById(R.id.what\_time\_text);

what\_contents\_text = (EditText)findViewById(R.id.what\_contents\_text);

what\_room\_text = (EditText)findViewById(R.id.what\_room\_text);

save\_btn\_alpha = table\_save.getBackground();

}

// 폰트 설정

private void setFont() {

// TODO Auto-generated method stub

what\_day.setTypeface(font);

what\_time.setTypeface(font);

what\_contents.setTypeface(font);

color\_contents.setTypeface(font);

extend\_contents.setTypeface(font);

what\_day\_text.setTypeface(font);

what\_time\_text.setTypeface(font);

what\_contents\_text.setTypeface(font);

what\_room\_text.setTypeface(font);

what\_room.setTypeface(font);

table\_save.setTypeface(font);

table\_delete.setTypeface(font);

color.setTypeface(font);

extend.setTypeface(font);

}

// 투명도 조정

private void makealpha(){

// TODO Auto-generated method stub

// back\_btn\_alpha.setAlpha(200);

save\_btn\_alpha.setAlpha(200);

}

// 받은 인텐트값 설정

// TimeTable에서 선택한 요일, 교시 인텐트 값을 받아온다.

private void getintent(){

what\_day\_text.setText(get\_intent.getStringExtra("day"));

what\_time\_text.setText(get\_intent.getStringExtra("time"));

what\_time\_text\_ = get\_intent.getStringExtra("time\_").toString();

what\_day\_text\_ = get\_intent.getStringExtra("day").toString();

// 선택한 교시와 요일에 저장된 정보를 받아온다.

sql\_db = db.getReadableDatabase();

cursor = sql\_db.rawQuery("select \* from TimeTable where time\_='"+what\_time\_text\_+"' and day='"+what\_day\_text\_+"';", null);

if(cursor.moveToNext()) {

what\_contents\_text.setText(cursor.getString(3).toString());

what\_room\_text.setText(cursor.getString(4).toString());

color\_contents.setText(cursor.getString(5).toString());

}

cursor.close();

db.close();

}

//연강 버튼 다이얼로그 호출

private void Select\_Time(){

final CharSequence[] Extendtime = {"1", "2", "3"};

AlertDialog.Builder alt\_bld = new AlertDialog.Builder(this);

alt\_bld.setIcon(R.drawable.icon);

alt\_bld.setTitle("연강 시간 선택");

alt\_bld.setSingleChoiceItems(Extendtime, -1, new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int item) {

dialog.dismiss();

extend\_contents.setText(Extendtime[item]);

}

});

AlertDialog alert = alt\_bld.create();

alert.show();

}

private void Select\_Color()

{

final CharSequence[] Color = {"LightPink", "LightGold", "Peachpuff","Lavender","Palegreen","Violet","SpringGreen","LightCyan"};

AlertDialog.Builder alt\_bld = new AlertDialog.Builder(this);

alt\_bld.setIcon(R.drawable.icon);

alt\_bld.setTitle("연강 시간 선택");

alt\_bld.setSingleChoiceItems(Color, -1, new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int item) {

dialog.dismiss();

color\_contents.setText(Color[item]);

}

});

AlertDialog alert = alt\_bld.create();

alert.show();

}

// 뒤로가기 기능 지원

public void onBackPressed() {

super.onBackPressed();

finish();

intent = new Intent(TableSetting.this, TimeTable.class);

intent.putExtra("notstart", "ok");

startActivity(intent);

}

// 이벤트 설정

public void onClick(View v) {

// TODO Auto-generated method stub

if(v == table\_delete) // 해당 정보 삭제

{

finish();

sql\_db = db.getWritableDatabase();

sql\_db.execSQL("delete from TimeTable where time\_='"+what\_time\_text\_+"' and day='"+what\_day\_text\_+"';");

db.close();

intent.putExtra("notstart", "ok");

startActivity(intent);

}

else if(v == extend\_contents) // 연강 버튼 기능 호출

{

Select\_Time();

}

else if(v==color\_contents) // 색상 버튼 기능 호출

{

Select\_Color();

}

else // 저장 버튼 기능 호출

// 입력한 정보를 DB에 저장 후 TimeTable 클래스로 인텐트 형태로 넘김

{

finish();

if(extend\_contents.getText().toString().equals("연강"))

{

extend\_contents.setText("1");

}

if(color\_contents.getText().toString().equals("색상"))

{

color\_contents.setText("LightPink");

}

int count=Integer.parseInt(extend\_contents.getText().toString());

for(int i=0; i<count; ++i)

{

int temp\_time = Integer.parseInt(what\_time\_text.getText().toString())+i;

what\_time\_text\_= Integer.toString(temp\_time) + "교시";

what\_color\_text\_= color\_contents.getText().toString();

sql\_db = db.getReadableDatabase();

cursor = sql\_db.rawQuery("select contents and room and color from TimeTable where time\_='"+what\_time\_text\_+"' and day='"+what\_day\_text\_+"';", null);

sql\_db = db.getWritableDatabase();

sql\_db = db.getWritableDatabase();

if(cursor.moveToNext()) {

sql\_db.execSQL("UPDATE TimeTable SET contents = '" + what\_contents\_text.getText().toString() +"' WHERE time\_ = '" + what\_time\_text\_ + "' and day = '" + what\_day\_text\_ + "';");

sql\_db.execSQL("UPDATE TimeTable SET room = '" + what\_room\_text.getText().toString() +"' WHERE time\_ = '" + what\_time\_text\_ + "' and day = '" + what\_day\_text\_ + "';");

sql\_db.execSQL("UPDATE TimeTable SET color = '" + what\_color\_text\_ +"' WHERE time\_ = '" + what\_time\_text\_ + "' and day = '" + what\_day\_text\_ + "';");

}

else // DB 추가하는 기능

if(!what\_contents\_text.getText().toString().equals("") && !what\_room\_text.getText().toString().equals(""))

sql\_db.execSQL("insert into TimeTable values ('"+what\_time\_text\_+"', '"+what\_time\_text.getText().toString()+"', '"+what\_day\_text.getText().toString()+"', '"+what\_contents\_text.getText().toString()+"', '"+what\_room\_text.getText().toString()+"', '"+what\_color\_text\_+"');");

cursor.close();

db.close();

}

intent.putExtra("notstart", "ok");

startActivity(intent);

}

}

}

FourFloor.java

package com.example.it103\_32.cnt.LeeFloor;

import android.content.Context;

import android.content.Intent;

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

import android.graphics.Canvas;

import android.graphics.Color;

import android.graphics.Paint;

import android.os.Bundle;

import android.os.Handler;

import android.os.Message;

import android.os.RemoteException;

import android.support.v7.app.AppCompatActivity;

import android.view.MotionEvent;

import android.view.View;

import android.widget.TextView;

import com.example.it103\_32.cnt.R;

import org.altbeacon.beacon.Beacon;

import org.altbeacon.beacon.BeaconConsumer;

import org.altbeacon.beacon.BeaconParser;

import org.altbeacon.beacon.Identifier;

import org.altbeacon.beacon.RangeNotifier;

import java.util.ArrayList;

import java.util.Collection;

import java.util.List;

/\*\*

\* Created by USER on 2016-11-22.

\*/

public class FourFloor extends AppCompatActivity implements BeaconConsumer {

private org.altbeacon.beacon.BeaconManager beaconManager;

float[] pointx={135f, 414f, 630f, 707f, 1000f, 1000f} ;

float[] pointy={1725f, 1725f, 1725f, 977f, 840f, 120f} ;

int Rssi ;

Double Distance;

Identifier BeaconMionr ;

String BM;

private Intent intent;

private int x;

private int y;

private MyView view;

// 감지된 비콘들을 임시로 담을 리스트

private List<org.altbeacon.beacon.Beacon> beaconList = new ArrayList<>();

TextView textView;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

view = new MyView(this);

setContentView(view);

// 실제로 비콘을 탐지하기 위한 비콘매니저 객체를 초기화

beaconManager = org.altbeacon.beacon.BeaconManager.getInstanceForApplication(this);

textView = (TextView)findViewById(R.id.tvId);

// 여기가 중요한데, 기기에 따라서 setBeaconLayout 안의 내용을 바꿔줘야

하는듯 싶다.

// 필자의 경우에는 아래처럼 하니 잘 동작했음.

beaconManager.getBeaconParsers().add(new BeaconParser().setBeaconLayout("m:2-3=0215,i:4-19,i:20-21,i:22-23,p:24-24,d:25-25"));

// 비콘 탐지를 시작한다. 실제로는 서비스를 시작하는것.

beaconManager.bind(this);

}

public boolean onTouchEvent(MotionEvent event){

x = (int)event.getX(); // 화면상의 x좌표(가로)

y = (int)event.getY(); // 화면상의 y좌표(세로)

if(event.getAction() == MotionEvent.ACTION\_DOWN && x <150 && y <150) // 터치가 발생했을때 x좌표가 150미만, y좌표가 150미만인 경우 공지사항을 intent한다.

{

intent = new Intent(FourFloor.this, LeeNotify.class);

startActivity(intent);

}

return true;

}

public class MyView extends View {

public MyView(Context context) {

super(context);

}

public void onDraw(Canvas canvas) { // 이공관 4층 지도 실행 부분

Paint pnt = new Paint();

Bitmap leemap = BitmapFactory.decodeResource(getResources(), R.drawable.leefour);

canvas.drawBitmap(leemap, 0, 0, null);

pnt.setColor(Color.RED); // 원의 색을 빨간색으로 설정

for (int j = 0; j < beaconList.size(); ++j) {

Rssi = beaconList.get(j).getRssi(); // 비콘의 신호세기

Distance = beaconList.get(j).getDistance(); // 비콘 거리

BeaconMionr = beaconList.get(j).getId3(); // 비콘의 마이너

BM = BeaconMionr.toString(); // 마이너값을 switch문을 사용하기위해 String형식으로 변환

if (Rssi > -80 && Distance < 2.5) { // Rssi값이 -80보다 크고, Distance값이 2.5보다 작을 경우 아래의 switch문 실행

switch (BM) {

case "400": { // 비콘의 마이너값 400

canvas.drawCircle(pointx[3], pointy[3], 15, pnt);

}

break;

case "401": { // 비콘의 마이너값 401

canvas.drawCircle(pointx[0], pointy[0], 15, pnt);

}

break;

case "402": { // 비콘의 마이너값 402

canvas.drawCircle(pointx[1], pointy[1], 15, pnt);

}

break;

case "403": { // 비콘의 마이너값 403

canvas.drawCircle(pointx[2], pointy[2], 15, pnt);

}

break;

case "404": { // 비콘의 마이너값 404

canvas.drawCircle(pointx[4], pointy[4], 15, pnt);

}

break;

case "405": { // 비콘의 마이너값 405

canvas.drawCircle(pointx[5], pointy[5], 15, pnt);

}

break;

}

break;

}

}

}

}

@Override

protected void onDestroy() {

super.onDestroy();

beaconManager.unbind(this);

}

@Override

public void onBeaconServiceConnect() {

beaconManager.setRangeNotifier(new RangeNotifier() {

@Override

// 비콘이 감지되면 해당 함수가 호출된다. Collection<Beacon> beacons에는 감지된 비콘의 리스트가,

// region에는 비콘들에 대응하는 Region 객체가 들어온다.

public void didRangeBeaconsInRegion(Collection<Beacon> beacons, org.altbeacon.beacon.Region region) {

if (beacons.size() > 0) {

beaconList.clear();

for (org.altbeacon.beacon.Beacon beacon : beacons) {

beaconList.add(beacon);

}

handler.sendEmptyMessage(0);

}

}

});

try {

beaconManager.startRangingBeaconsInRegion(new org.altbeacon.beacon.Region("myRangingUniqueId", null, null, null));

} catch (RemoteException e) { }

}

Handler handler = new Handler() {

public void handleMessage(Message msg) {

// 자기 자신을 1초마다 호출

handler.sendEmptyMessageDelayed(0, 1000);

view.invalidate();

}

};

}