Java Files

# AndroidManifest.xml

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

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

package="com.example.jackson.diabetesapp">

<uses-permission android:name="android.permission.VIBRATE" />

<uses-permission android:name="com.android.alarm.permission.SET\_ALARM" />

<uses-permission android:name="android.permission.WAKE\_LOCK" />

<uses-permission android:name="android.permission.RECEIVE\_BOOT\_COMPLETED" />

<application

android:allowBackup="true"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:theme="@style/AppTheme">

<receiver android:name=".AlarmBroadcastReceiver">

<intent-filter>

<action android:name="android.intent.action.BOOT\_COMPLETED" />

</intent-filter>

</receiver>

<receiver android:name=".AlarmReceiver"></receiver>

<activity

android:name=".MainActivity"

android:label="@string/app\_name"

android:screenOrientation="portrait">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<activity

android:name=".NewEntryActivity"

android:label="Create New Entry:"

android:parentActivityName=".MainActivity"

android:windowSoftInputMode="stateVisible"

android:screenOrientation="portrait">

<meta-data

android:name="android.support.PARENT\_ACTIVITY"

android:value="com.example.jackson.diabetesapp.MainActivity" />

</activity>

<activity

android:name=".RemindersActivity"

android:label="Reminders"

android:parentActivityName=".MainActivity"

android:screenOrientation="portrait">

<meta-data

android:name="android.support.PARENT\_ACTIVITY"

android:value="com.example.jackson.diabetesapp.MainActivity" />

</activity>

<activity

android:name=".AlarmActivity"

android:label=""></activity>

<service android:name=".RingtonePlayingService"></service>

<receiver

android:name=".AutoStartup"

android:enabled="true"

android:exported="true"

android:permission="android.permission.RECEIVE\_BOOT\_COMPLETED"></receiver>

<service

android:name=".AlarmService"

android:enabled="true"

android:exported="true"></service>

<activity

android:name=".TrackingActivity"

android:label="Tracking"

android:parentActivityName=".MainActivity">

<meta-data

android:name="android.support.PARENT\_ACTIVITY"

android:value="com.example.jackson.diabetesapp.MainActivity" />

</activity>

<activity

android:name=".SettingsActivity"

android:label="@string/title\_activity\_settings"

android:screenOrientation="portrait"></activity>

<activity

android:name=".DeleteData"

android:label="@string/title\_activity\_delete\_data"

android:parentActivityName=".MainActivity">

<intent-filter>

<action android:name=".activities.DeleteData" />

<category android:name="android.intent.category.DEFAULT" />

</intent-filter>

<meta-data

android:name="android.support.PARENT\_ACTIVITY"

android:value="com.example.jackson.diabetesapp.MainActivity" />

</activity>

</application>

</manifest>

# AlarmActivity.class

package com.example.jackson.diabetesapp;

import android.content.Context;

import android.content.Intent;

import android.os.Bundle;

import android.support.v7.app.ActionBarActivity;

import android.util.Log;

import android.view.View;

import android.view.Window;

import android.view.WindowManager;

import android.widget.Button;

import android.widget.TextView;

public class AlarmActivity extends ActionBarActivity {

Context context = this;

String cancel = "1";

@Override

protected void onCreate(Bundle savedInstanceState) {

this.requestWindowFeature(Window.FEATURE\_NO\_TITLE);

super.onCreate(savedInstanceState);

getWindow().addFlags(WindowManager.LayoutParams.FLAG\_TURN\_SCREEN\_ON);

getWindow().addFlags(WindowManager.LayoutParams.FLAG\_KEEP\_SCREEN\_ON);

getWindow().addFlags(WindowManager.LayoutParams.FLAG\_SHOW\_WHEN\_LOCKED);

getWindow().addFlags(WindowManager.LayoutParams.FLAG\_DISMISS\_KEYGUARD);

setContentView(R.layout.activity\_alarm);

Intent startIntent = getIntent();

cancel = "1";

String alarmName = ("Alarm Name");

String alarmMsg = ("Alarm Message");

if (startIntent.getStringExtra("alarmName") != null) {

alarmName = startIntent.getStringExtra("alarmName");

}

if (startIntent.getStringExtra("alarmMsg") != null) {

alarmMsg = startIntent.getStringExtra("alarmMsg");

}

if (startIntent.getStringExtra("alarmMsg") != null && startIntent.getStringExtra("alarmName") != null) {

Log.i("stringsFromIntent", "The intent returned the following strings: " + "\n" + alarmName + "\n" + alarmMsg);

} else {

Log.i("stringsFromIntent", "one of the values was null");

}

final Button shutup\_alarm = (Button) findViewById(R.id.shutup\_alarm);

final TextView tvName = (TextView) findViewById(R.id.textViewAlarmName);

final TextView tvMsg = (TextView) findViewById(R.id.textViewAlarmMsg);

tvName.setText(alarmName);

tvMsg.setText(alarmMsg);

shutup\_alarm.setOnClickListener(new View.OnClickListener() {

public void onClick(View v) {

Intent intent = new Intent(context, AlarmReceiver.class);

intent.putExtra("cancel", cancel);

sendBroadcast(intent);

}

});

}

}

# AlarmBroadCastReceiver.class

package com.example.jackson.diabetesapp;

import android.content.BroadcastReceiver;

import android.content.Context;

import android.content.Intent;

/\*\*

\* Created by Jackson on 4/13/2015.

\*/

public class AlarmBroadcastReceiver extends BroadcastReceiver {

@Override

public void onReceive(Context context, Intent intent) {

Intent startServiceIntent = new Intent(context, MainActivity.class);

context.startService(startServiceIntent);

}

}

# AlarmReceiver.class

package com.example.jackson.diabetesapp;

import android.annotation.TargetApi;

import android.app.Notification;

import android.app.NotificationManager;

import android.app.PendingIntent;

import android.app.TaskStackBuilder;

import android.content.BroadcastReceiver;

import android.content.Context;

import android.content.Intent;

import android.os.Build;

import android.os.PowerManager;

import android.os.Vibrator;

import android.support.v4.app.NotificationCompat;

import android.util.Log;

public class AlarmReceiver extends BroadcastReceiver {

public static Vibrator myVib;

@TargetApi(Build.VERSION\_CODES.JELLY\_BEAN)

public void onReceive(Context context, Intent intent) {

PowerManager pm = (PowerManager) context.getSystemService(Context.POWER\_SERVICE);

PowerManager.WakeLock wl = pm.newWakeLock(PowerManager.PARTIAL\_WAKE\_LOCK, "");

wl.acquire();

String alarmName = intent.getStringExtra("alarmName");

String alarmMsg = intent.getStringExtra("alarmMsg");

String cancel = intent.getStringExtra("cancel");

Log.i("alarmCancel", "" + cancel);

if (cancel != null) {

Log.i("alarmCancel", "Inside if statement: " + cancel);

Intent resultIntent = new Intent(context, MainActivity.class);

resultIntent.setFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK);

context.startActivity(resultIntent);

NotificationManager mNotificationManager =

(NotificationManager) context.getSystemService(Context.NOTIFICATION\_SERVICE);

mNotificationManager.cancel(119);

mNotificationManager.cancelAll();

/\*

Uri notification = RingtoneManager.getDefaultUri(RingtoneManager.TYPE\_RINGTONE);

Ringtone r = RingtoneManager.getRingtone(context, notification);

r.stop();

\*/

Intent stopIntent = new Intent(context, RingtonePlayingService.class);

context.stopService(stopIntent);

} else {

Log.i("alarmCancel", "Inside else statement: " + cancel);

myVib = (Vibrator) context.getSystemService(Context.VIBRATOR\_SERVICE);

myVib.vibrate(1000);

/\*

\*/

Intent startRIntent = new Intent(context, RingtonePlayingService.class);

context.startService(startRIntent);

NotificationCompat.Builder mBuilder =

new NotificationCompat.Builder(context)

.setSmallIcon(R.mipmap.ic\_launcher)

.setContentTitle(alarmName)

.setContentText(alarmMsg);

Intent resultIntent = new Intent(context, AlarmActivity.class);

resultIntent.putExtra("alarmName", alarmName);

resultIntent.putExtra("alarmMsg", alarmMsg);

resultIntent.setFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK);

context.startActivity(resultIntent);

TaskStackBuilder stackBuilder = TaskStackBuilder.create(context);

stackBuilder.addParentStack(AlarmActivity.class);

stackBuilder.addNextIntent(resultIntent);

PendingIntent resultPendingIntent =

stackBuilder.getPendingIntent(

0,

PendingIntent.FLAG\_UPDATE\_CURRENT

);

// Uses the default lighting scheme

// Will show lights and make the notification disappear when the presses it

mBuilder.setContentIntent(resultPendingIntent);

NotificationManager mNotificationManager =

(NotificationManager) context.getSystemService(Context.NOTIFICATION\_SERVICE);

Notification note = mBuilder.build();

note.flags |= Notification.FLAG\_AUTO\_CANCEL | Notification.FLAG\_SHOW\_LIGHTS;

note.defaults |= Notification.DEFAULT\_LIGHTS;

mNotificationManager.notify("", 119, mBuilder.build());

}

wl.release();

}

}

# AlarmService.class

package com.example.jackson.diabetesapp;

import android.app.Service;

import android.content.Intent;

import android.os.Bundle;

import android.os.IBinder;

import android.util.Log;

public class AlarmService extends Service {

public AlarmService() {

}

public void onStartCommand() {

}

@Override

public IBinder onBind(Intent intent) {

throw new UnsupportedOperationException("Not yet implemented");

}

protected void onCreate(Bundle savedInstanceState) {

Log.i("alarmService", "Alarm Service Started");

int[] alarms = new int[7];

for (int i = 0; i < alarms.length; i++) {

alarms[i] = 0;

}

}

}

# AutoStartup.class

package com.example.jackson.diabetesapp;

import android.content.BroadcastReceiver;

import android.content.Context;

import android.content.Intent;

public class AutoStartup extends BroadcastReceiver {

public AutoStartup() {

}

@Override

public void onReceive(Context context, Intent intent) {

throw new UnsupportedOperationException();

}

}

# DataHandler.class

package com.example.jackson.diabetesapp;

import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

public class DataHandler {

public static final int TIME = 0;

public static final int BG = 0;

public static final int CARBS = 0;

public static final int PROTEIN = 0;

public static final int FAT = 0;

public static final String TABLE\_NAME = "dataTable";

public static final String DATABASE\_NAME = "database.sqlite";

public static final int DATABASE\_VERSION = 1;

public static final String TABLE\_CREATE = "create table dataTable (time integer not null,bg integer not null, carbs integer not null,protein integer not null,fat integer not null);";

DataBaseHelper dbhelper;

Context context;

SQLiteDatabase db;

public DataHandler(Context context) {

this.context = context;

dbhelper = new DataBaseHelper(context);

}

private static class DataBaseHelper extends SQLiteOpenHelper {

public DataBaseHelper(Context context) {

super(context, DATABASE\_NAME, null, DATABASE\_VERSION);

}

@Override

public void onCreate(SQLiteDatabase db) {

db.execSQL(TABLE\_CREATE);

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

db.execSQL("DROP TABLE IF EXISTS table");

onCreate(db);

}

}

public DataHandler open() {

db = dbhelper.getWritableDatabase();

return this;

}

public void close() {

dbhelper.close();

}

public void delete() {

db.execSQL("delete from " + TABLE\_NAME);

}

public long insertData(int time, int bg, int carbs, int protein, int fat) {

ContentValues content = new ContentValues();

content.put("TIME", time);

content.put("BG", bg);

content.put("CARBS", carbs);

content.put("PROTEIN", protein);

content.put("FAT", fat);

return db.insertOrThrow(TABLE\_NAME, null, content);

}

public Cursor returnData() {

return db.query(TABLE\_NAME, new String[]{"TIME", "BG", "CARBS", "PROTEIN", "FAT"}, null, null, null, null, null);

}

}

# DeleteData.class

package com.example.jackson.diabetesapp;

import android.app.AlertDialog;

import android.content.Context;

import android.content.DialogInterface;

import android.content.Intent;

import android.os.Bundle;

import android.support.v7.app.ActionBarActivity;

import android.view.Menu;

import android.view.MenuItem;

import android.view.Window;

import android.widget.Toast;

public class DeleteData extends ActionBarActivity {

DataHandler handler;

@Override

protected void onCreate(Bundle savedInstanceState) {

this.requestWindowFeature(Window.FEATURE\_NO\_TITLE);

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

Context context = this;

DialogInterface.OnClickListener dialogClickListener = new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

switch (which) {

case DialogInterface.BUTTON\_POSITIVE:

deleteData();

break;

case DialogInterface.BUTTON\_NEGATIVE:

Intent intent = new Intent(DeleteData.this, MainActivity.class);

startActivity(intent);

break;

}

}

};

AlertDialog.Builder builder = new AlertDialog.Builder(context);

builder.setMessage("Permanently delete all user data?").setPositiveButton("Yes", dialogClickListener)

.setNegativeButton("No", dialogClickListener).show();

}

public void deleteData() {

handler = new DataHandler(getBaseContext());

handler.open();

handler.delete();

handler.close();

Intent intent = new Intent(DeleteData.this, MainActivity.class);

startActivity(intent);

Toast

.makeText(getApplicationContext(), "User Data Deleted", Toast.LENGTH\_LONG)

.show();

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.menu\_delete\_data, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

// Handle action bar item clicks here. The action bar will

// automatically handle clicks on the Home/Up button, so long

// as you specify a parent activity in AndroidManifest.xml.

int id = item.getItemId();

//noinspection SimplifiableIfStatement

if (id == R.id.action\_settings) {

return true;

}

return super.onOptionsItemSelected(item);

}

}

# MainActivity.class

package com.example.jackson.diabetesapp;

import android.content.Intent;

import android.os.Bundle;

import android.os.Vibrator;

import android.support.v7.app.ActionBarActivity;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.widget.Button;

public class MainActivity extends ActionBarActivity {

public static Vibrator myVib;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

getSupportActionBar().setDisplayHomeAsUpEnabled(true);

getSupportActionBar().setHomeButtonEnabled(true);

myVib = (Vibrator) this.getSystemService(VIBRATOR\_SERVICE);

final Button newEntry = (Button) findViewById(R.id.new\_entry);

final Button settings = (Button) findViewById(R.id.settings);

final Button reminders = (Button) findViewById(R.id.reminders\_button);

final Button tracking = (Button) findViewById(R.id.tracking);

settings.setOnClickListener(new View.OnClickListener() {

public void onClick(View v) {

myVib.vibrate(65);

Intent intent = new Intent(MainActivity.this, SettingsActivity.class);

startActivity(intent);

overridePendingTransition(R.transition.in\_from\_left, R.transition.out\_to\_right);

finish();

}

});

reminders.setOnClickListener(new View.OnClickListener() {

public void onClick(View v) {

myVib.vibrate(65);

Intent intent = new Intent(MainActivity.this, RemindersActivity.class);

startActivity(intent);

overridePendingTransition(R.transition.in\_from\_right, R.transition.out\_to\_left);

finish();

}

});

tracking.setOnClickListener(new View.OnClickListener() {

public void onClick(View v) {

myVib.vibrate(65);

Intent intent = new Intent(MainActivity.this, TrackingActivity.class);

startActivity(intent);

overridePendingTransition(R.anim.abc\_slide\_in\_bottom, R.anim.abc\_slide\_out\_top);

finish();

}

});

newEntry.setOnClickListener(new View.OnClickListener() {

public void onClick(View v) {

myVib.vibrate(65);

Intent intent = new Intent(MainActivity.this, NewEntryActivity.class);

startActivity(intent);

overridePendingTransition(R.anim.abc\_slide\_in\_bottom, R.anim.abc\_slide\_out\_top);

finish();

}

});

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.menu\_main, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

// Handle action bar item clicks here. The action bar will

// automatically handle clicks on the Home/Up button, so long

// as you specify a parent activity in AndroidManifest.xml.

int id = item.getItemId();

//noinspection SimplifiableIfStatement

if (id == R.id.action\_settings) {

return true;

}

return super.onOptionsItemSelected(item);

}

}

# NewEntryActivity.class

package com.example.jackson.diabetesapp;

import android.content.Context;

import android.content.Intent;

import android.database.Cursor;

import android.os.Bundle;

import android.support.v7.app.ActionBarActivity;

import android.util.Log;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

import com.google.gson.Gson;

import org.joda.time.DateTime;

import org.json.JSONException;

import org.json.JSONObject;

import org.json.simple.JSONValue;

import java.io.BufferedReader;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.io.StringWriter;

import java.util.ArrayList;

import java.util.Calendar;

import java.util.Date;

import java.util.LinkedHashMap;

import java.util.List;

import java.util.Map;

public class NewEntryActivity extends ActionBarActivity {

private final static String STORETEXT = "data.txt";

static Gson gsonArray;

private EditText bg\_field = null;

private EditText carbs\_field = null;

private EditText protein\_field = null;

private EditText fat\_field = null;

DataHandler handler;

public NewEntryActivity() {

}

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_new\_entry);

getSupportActionBar().setDisplayHomeAsUpEnabled(true);

getSupportActionBar().setHomeButtonEnabled(true);

bg\_field = (EditText) findViewById(R.id.bg\_field);

carbs\_field = (EditText) findViewById(R.id.carbs\_field);

protein\_field = (EditText) findViewById(R.id.protein\_field);

fat\_field = (EditText) findViewById(R.id.fat\_field);

bg\_field.requestFocus();

final Button cancel = (Button) findViewById(R.id.new\_entry\_cancel\_button);

cancel.setOnClickListener(new View.OnClickListener() {

public void onClick(View v) {

MainActivity.myVib.vibrate(65);

Intent intent = new Intent(NewEntryActivity.this, MainActivity.class);

startActivity(intent);

overridePendingTransition(R.anim.abc\_slide\_in\_top, R.anim.abc\_slide\_out\_bottom);

finish();

}

});

final Button okay = (Button) findViewById(R.id.new\_entry\_okay\_button);

okay.setOnClickListener(new View.OnClickListener() {

public void onClick(View v) {

int retId = 0;

int retIds[];

int id = 0;

try {

retIds = retIds();

List<Integer> list = new ArrayList<Integer>(retIds.length);

for (int i : retIds) list.add(i);

} catch (IOException e) {

Toast

.makeText(getApplicationContext(), "FAILED TO RETRIEVE DATA", Toast.LENGTH\_LONG)

.show();

}

//assignId(id);

String bg\_string = bg\_field.getText().toString();

Log.i("BG", "" + bg\_string);

if (bg\_string.equals("")) {

bg\_string = "0";

}

int bg = Integer.parseInt(bg\_string);

String carbs\_string = carbs\_field.getText().toString();

if (carbs\_string.equals("")) {

carbs\_string = "0";

}

int carbs = Integer.parseInt(carbs\_string);

String protein\_string = protein\_field.getText().toString();

if (protein\_string.equals("")) {

protein\_string = "0";

}

int protein = Integer.parseInt(protein\_string);

String fat\_string = fat\_field.getText().toString();

if (fat\_string.equals("")) {

fat\_string = "0";

}

int fat = Integer.parseInt(fat\_string);

MainActivity.myVib.vibrate(65);

/\*

try {

writeJSON(bg, carbs, protein, fat);

} catch (IOException e) {

e.printStackTrace();

}

\*/

Date dat = new Date();

Calendar cal\_now = Calendar.getInstance();

cal\_now.setTime(dat);

Log.i("DATE", "" + dat);

long datLong = cal\_now.getTimeInMillis() / 1000;

int time = (int) datLong;

handler = new DataHandler(getBaseContext());

handler.open();

long dataId = handler.insertData(time, bg, carbs, protein, fat);

handler.close();

Toast

.makeText(getApplicationContext(), "Saved!", Toast.LENGTH\_LONG)

.show();

MainActivity.myVib.vibrate(65);

Intent intent = new Intent(NewEntryActivity.this, MainActivity.class);

startActivity(intent);

overridePendingTransition(R.anim.abc\_slide\_in\_top, R.anim.abc\_slide\_out\_bottom);

finish();

}

});

}

public ArrayList<Integer> recallSQLOLD(String reqColumn) {

handler = new DataHandler(getBaseContext());

int colNum = 0;

int testVar = 0;

int[] retArray;

ArrayList<Integer> retArrayList = new ArrayList<Integer>();

int iLast = 11;

handler.open();

Cursor C = handler.returnData();

if (C.moveToLast()) {

do {

iLast = iLast - 1;

if (iLast == 0) {

break;

}

Log.i("SQL\_DEBUG", "colNum: " + colNum);

testVar = C.getInt(0);

//retArray[iLast] = C.getInt(colNum);

switch (reqColumn) {

case ("time"):

retArrayList.add(C.getInt(0));

case ("bg"):

retArrayList.add(C.getInt(1));

case ("carbs"):

retArrayList.add(C.getInt(2));

case ("protein"):

retArrayList.add(C.getInt(3));

case ("fat"):

retArrayList.add(C.getInt(4));

}

//Log.i("SQL\_DEBUG", "iLast: " + iLast + "\n" + "array[ilast]: " + retArrayList.get(iLast));

} while (C.moveToPrevious());

int iFirst = iLast;

Log.i("SQL\_DEBUG", "iFirst: " + iFirst);

//Log.i("SQL\_DEBUG", "array final: " + retArray);

handler.close();

}

return retArrayList;

}

public void getIds(int[] ids) throws IOException {

try {

ids = retIds();

} catch (IOException e) {

e.printStackTrace();

}

}

public int[] retIds() throws IOException {

Gson retgson = new Gson();

String ret = "";

try {

InputStream inputStream = openFileInput("ids.txt");

if (inputStream != null) {

InputStreamReader inputStreamReader = new InputStreamReader(inputStream);

BufferedReader br = new BufferedReader(inputStreamReader);

String receiveString = "";

StringBuilder stringBuilder = new StringBuilder();

while ((receiveString = br.readLine()) != null) {

stringBuilder.append(receiveString);

}

inputStream.close();

ret = stringBuilder.toString();

}

} catch (FileNotFoundException e) {

} catch (IOException e) {

}

String[] items = ret.replaceAll("\\[", "").replaceAll("\\]", "").split(",");

int[] ids = new int[items.length];

for (int i = 0; i < items.length; i++) {

try {

ids[i] = Integer.parseInt(items[i]);

} catch (NumberFormatException nfe) {

}

;

}

Log.i("DEBUG GSON", "" + retgson);

return ids;

}

public void assignId(int id) throws IOException {

Gson gson = new Gson();

// convert java object to JSON format,

// and returned as JSON formatted string

String json = gson.toJson(id);

StringWriter out = new StringWriter();

try {

JSONValue.writeJSONString(json, out);

} catch (IOException e) {

e.printStackTrace();

}

String jsonText = out.toString();

System.out.print(jsonText);

String FILENAME = "ids.txt";

FileOutputStream fos = null;

try {

fos = openFileOutput(FILENAME, Context.MODE\_APPEND);

fos.write(jsonText.getBytes());

fos.close();

} catch (FileNotFoundException e) {

e.printStackTrace();

}

}

public void writeGSON(int id, int bg, int carbs, int protein, int fat) throws IOException {

Date dat = new Date();

Calendar cal\_now = Calendar.getInstance();

cal\_now.setTime(dat);

Log.i("DATE", "" + dat);

long datLong = cal\_now.getTimeInMillis() / 1000;

int datInt = (int) datLong;

final int bgFinal = bg;

final int carbsFinal = carbs;

final int proteinFinal = protein;

final int fatFinal = fat;

int currentId = id;

final int array[] = {

currentId, datInt, bgFinal, carbsFinal, proteinFinal, fatFinal

};

//gsonArray = Gson.toJson(array, "data.txt");

Gson gson = new Gson();

// convert java object to JSON format,

// and returned as JSON formatted string

String json = gson.toJson(array);

StringWriter out = new StringWriter();

try {

JSONValue.writeJSONString(json, out);

} catch (IOException e) {

e.printStackTrace();

}

String jsonText = out.toString();

System.out.print(jsonText);

String FILENAME = "data.txt";

FileOutputStream fos = null;

try {

fos = openFileOutput(FILENAME, Context.MODE\_APPEND);

fos.write(jsonText.getBytes());

fos.close();

} catch (FileNotFoundException e) {

e.printStackTrace();

}

}

public void writeJSON(int bg, int carbs, int protein, int fat) throws IOException {

DateTime time = new DateTime();

Map obj = new LinkedHashMap();

obj.put("time", time);

obj.put("bg", bg);

obj.put("carbs", carbs);

obj.put("protein", protein);

obj.put("fat", fat);

StringWriter out = new StringWriter();

try {

JSONValue.writeJSONString(obj, out);

} catch (IOException e) {

e.printStackTrace();

}

String jsonText = out.toString();

System.out.print(jsonText);

String FILENAME = "data.txt";

FileOutputStream fos = null;

try {

fos = openFileOutput(FILENAME, Context.MODE\_APPEND);

fos.write(jsonText.getBytes());

fos.close();

} catch (FileNotFoundException e) {

e.printStackTrace();

}

}

public ArrayList recallEntryJson(View v) {

Log.i("DEBUG", "Recalling ENTRY");

String ret = "";

try {

InputStream inputStream = openFileInput("data.txt");

if (inputStream != null) {

InputStreamReader inputStreamReader = new InputStreamReader(inputStream);

BufferedReader bufferedReader = new BufferedReader(inputStreamReader);

String receiveString = "";

StringBuilder stringBuilder = new StringBuilder();

while ((receiveString = bufferedReader.readLine()) != null) {

stringBuilder.append(receiveString);

}

inputStream.close();

ret = stringBuilder.toString();

Log.i("DEBUG", "Ret: " + ret);

JSONObject jsonObj = null;

try {

ret.replaceAll("\"", "\\\\\"");

Log.i("DEBUG", "Ret in TRY: " + ret);

jsonObj = new JSONObject("\"" + ret + "\"");

int t = jsonObj.getInt("time");

Log.i("DEBUG", "tString= " + t);

int bg = jsonObj.getInt("bg");

//System.out.println(t);

String tString = Integer.toString(t);

Log.i("DEBUG", "Ret2: " + ret);

Log.i("DEBUG", "tString= " + tString);

Toast

.makeText(this, tString, Toast.LENGTH\_LONG)

.show();

} catch (JSONException e) {

Toast

.makeText(this, "FAILED", Toast.LENGTH\_LONG)

.show();

} finally {

}

}

} catch (FileNotFoundException e) {

} catch (IOException e) {

}

return null;

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.menu\_new\_entry, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

// Handle action bar item clicks here. The action bar will

// automatically handle clicks on the Home/Up button, so long

// as you specify a parent activity in AndroidManifest.xml.

int id = item.getItemId();

switch (id) {

case android.R.id.home:

Intent homeIntent = new Intent(this, MainActivity.class);

homeIntent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP);

startActivity(homeIntent);

overridePendingTransition(R.anim.abc\_slide\_in\_top, R.anim.abc\_slide\_out\_bottom);

finish();

}

return (super.onOptionsItemSelected(item));

//noinspection SimplifiableIfStatement

}

}

# RemindersActivity.class

package com.example.jackson.diabetesapp;

import android.app.AlarmManager;

import android.app.AlertDialog;

import android.app.PendingIntent;

import android.content.Context;

import android.content.DialogInterface;

import android.content.Intent;

import android.os.Bundle;

import android.os.Vibrator;

import android.support.v7.app.ActionBarActivity;

import android.text.InputType;

import android.util.Log;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.LinearLayout;

import android.widget.NumberPicker;

import android.widget.TabHost;

import android.widget.TextView;

import android.widget.TimePicker;

import android.widget.Toast;

import org.json.simple.JSONValue;

import java.io.BufferedReader;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.io.StringWriter;

import java.text.SimpleDateFormat;

import java.util.Calendar;

import java.util.Date;

import java.util.GregorianCalendar;

import java.util.LinkedHashMap;

import java.util.Map;

public class RemindersActivity extends ActionBarActivity {

public static Vibrator myVib;

int day\_alarm = 0;

int hour\_alarm = 0;

int minute\_alarm = 0;

public int RQ = 119;

public String alarmName, alarmName1, alarmName2, alarmName3, alarmName4, alarmName5, alarmName6, alarmName7;

public String alarmMsg, alarmMsg1, alarmMsg2, alarmMsg3, alarmMsg4, alarmMsg5, alarmMsg6, alarmMsg7;

Context context = this;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_reminders);

getSupportActionBar().setHomeButtonEnabled(true);

getSupportActionBar().setDisplayHomeAsUpEnabled(true);

//Intent serviceIntent=new Intent("com.example.jackson.diabetesapp.AlarmService");

//this.startService(serviceIntent);

RQ = 119;

alarmName = alarmName1 = alarmName2 = alarmName3 = alarmName4 = alarmName5 = alarmName6 = alarmName7 = "Alarm";

alarmMsg = alarmMsg1 = alarmMsg2 = alarmMsg3 = alarmMsg4 = alarmMsg5 = alarmMsg6 = alarmMsg7 = "Hey you!";

final TabHost tabHost = (TabHost) findViewById(R.id.tabHost);

tabHost.setup();

TabHost.TabSpec tabSpec = tabHost.newTabSpec("remindersList");

tabSpec.setContent(R.id.tabRemindersList);

tabSpec.setIndicator("Reminders");

tabHost.addTab(tabSpec);

tabSpec = tabHost.newTabSpec("remindersNew");

tabSpec.setContent(R.id.tabRemindersNew);

tabSpec.setIndicator("+");

tabHost.addTab(tabSpec);

String[] nums = new String[31];

for (int i = 0; i < nums.length; i++)

nums[i] = Integer.toString(i);

NumberPicker np = (NumberPicker) findViewById(R.id.npr);

TimePicker timePicker = (TimePicker) findViewById(R.id.timePicker);

np.setMaxValue(30);

np.setMinValue(0);

np.setValue(0);

np.setWrapSelectorWheel(true);

np.setDisplayedValues(nums);

/\*

File f = new File("/data/data/com.example.jackson.diabetesapp/AlarmData");

if(!f.exists()) {

String[] alarms = new String[7];

for(int i=0; i<alarms.length; i++) {

alarms[i] = "0";

}

try {

writeJSONAlarm(alarms);

} catch (IOException e) {

e.printStackTrace();

}

}

\*/

String[] alarmsStringArray = recallEntryJsonAlarm();

final String[] alarmsFinal = alarmsStringArray;

final Button cancelAlarmsButton = (Button) findViewById(R.id.cancel\_alarms\_button);

final Button editAlarmTextButton = (Button) findViewById(R.id.rename\_alarm\_button);

final Button createAlarmsButton = (Button) findViewById(R.id.create\_alarms\_button);

final TextView alarmtv1 = (TextView) findViewById(R.id.alarmtv1);

final TextView alarmtv2 = (TextView) findViewById(R.id.alarmtv2);

final TextView alarmtv3 = (TextView) findViewById(R.id.alarmtv3);

final TextView alarmtv4 = (TextView) findViewById(R.id.alarmtv4);

final TextView alarmtv5 = (TextView) findViewById(R.id.alarmtv5);

final TextView alarmtv6 = (TextView) findViewById(R.id.alarmtv6);

final TextView alarmtv7 = (TextView) findViewById(R.id.alarmtv7);

/\*

alarmtv1.setOnClickListener(new View.OnClickListener()

{ public void onClick(View v)

{

RQ = 1;

createAlarm(RQ, alarmsFinal);

}

});

alarmtv2.setOnClickListener(new View.OnClickListener()

{ public void onClick(View v)

{

RQ = 2;

createAlarm(RQ, alarmsFinal);

}

});

\*/

cancelAlarmsButton.setOnClickListener(new View.OnClickListener() {

public void onClick(View v) {

cancelAlarm(RQ);

}

});

editAlarmTextButton.setOnClickListener(new View.OnClickListener() {

public void onClick(View v) {

editAlarmText(RQ);

}

});

createAlarmsButton.setOnClickListener(new View.OnClickListener() {

public void onClick(View v) {

createAlarm(RQ, alarmsFinal);

}

});

}

public void editAlarmText(int RQ) {

final int currentRQ = RQ;

AlertDialog.Builder alert = new AlertDialog.Builder(this);

//alert.setTitle("Title");

//alert.setMessage("Message");

// Set an EditText view to get user input

final EditText inputName = new EditText(this);

final EditText inputMsg = new EditText(this);

final TextView tvInputName = new TextView(this);

final TextView tvInputMsg = new TextView(this);

LinearLayout.LayoutParams layoutParams = new LinearLayout.LayoutParams(

LinearLayout.LayoutParams.FILL\_PARENT, LinearLayout.LayoutParams.WRAP\_CONTENT);

layoutParams.setMargins(30, 20, 30, 0);

tvInputName.setTextSize(22);

tvInputMsg.setTextSize(22);

tvInputMsg.setTextSize(22);

tvInputName.setText("Alarm Name:");

tvInputMsg.setText("Alarm Message:");

inputName.setInputType(InputType.TYPE\_TEXT\_FLAG\_CAP\_SENTENCES);

inputMsg.setInputType(InputType.TYPE\_TEXT\_FLAG\_CAP\_SENTENCES);

LinearLayout ll = new LinearLayout(this);

ll.setOrientation(LinearLayout.VERTICAL);

ll.addView(tvInputName, layoutParams);

ll.addView(inputName, layoutParams);

ll.addView(tvInputMsg, layoutParams);

ll.addView(inputMsg, layoutParams);

alert.setView(ll);

Log.i("RQ\_CODE", "RQ in editAlarmText() is: " + RQ);

alert.setPositiveButton("Ok", new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int whichButton) {

if (!inputName.getText().toString().equals("")) {

switch (currentRQ) {

case 1:

alarmName1 = inputName.getText().toString();

break;

case 2:

alarmName2 = inputName.getText().toString();

break;

case 3:

alarmName3 = inputName.getText().toString();

break;

case 4:

alarmName4 = inputName.getText().toString();

break;

case 5:

alarmName5 = inputName.getText().toString();

break;

case 6:

alarmName6 = inputName.getText().toString();

break;

case 7:

alarmName7 = inputName.getText().toString();

break;

default:

alarmName = inputName.getText().toString();

break;

}

}

if (!inputMsg.getText().toString().equals("")) {

switch (currentRQ) {

case 1:

alarmMsg1 = inputMsg.getText().toString();

break;

case 2:

alarmMsg2 = inputMsg.getText().toString();

break;

case 3:

alarmMsg3 = inputMsg.getText().toString();

break;

case 4:

alarmMsg4 = inputMsg.getText().toString();

break;

case 5:

alarmMsg5 = inputMsg.getText().toString();

break;

case 6:

alarmMsg6 = inputMsg.getText().toString();

break;

case 7:

alarmMsg7 = inputMsg.getText().toString();

break;

default:

alarmMsg = inputMsg.getText().toString();

break;

}

}

}

});

alert.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int whichButton) {

// Canceled.

}

});

alert.show();

}

public void createAlarm(int RQ, String[] alarms) {

Log.i("RQ\_CODE", "RQ in createAlarm() is: " + RQ);

final TabHost tabHost = (TabHost) findViewById(R.id.tabHost);

tabHost.setCurrentTab(1);

NumberPicker np = (NumberPicker) findViewById(R.id.npr);

TimePicker timePicker = (TimePicker) findViewById(R.id.timePicker);

day\_alarm = np.getValue();

hour\_alarm = timePicker.getCurrentHour();

minute\_alarm = timePicker.getCurrentMinute();

Calendar cal\_now = Calendar.getInstance();

Calendar cal\_alarm = Calendar.getInstance();

Date dat = new Date();

cal\_now.setTime(dat);

cal\_alarm.setTime(dat);

cal\_alarm.set(Calendar.HOUR\_OF\_DAY, hour\_alarm);

cal\_alarm.set(Calendar.MINUTE, minute\_alarm);

if (cal\_alarm.before(cal\_now)) {

cal\_alarm.add(Calendar.DATE, 1);

}

final Date cal\_alarm\_final = cal\_alarm.getTime();

final long alarm\_time\_ms = cal\_alarm.getTimeInMillis();

AlarmManager am = (AlarmManager) context.getSystemService(Context.ALARM\_SERVICE);

Intent i = new Intent(context, AlarmReceiver.class);

// if (RQ == 0) {

// for (int n = 0; n < alarms.length; n++) {

// Log.i("RQ\_CODE", "Array Values: " + alarms[n]);

// if (alarms[n] == "0") {

// RQ = Integer.parseInt(alarms[n]);

// Log.i("RQ\_CODE", "RQ in FOR " + RQ);

// final int currentRQ = RQ;

setAlarm(am, i, 0, cal\_alarm\_final, alarm\_time\_ms);

// break;

// } else {

// Toast

// .makeText(this, "Delete some first!", Toast.LENGTH\_LONG)

// .show();

// }

//}

//}

}

public void setAlarm(AlarmManager am, Intent i, int currentRQ, Date cal\_alarm\_final, Long alarm\_time\_ms) {

Log.i("RQ\_CODE", "RQ in setAlarm() is: " + RQ);

switch (currentRQ) {

case 1:

i.putExtra("alarmName", alarmName1);

break;

case 2:

i.putExtra("alarmName", alarmName2);

break;

case 3:

i.putExtra("alarmName", alarmName3);

break;

case 4:

i.putExtra("alarmName", alarmName4);

break;

case 5:

i.putExtra("alarmName", alarmName5);

break;

case 6:

i.putExtra("alarmName", alarmName6);

break;

case 7:

i.putExtra("alarmName", alarmName7);

break;

default:

i.putExtra("alarmName", alarmName);

break;

}

switch (currentRQ) {

case 1:

i.putExtra("alarmMsg", alarmMsg1);

break;

case 2:

i.putExtra("alarmMsg", alarmMsg2);

break;

case 3:

i.putExtra("alarmMsg", alarmMsg3);

break;

case 4:

i.putExtra("alarmMsg", alarmMsg4);

break;

case 5:

i.putExtra("alarmMsg", alarmMsg5);

break;

case 6:

i.putExtra("alarmMsg", alarmMsg6);

break;

case 7:

i.putExtra("alarmMsg", alarmMsg7);

break;

default:

i.putExtra("alarmMsg", alarmMsg);

break;

}

PendingIntent alarmIntent = PendingIntent.getBroadcast(context, currentRQ, i, PendingIntent.FLAG\_UPDATE\_CURRENT);

am.set(AlarmManager.RTC\_WAKEUP, alarm\_time\_ms, alarmIntent);

SimpleDateFormat df = new SimpleDateFormat("hh:mm aa");

Date d1 = cal\_alarm\_final;

Calendar c1 = GregorianCalendar.getInstance();

c1.setTime(d1);

String alarmTime = df.format(c1.getTime());

alarmToast(currentRQ, alarmTime);

}

private static PendingIntent createPendingIntent(Context context)

{

Intent alarmServiceIntent = new Intent(context, AlarmService.class);

return null;

}

public void alarmToast(int RQ, String alarmTime) {

final int currentRQ = RQ;

final String currentAlarmTime = alarmTime;

switch (currentRQ) {

case 1:

Toast

.makeText(this, "Set alarm: " + alarmName1 + "\nFor: " + currentAlarmTime, Toast.LENGTH\_LONG)

.show();

break;

case 2:

Toast

.makeText(this, "Set alarm: " + alarmName2 + "\nFor: " + currentAlarmTime, Toast.LENGTH\_LONG)

.show();

break;

case 3:

Toast

.makeText(this, "Set alarm: " + alarmName3 + "\nFor: " + currentAlarmTime, Toast.LENGTH\_LONG)

.show();

break;

case 4:

Toast

.makeText(this, "Set alarm: " + alarmName4 + "\nFor: " + currentAlarmTime, Toast.LENGTH\_LONG)

.show();

break;

case 5:

Toast

.makeText(this, "Set alarm: " + alarmName5 + "\nFor: " + currentAlarmTime, Toast.LENGTH\_LONG)

.show();

break;

case 6:

Toast

.makeText(this, "Set alarm: " + alarmName6 + "\nFor: " + currentAlarmTime, Toast.LENGTH\_LONG)

.show();

break;

case 7:

Toast

.makeText(this, "Set alarm: " + alarmName7 + "\nFor: " + currentAlarmTime, Toast.LENGTH\_LONG)

.show();

break;

default:

Toast

.makeText(this, "Set alarm: " + alarmName + "\nFor: " + currentAlarmTime, Toast.LENGTH\_LONG)

.show();

;

break;

}

}

public void cancelAlarm(int RQ) {

Intent intent = new Intent(context, AlarmReceiver.class);

PendingIntent sender = PendingIntent.getBroadcast(context, RQ, intent, PendingIntent.FLAG\_CANCEL\_CURRENT);

AlarmManager alarmManager = (AlarmManager) context.getSystemService(Context.ALARM\_SERVICE);

alarmManager.cancel(sender);

}

/\*

public void onDestroy(){

createAlarms();

}

\*/

public void writeJSONAlarm(String[] alarms) throws IOException {

Map obj = new LinkedHashMap();

obj.put("0", alarms[0]);

obj.put("1", alarms[1]);

obj.put("2", alarms[2]);

obj.put("3", alarms[3]);

obj.put("4", alarms[4]);

obj.put("5", alarms[5]);

obj.put("6", alarms[6]);

StringWriter out = new StringWriter();

try {

JSONValue.writeJSONString(obj, out);

} catch (IOException e) {

e.printStackTrace();

}

String jsonText = out.toString();

System.out.print(jsonText);

String FILENAME = "AlarmData.txt";

FileOutputStream fos = null;

try {

fos = openFileOutput(FILENAME, Context.MODE\_PRIVATE);

fos.write(jsonText.getBytes());

fos.close();

} catch (FileNotFoundException e) {

e.printStackTrace();

}

}

public String[] recallEntryJsonAlarm() {

String[] ret = new String[7];

try {

InputStream inputStream = openFileInput("AlarmData.txt");

if (inputStream != null) {

InputStreamReader inputStreamReader = new InputStreamReader(inputStream);

BufferedReader bufferedReader = new BufferedReader(inputStreamReader);

String receiveString = "";

StringBuilder stringBuilder = new StringBuilder();

while ((receiveString = bufferedReader.readLine()) != null) {

stringBuilder.append(receiveString);

}

inputStream.close();

for (int i = 0; i < ret.length; i++) {

Log.i("RET\_VALUES", "" + ret[i]);

}

}

} catch (FileNotFoundException e) {

} catch (IOException e) {

}

return ret;

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.menu\_reminders, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

// Handle action bar item clicks here. The action bar will

// automatically handle clicks on the Home/Up button, so long

// as you specify a parent activity in AndroidManifest.xml.

int id = item.getItemId();

switch (item.getItemId()) {

// Respond to the action bar's Up/Home button

case android.R.id.home:

Intent homeIntent = new Intent(this, MainActivity.class);

homeIntent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP);

startActivity(homeIntent);

overridePendingTransition(R.transition.in\_from\_left, R.transition.out\_to\_right);

finish();

return (super.onOptionsItemSelected(item));

}

//noinspection SimplifiableIfStatement

if (id == R.id.action\_settings) {

return true;

}

return super.onOptionsItemSelected(item);

}

}

# RingtonePlayingService.class

package com.example.jackson.diabetesapp;

import android.app.Service;

import android.content.Intent;

import android.content.SharedPreferences;

import android.media.Ringtone;

import android.media.RingtoneManager;

import android.net.Uri;

import android.os.IBinder;

import android.preference.PreferenceManager;

import android.util.Log;

/\*\*

\* Created by Jackson on 4/14/2015.

\*/

public class RingtonePlayingService extends Service {

private Ringtone ringtone;

@Override

public IBinder onBind(Intent intent) {

return null;

}

@Override

public int onStartCommand(Intent intent, int flags, int startId) {

SharedPreferences preference = PreferenceManager.getDefaultSharedPreferences(this);

String strRingtonePreference = preference.getString("notifications\_ringtone\_pref", "DEFAULT\_SOUND");

Uri notification = RingtoneManager.getDefaultUri(RingtoneManager.TYPE\_RINGTONE);

notification = Uri.parse(strRingtonePreference);

ringtone = RingtoneManager.getRingtone(this, notification);

ringtone.play();

Log.i("debugRingtone", "Ringtone Service started");

//this.ringtone = RingtoneManager.getRingtone(this, notification);

return START\_NOT\_STICKY;

}

@Override

public void onDestroy() {

ringtone.stop();

Log.i("debugRingtone", "Ringtone Service destroyed");

}

}

# SettingsActivity.Class

package com.example.jackson.diabetesapp;

import android.annotation.TargetApi;

import android.content.Context;

import android.content.res.Configuration;

import android.media.Ringtone;

import android.media.RingtoneManager;

import android.net.Uri;

import android.os.Build;

import android.os.Bundle;

import android.preference.ListPreference;

import android.preference.Preference;

import android.preference.PreferenceActivity;

import android.preference.PreferenceCategory;

import android.preference.PreferenceFragment;

import android.preference.PreferenceManager;

import android.preference.RingtonePreference;

import android.text.TextUtils;

import java.util.List;

/\*\*

\* A {@link PreferenceActivity} that presents a set of application settings. On

\* handset devices, settings are presented as a single list. On tablets,

\* settings are split by category, with category headers shown to the left of

\* the list of settings.

\* <p/>

\* See <a href="http://developer.android.com/design/patterns/settings.html">

\* Android Design: Settings</a> for design guidelines and the <a

\* href="http://developer.android.com/guide/topics/ui/settings.html">Settings

\* API Guide</a> for more information on developing a Settings UI.

\*/

public class SettingsActivity extends PreferenceActivity {

/\*\*

\* Determines whether to always show the simplified settings UI, where

\* settings are presented in a single list. When false, settings are shown

\* as a master/detail two-pane view on tablets. When true, a single pane is

\* shown on tablets.

\*/

private static final boolean ALWAYS\_SIMPLE\_PREFS = false;

DataHandler handler;

@Override

protected void onPostCreate(Bundle savedInstanceState) {

super.onPostCreate(savedInstanceState);

setupSimplePreferencesScreen();

}

/\*\*

\* Shows the simplified settings UI if the device configuration if the

\* device configuration dictates that a simplified, single-pane UI should be

\* shown.

\*/

private void setupSimplePreferencesScreen() {

if (!isSimplePreferences(this)) {

return;

}

// In the simplified UI, fragments are not used at all and we instead

// use the older PreferenceActivity APIs.

//add container

addPreferencesFromResource(R.xml.pref\_container);

// Add 'general' preferences.

// addPreferencesFromResource(R.xml.pref\_general);

// Add 'notifications' preferences, and a corresponding header.

PreferenceCategory fakeHeader = new PreferenceCategory(this);

fakeHeader.setTitle(R.string.pref\_header\_notifications);

getPreferenceScreen().addPreference(fakeHeader);

addPreferencesFromResource(R.xml.pref\_notification);

// Add 'data and sync' preferences, and a corresponding header.

fakeHeader = new PreferenceCategory(this);

fakeHeader.setTitle(R.string.pref\_header\_data\_sync);

getPreferenceScreen().addPreference(fakeHeader);

addPreferencesFromResource(R.xml.pref\_data\_sync);

// Bind the summaries of EditText/List/Dialog/Ringtone preferences to

// their values. When their values change, their summaries are updated

// to reflect the new value, per the Android Design guidelines.

//bindPreferenceSummaryToValue(findPreference("example\_text"));

//bindPreferenceSummaryToValue(findPreference("example\_list"));

bindPreferenceSummaryToValue(findPreference("notifications\_ringtone\_pref"));

bindPreferenceSummaryToValue(findPreference("sync\_frequency"));

//add Clear data Prefs

addPreferencesFromResource(R.xml.pref\_clear\_data);

//bindPreferenceSummaryToValue(findPreference("clear\_data"));

}

/\*\*

\* {@inheritDoc}

\*/

@Override

public boolean onIsMultiPane() {

return isXLargeTablet(this) && !isSimplePreferences(this);

}

/\*\*

\* Helper method to determine if the device has an extra-large screen. For

\* example, 10" tablets are extra-large.

\*/

private static boolean isXLargeTablet(Context context) {

return (context.getResources().getConfiguration().screenLayout

& Configuration.SCREENLAYOUT\_SIZE\_MASK) >= Configuration.SCREENLAYOUT\_SIZE\_XLARGE;

}

/\*\*

\* Determines whether the simplified settings UI should be shown. This is

\* true if this is forced via {@link #ALWAYS\_SIMPLE\_PREFS}, or the device

\* doesn't have newer APIs like {@link PreferenceFragment}, or the device

\* doesn't have an extra-large screen. In these cases, a single-pane

\* "simplified" settings UI should be shown.

\*/

private static boolean isSimplePreferences(Context context) {

return ALWAYS\_SIMPLE\_PREFS

|| Build.VERSION.SDK\_INT < Build.VERSION\_CODES.HONEYCOMB

|| !isXLargeTablet(context);

}

/\*\*

\* {@inheritDoc}

\*/

@Override

@TargetApi(Build.VERSION\_CODES.HONEYCOMB)

public void onBuildHeaders(List<Header> target) {

if (!isSimplePreferences(this)) {

loadHeadersFromResource(R.xml.pref\_headers, target);

}

}

/\*\*

\* A preference value change listener that updates the preference's summary

\* to reflect its new value.

\*/

private static Preference.OnPreferenceChangeListener sBindPreferenceSummaryToValueListener = new Preference.OnPreferenceChangeListener() {

@Override

public boolean onPreferenceChange(Preference preference, Object value) {

String stringValue = value.toString();

if (preference instanceof ListPreference) {

// For list preferences, look up the correct display value in

// the preference's 'entries' list.

ListPreference listPreference = (ListPreference) preference;

int index = listPreference.findIndexOfValue(stringValue);

// Set the summary to reflect the new value.

preference.setSummary(

index >= 0

? listPreference.getEntries()[index]

: null);

} else if (preference instanceof RingtonePreference) {

// For ringtone preferences, look up the correct display value

// using RingtoneManager.

if (TextUtils.isEmpty(stringValue)) {

// Empty values correspond to 'silent' (no ringtone).

preference.setSummary(R.string.pref\_ringtone\_silent);

} else {

Ringtone ringtone = RingtoneManager.getRingtone(

preference.getContext(), Uri.parse(stringValue));

if (ringtone == null) {

// Clear the summary if there was a lookup error.

preference.setSummary(null);

} else {

// Set the summary to reflect the new ringtone display

// name.

String name = ringtone.getTitle(preference.getContext());

preference.setSummary(name);

}

}

} else {

// For all other preferences, set the summary to the value's

// simple string representation.

preference.setSummary(stringValue);

}

return true;

}

};

/\*\*

\* Binds a preference's summary to its value. More specifically, when the

\* preference's value is changed, its summary (line of text below the

\* preference title) is updated to reflect the value. The summary is also

\* immediately updated upon calling this method. The exact display format is

\* dependent on the type of preference.

\*

\* @see #sBindPreferenceSummaryToValueListener

\*/

private static void bindPreferenceSummaryToValue(Preference preference) {

// Set the listener to watch for value changes.

preference.setOnPreferenceChangeListener(sBindPreferenceSummaryToValueListener);

// Trigger the listener immediately with the preference's

// current value.

sBindPreferenceSummaryToValueListener.onPreferenceChange(preference,

PreferenceManager

.getDefaultSharedPreferences(preference.getContext())

.getString(preference.getKey(), ""));

}

/\*\*

\* This fragment shows general preferences only. It is used when the

\* activity is showing a two-pane settings UI.

\*/

@TargetApi(Build.VERSION\_CODES.HONEYCOMB)

public static class GeneralPreferenceFragment extends PreferenceFragment {

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

addPreferencesFromResource(R.xml.pref\_general);

// Bind the summaries of EditText/List/Dialog/Ringtone preferences

// to their values. When their values change, their summaries are

// updated to reflect the new value, per the Android Design

// guidelines.

bindPreferenceSummaryToValue(findPreference("example\_text"));

bindPreferenceSummaryToValue(findPreference("example\_list"));

}

}

@TargetApi(Build.VERSION\_CODES.HONEYCOMB)

public static class PrefClearDataFragment extends PreferenceFragment {

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

addPreferencesFromResource(R.xml.pref\_clear\_data);

// Bind the summaries of EditText/List/Dialog/Ringtone preferences

// to their values. When their values change, their summaries are

// updated to reflect the new value, per the Android Design

// guidelines.

// bindPreferenceSummaryToValue(findPreference("sync\_frequency"));

}

}

/\*\*

\* This fragment shows notification preferences only. It is used when the

\* activity is showing a two-pane settings UI.

\*/

@TargetApi(Build.VERSION\_CODES.HONEYCOMB)

public static class NotificationPreferenceFragment extends PreferenceFragment {

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

addPreferencesFromResource(R.xml.pref\_notification);

// Bind the summaries of EditText/List/Dialog/Ringtone preferences

// to their values. When their values change, their summaries are

// updated to reflect the new value, per the Android Design

// guidelines.

bindPreferenceSummaryToValue(findPreference("notifications\_ringtone\_pref"));

}

}

/\*\*

\* This fragment shows data and sync preferences only. It is used when the

\* activity is showing a two-pane settings UI.

\*/

@TargetApi(Build.VERSION\_CODES.HONEYCOMB)

public static class DataSyncPreferenceFragment extends PreferenceFragment {

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

addPreferencesFromResource(R.xml.pref\_data\_sync);

// Bind the summaries of EditText/List/Dialog/Ringtone preferences

// to their values. When their values change, their summaries are

// updated to reflect the new value, per the Android Design

// guidelines.

bindPreferenceSummaryToValue(findPreference("sync\_frequency"));

}

}

}

# TrackingActivity.class

package com.example.jackson.diabetesapp;

import android.content.Intent;

import android.content.pm.ActivityInfo;

import android.database.Cursor;

import android.os.Bundle;

import android.support.v7.app.ActionBarActivity;

import android.util.Log;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.widget.AdapterView;

import android.widget.ArrayAdapter;

import android.widget.Spinner;

import com.jjoe64.graphview.GraphView;

import com.jjoe64.graphview.helper.DateAsXAxisLabelFormatter;

import com.jjoe64.graphview.series.DataPoint;

import com.jjoe64.graphview.series.PointsGraphSeries;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

public class TrackingActivity extends ActionBarActivity {

DataHandler handler;

String reqColumn = "bg";

private PointsGraphSeries<DataPoint> series;

//private LineGraphSeries<DataPoint> seriesLine;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setRequestedOrientation(ActivityInfo.SCREEN\_ORIENTATION\_SENSOR);

getSupportActionBar().setHomeButtonEnabled(true);

getSupportActionBar().setDisplayHomeAsUpEnabled(true);

setContentView(R.layout.activity\_tracking);

//INITIALIZE SPINNER

final Spinner spinner = (Spinner) findViewById(R.id.spinner);

// Create an ArrayAdapter using the string array and a default spinner layout

ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this,

R.array.spinner\_array, android.R.layout.simple\_spinner\_item);

// Specify the layout to use when the list of choices appears

adapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item);

// Apply the adapter to the spinner

spinner.setAdapter(adapter);

GraphView graph = (GraphView) findViewById(R.id.graph);

series = new PointsGraphSeries<>();

//seriesLine = new LineGraphSeries<>();

graph.addSeries(series);

//graph.addSeries(seriesLine);

setValues(reqColumn);

spinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {

@Override

public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {

if (spinner.getSelectedItem().toString().toLowerCase().equals("blood glucose vs. time")) {

reqColumn = "bg";

Log.i("DEBUG:", "spinner is bg");

} else if (spinner.getSelectedItem().toString().toLowerCase().equals("carbs vs. time")) {

reqColumn = "carbs";

Log.i("DEBUG:", "spinner is carbs");

} else if (spinner.getSelectedItem().toString().toLowerCase().equals("protein vs. time")) {

reqColumn = "protein";

Log.i("DEBUG:", "spinner is protein");

} else if (spinner.getSelectedItem().toString().toLowerCase().equals("fat vs. time")) {

reqColumn = "fat";

Log.i("DEBUG:", "spinner is fat");

} else {

reqColumn = "bg";

Log.i("DEBUG:", "spinner is fat");

}

setValues(reqColumn);

}

@Override

public void onNothingSelected(AdapterView<?> parent) {

}

});

}

public void setValues(String reqColumn) {

ArrayList<Integer> xArrayList = recallSQL("time");

ArrayList<Integer> yArrayList = recallSQL(reqColumn);

if (xArrayList.size() == 0) {

for (int i = 0; i < 10; i++) {

xArrayList.add(0);

}

}

Log.i("xArrayList: ", "" + xArrayList);

int[] xArray = convertIntegers(xArrayList);

Date[] dateArray = new Date[xArray.length];

for (int i = 0; i < xArray.length; i++) {

Date d = new Date(xArray[i] \* 1000L);

dateArray[i] = (d);

Log.i("DEBUG", "dateArray pos " + i + ": " + dateArray[i]);

}

for (int it = 0; it < xArray.length; it++) {

Log.i("DEBUG", "xARRAY POS " + it + ": " + xArray[it]);

}

//FOR DEBUGGING TIMES:

/\*

dateArray[0]= new Date(1429255810\*1000L);

dateArray[1]= new Date(1429039210\*1000L);

dateArray[2] = new Date(1429555510\*1000L);

\*/

GraphView graph = (GraphView) findViewById(R.id.graph);

series.resetData(generateData(dateArray, yArrayList));

// seriesLine.resetData(generateData(dateArray, yArrayList));

graph.getGridLabelRenderer().setLabelFormatter(new DateAsXAxisLabelFormatter(this));

graph.getGridLabelRenderer().setNumHorizontalLabels(3);//MAX 4

//graph.getViewport().setMinX(dateArray[0].getTime());

// graph.getViewport().setMaxX(dateArray[dateArray.length - 1].getTime());

//graph.getViewport().setXAxisBoundsManual(true);

}

private DataPoint[] generateData(Date[] dateArray, ArrayList<Integer> yArrayList) {

int count = dateArray.length;

DataPoint[] values = new DataPoint[count];

if (yArrayList.size() == 0) {

for (int i = 0; i < 10; i++) {

yArrayList.add(0);

}

}

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

Log.i("xValue:", "" + dateArray[i]);

Log.i("yValue:", "" + yArrayList.get(i));

DataPoint v = new DataPoint(dateArray[i], yArrayList.get(i));

values[i] = v;

}

return values;

}

public ArrayList<Integer> recallSQL(String reqColumn) {

handler = new DataHandler(getBaseContext());

ArrayList<Integer> retArrayList = new ArrayList<>();

int rowNum = 0;

handler.open();

Cursor C = handler.returnData();

retArrayList.clear();

if (C.moveToFirst()) {

do {

rowNum = rowNum + 1;

//retArray[iLast] = C.getInt(colNum);

if (reqColumn.equals("time")) {

retArrayList.add(C.getInt(0));

} else if (reqColumn.equals("bg")) {

retArrayList.add(C.getInt(1));

} else if (reqColumn.equals("carbs")) {

retArrayList.add(C.getInt(2));

} else if (reqColumn.equals("protein")) {

retArrayList.add(C.getInt(3));

} else {

retArrayList.add(C.getInt(4));

}

} while (C.moveToNext() && rowNum < 10);

Log.i("SQL\_DEBUG", "rowNum: " + rowNum);

handler.close();

}

return retArrayList;

}

public static int[] convertIntegers(List<Integer> retArrayList) {

int[] ret = new int[retArrayList.size()];

Log.i("SQL\_DEBUG", "ret size: " + ret.length);

for (int i = 0; i < ret.length; i++) {

ret[i] = retArrayList.get(i);

}

return ret;

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.menu\_tracking, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

// Handle action bar item clicks here. The action bar will

// automatically handle clicks on the Home/Up button, so long

// as you specify a parent activity in AndroidManifest.xml.

int id = item.getItemId();

switch (item.getItemId()) {

// Respond to the action bar's Up/Home button

case android.R.id.home:

Intent homeIntent = new Intent(this, MainActivity.class);

homeIntent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP);

startActivity(homeIntent);

overridePendingTransition(R.transition.in\_from\_left, R.transition.out\_to\_right);

finish();

return (super.onOptionsItemSelected(item));

}

//noinspection SimplifiableIfStatement

if (id == R.id.action\_settings) {

return true;

}

return super.onOptionsItemSelected(item);

}

}

Resource Files (except drawables, menus):

# Activity\_alarm.xml

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

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:orientation="vertical"

tools:context="com.example.jackson.diabetesapp.AlarmActivity">

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:layout\_weight="1"

android:orientation="vertical">

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:layout\_weight="1"

android:orientation="vertical"

android:textAlignment="center">

<TextView

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="Alarm Name"

android:id="@+id/textViewAlarmName"

android:textSize="40dp"

android:layout\_gravity="center\_horizontal"

android:gravity="bottom|center\_horizontal"

android:textStyle="bold"

android:layout\_weight="1"

android:textAlignment="center" />

<TextView

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="Alarm Message"

android:id="@+id/textViewAlarmMsg"

android:textSize="28dp"

android:layout\_weight=".5"

android:textAlignment="center"

android:singleLine="false"

android:gravity="center\_horizontal"

android:layout\_gravity="center\_vertical"

android:textStyle="bold" />

</LinearLayout>

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:layout\_weight="1"

android:orientation="vertical"

>

<Button

android:id="@+id/shutup\_alarm"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:text="CANCEL"

android:layout\_gravity="center"

android:minHeight="100dp"

android:background="@drawable/button\_purple"

android:textSize="34sp"

android:textStyle="bold"

android:shadowColor="#ffffffff"

android:textColor="#ff351034" />

</LinearLayout>

</LinearLayout>

</LinearLayout>

# activity\_delete\_data.xml

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

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:paddingBottom="@dimen/activity\_vertical\_margin"

tools:context="com.example.jackson.diabetesapp.DeleteData">

</RelativeLayout>

# activity\_main.xml

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

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:padding="0dp"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:weightSum="2"

tools:context=".MainActivity"

>

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:orientation="horizontal"

android:padding="0dp"

android:showDividers="none"

android:textSize="@dimen/button\_font\_size">

<Button

android:id="@+id/settings"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:layout\_margin="10dp"

android:layout\_weight="1"

android:background="@drawable/button\_solid"

android:padding="20dp"

android:text="@string/settings"

android:textColor="@color/button\_text\_blue"

android:textSize="@dimen/button\_font\_size" />

<Button

android:id="@+id/reminders\_button"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:layout\_margin="10dp"

android:layout\_weight="1"

android:background="@drawable/button\_solid"

android:padding="20dp"

android:text="@string/reminders"

android:textColor="@color/button\_text\_blue"

android:textSize="@dimen/button\_font\_size" />

</LinearLayout>

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:minHeight="230dp"

android:orientation="vertical"

android:padding="0dp"

android:showDividers="none">

<Button

android:id="@+id/tracking"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:layout\_margin="10dp"

android:layout\_weight="1"

android:background="@drawable/button\_solid"

android:padding="20dp"

android:text="@string/tracking"

android:textColor="@color/button\_text\_blue"

android:textSize="@dimen/button\_font\_size" />

<Button

android:id="@+id/new\_entry"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:layout\_margin="10dp"

android:layout\_weight="1"

android:background="@drawable/button\_solid"

android:padding="20dp"

android:text="@string/new\_entry"

android:textColor="@color/button\_text\_blue"

android:textSize="@dimen/button\_font\_size" />

</LinearLayout>

</LinearLayout>

# activity\_new\_entry.xml

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

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:weightSum="2"

tools:context="com.example.jackson.diabetesapp.NewEntryActivity">

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="8dp"

android:layout\_weight=".1"

android:background="@drawable/border"

android:orientation="horizontal">

<TextView

android:id="@+id/new\_entry\_bg"

android:layout\_width="180dp"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_vertical|bottom"

android:layout\_weight="1.5"

android:capitalize="words"

android:maxWidth="180dp"

android:text="@string/blood\_glucose\_mg\_dl"

android:textSize="@dimen/entry\_font\_size" />

<EditText

android:id="@+id/bg\_field"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="bottom|center"

android:layout\_marginLeft="65dp"

android:layout\_weight="1"

android:gravity="center\_horizontal"

android:hint="0"

android:inputType="number"

android:singleLine="true"

android:textSize="20sp">

<requestFocus />

</EditText>

</LinearLayout>

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="8dp"

android:layout\_weight=".1"

android:background="@drawable/border"

android:orientation="horizontal">

<TextView

android:id="@+id/new\_entry\_carbs"

android:layout\_width="180dp"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_vertical|bottom"

android:layout\_weight="1.5"

android:maxWidth="180dp"

android:text="@string/carbohydrates\_g"

android:textSize="@dimen/entry\_font\_size" />

<EditText

android:id="@+id/carbs\_field"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="bottom|center"

android:layout\_marginLeft="65dp"

android:layout\_weight="1"

android:gravity="center\_horizontal"

android:hint="0"

android:inputType="number"

android:singleLine="true"

android:textSize="20sp" />

</LinearLayout>

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="8dp"

android:layout\_weight=".1"

android:background="@drawable/border"

android:orientation="horizontal">

<TextView

android:id="@+id/new\_entry\_protein"

android:layout\_width="180dp"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_vertical|bottom"

android:layout\_weight="1.5"

android:capitalize="words"

android:text="@string/protein\_g"

android:textSize="@dimen/entry\_font\_size" />

<EditText

android:id="@+id/protein\_field"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="bottom|center"

android:layout\_marginLeft="65dp"

android:layout\_weight="1"

android:gravity="center\_horizontal"

android:hint="0"

android:inputType="number"

android:singleLine="true"

android:textSize="20sp" />

</LinearLayout>

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_weight=".1"

android:background="@drawable/border"

android:orientation="horizontal">

<TextView

android:id="@+id/new\_entry\_fat"

android:layout\_width="180dp"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_vertical|bottom"

android:layout\_weight="1.5"

android:capitalize="words"

android:text="@string/fat\_g"

android:textSize="@dimen/entry\_font\_size" />

<EditText

android:id="@+id/fat\_field"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="bottom|center"

android:layout\_marginLeft="65dp"

android:layout\_weight="1"

android:gravity="center\_horizontal"

android:hint="0"

android:inputType="number"

android:singleLine="true"

android:textSize="20sp" />

</LinearLayout>

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:layout\_weight="1"

android:orientation="horizontal">

<Button

android:id="@+id/new\_entry\_cancel\_button"

android:layout\_width="wrap\_content"

android:layout\_height="100dp"

android:layout\_margin="10dp"

android:text="@string/cancel"

android:layout\_weight="1" />

<Button

android:id="@+id/new\_entry\_okay\_button"

android:layout\_width="wrap\_content"

android:layout\_height="100dp"

android:layout\_margin="10dp"

android:text="@string/okay"

android:layout\_weight="1" />

</LinearLayout>

</LinearLayout>

# activity\_reminders.xml

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

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

tools:context="com.example.jackson.diabetesapp.RemindersActivity">

<TabHost

android:id="@+id/tabHost"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1">

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:orientation="vertical">

<TabWidget

android:id="@android:id/tabs"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"></TabWidget>

<FrameLayout

android:id="@android:id/tabcontent"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent">

<LinearLayout

android:id="@+id/tabRemindersList"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:orientation="vertical">

<ScrollView

android:id="@+id/scrollView"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content">

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical">

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:layout\_marginBottom="2dp"

android:layout\_marginTop="8dp"

android:background="@drawable/border"

android:gravity="fill\_horizontal"

android:orientation="horizontal">

<TextView

android:id="@+id/alarmtv1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="left"

android:layout\_weight="1"

android:minLines="1"

android:padding="10dp"

android:textSize="22sp"

android:text="Bolus"

android:clickable="true" />

<Switch

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/switch1alarm"

android:layout\_gravity="center\_vertical" />

</LinearLayout>

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:layout\_marginBottom="2dp"

android:layout\_marginTop="8dp"

android:background="@drawable/border"

android:gravity="fill\_horizontal"

android:orientation="horizontal">

<TextView

android:id="@+id/alarmtv2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="left"

android:layout\_weight="1"

android:minLines="1"

android:padding="10dp"

android:textSize="22sp"

android:text="Set Change"

android:clickable="true" />

<Switch

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/switch2alarm"

android:layout\_gravity="center\_vertical"

android:checked="false" />

</LinearLayout>

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:layout\_marginBottom="2dp"

android:layout\_marginTop="8dp"

android:background="@drawable/border"

android:gravity="fill\_horizontal"

android:orientation="horizontal">

<TextView

android:id="@+id/alarmtv3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="left"

android:layout\_weight="1"

android:minLines="1"

android:padding="10dp"

android:textSize="22sp"

android:text="Test Before Lunch"

android:clickable="true" />

<Switch

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/switch3alarm"

android:layout\_gravity="center\_vertical"

android:checked="false" />

</LinearLayout>

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:layout\_marginBottom="2dp"

android:layout\_marginTop="8dp"

android:background="@drawable/border"

android:gravity="fill\_horizontal"

android:orientation="horizontal">

<TextView

android:id="@+id/alarmtv4"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="left"

android:layout\_weight="1"

android:minLines="1"

android:padding="10dp"

android:textSize="22sp"

android:text="Test Before Dinner"

android:clickable="true" />

<Switch

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/switch1"

android:layout\_gravity="center\_vertical"

android:checked="false" />

</LinearLayout>

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:layout\_marginBottom="2dp"

android:layout\_marginTop="8dp"

android:background="@drawable/border"

android:gravity="fill\_horizontal"

android:orientation="horizontal">

<TextView

android:id="@+id/alarmtv5"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="left"

android:layout\_weight="1"

android:minLines="1"

android:padding="10dp"

android:textSize="22sp"

android:clickable="true" />

<Switch

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/switch2"

android:layout\_gravity="center\_vertical"

android:checked="false" />

</LinearLayout>

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:layout\_marginBottom="2dp"

android:layout\_marginTop="8dp"

android:background="@drawable/border"

android:gravity="fill\_horizontal"

android:orientation="horizontal">

<TextView

android:id="@+id/alarmtv6"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="left"

android:layout\_weight="1"

android:minLines="1"

android:padding="10dp"

android:textSize="22sp"

android:clickable="true" />

<Switch

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/switch3"

android:layout\_gravity="center\_vertical"

android:checked="false" />

</LinearLayout>

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:layout\_marginBottom="2dp"

android:layout\_marginTop="8dp"

android:background="@drawable/border"

android:gravity="fill\_horizontal"

android:orientation="horizontal">

<TextView

android:id="@+id/alarmtv7"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="left"

android:layout\_weight="1"

android:minLines="1"

android:padding="10dp"

android:textSize="22sp"

android:clickable="true" />

<Switch

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/switch4"

android:layout\_gravity="center\_vertical"

android:checked="false" />

</LinearLayout>

</LinearLayout>

</ScrollView>

</LinearLayout>

<LinearLayout

android:id="@+id/tabRemindersNew"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:orientation="vertical">

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:background="@drawable/border\_blue"

android:layout\_weight="1.2"

android:layout\_margin="8dp"

android:gravity="fill\_horizontal|fill\_vertical">

<TimePicker

android:id="@+id/timePicker"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:layout\_weight="1"

android:layout\_margin="2dp" />

</LinearLayout>

<RelativeLayout

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:orientation="horizontal"

android:background="@drawable/border\_blue"

android:layout\_margin="8dp">

<NumberPicker

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/npr"

android:layout\_gravity="center\_horizontal"

android:layout\_weight="1"

android:layout\_centerInParent="true"

android:textSize="22sp"

android:inputType="number" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Day(s)"

android:layout\_weight="1"

android:textSize="22sp"

android:layout\_gravity="center\_vertical"

android:layout\_centerVertical="true"

android:layout\_toEndOf="@id/npr"

android:layout\_alignParentBottom="false"

android:layout\_margin="5dp" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Every"

android:layout\_weight="1"

android:textSize="22dp"

android:layout\_margin="5dp"

android:layout\_gravity="center\_vertical"

android:layout\_centerVertical="true"

android:layout\_toLeftOf="@id/npr" />

</RelativeLayout>

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="90dp"

android:orientation="horizontal">

<Button

android:id="@+id/rename\_alarm\_button"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:text="Edit Alarm Text"

android:layout\_weight="1" />

<Button

android:id="@+id/create\_alarms\_button"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:text="Create"

android:layout\_weight="1" />

<Button

android:id="@+id/cancel\_alarms\_button"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:text="Cancel"

android:layout\_weight="1" />

</LinearLayout>

</LinearLayout>

</FrameLayout>

</LinearLayout>

</TabHost>

</LinearLayout>

# activity\_tracking.xml

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

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:paddingBottom="@dimen/activity\_vertical\_margin"

tools:context="com.example.jackson.diabetesapp.TrackingActivity"

android:orientation="vertical">

<Spinner

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/spinner"

android:spinnerMode="dropdown"

android:clickable="true" />

<com.jjoe64.graphview.GraphView

android:layout\_width="match\_parent"

android:layout\_height="200dp"

android:id="@+id/graph" />

</LinearLayout>

# in\_from\_left.xml

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

<set xmlns:android="http://schemas.android.com/apk/res/android">

<translate

android:interpolator="@android:anim/decelerate\_interpolator"

android:fromXDelta="-100%p"

android:toXDelta="0"

android:duration="300" />

</set>

# in\_from\_right.xml

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

<set xmlns:android="http://schemas.android.com/apk/res/android">

<translate

android:interpolator="@android:anim/decelerate\_interpolator"

android:fromXDelta="100%p"

android:toXDelta="0"

android:duration="300" />

</set>

# out\_to\_left.xml

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

<set xmlns:android="http://schemas.android.com/apk/res/android">

<translate

android:interpolator="@android:anim/decelerate\_interpolator"

android:fromXDelta="0"

android:toXDelta="-100%p"

android:duration="300" />

</set>

# out\_to\_right.xml

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

<set xmlns:android="http://schemas.android.com/apk/res/android">

<translate

android:interpolator="@android:anim/decelerate\_interpolator"

android:fromXDelta="0"

android:toXDelta="100%p"

android:duration="300" />

</set>

# dimens.xml

<resources>

<!-- Default screen margins, per the Android Design guidelines. -->

<dimen name="activity\_horizontal\_margin">16dp</dimen>

<dimen name="activity\_vertical\_margin">16dp</dimen>

<dimen name="button\_font\_size">24sp</dimen>

<dimen name="entry\_font\_size">22sp</dimen>

entry\_font\_size

</resources>

# strings.xml

<resources>

<string name="app\_name">onTrack</string>

<string name="title\_activity\_new\_entry">Enter the amounts:</string>

<string name="action\_settings">Settings</string>

<string name="cancel">Cancel</string>

<string name="create">Create</string>

<string name="okay">Okay</string>

<string name="XML">XML</string>

<string name="tracking">Tracking</string>

<string name="settings">Settings</string>

<string name="reminders">Reminders</string>

<string name="new\_entry">New Entry</string>

<string name="blood\_glucose\_mg\_dl">Blood Glucose\n(mg/dL)</string>

<string name="carbohydrates\_g">Carbohydrates (g)</string>

<string name="protein\_g">Protein (g)</string>

<string name="fat\_g">Fat (g)</string>

<string name="title\_activity\_reminders">RemindersActivity</string>

<string name="title\_section1">Section 1</string>

<string name="title\_section2">Section 2</string>

<string name="title\_section3">Section 3</string>

<string name="hello\_world">Hello world!</string>

<string name="title\_activity\_reminders\_test">RemindersTestActivity</string>

<string name="title\_activity\_main">Main Menu</string>

<string name="title\_activity\_alarm\_receiver">AlarmReceiverActivity</string>

<string name="title\_activity\_alarm">AlarmActivity</string>

<string name="title\_activity\_tracking">TrackingActivity</string>

<string name="pref\_header\_clear">Clear User Data</string>

<string-array name="spinner\_array">

<item>Blood Glucose vs. Time</item>

<item>Carbs vs. Time</item>

<item>Protein vs. Time</item>

<item>Fat vs. Time</item>

</string-array>

<string name="title\_activity\_delete\_data">DeleteData</string>

</resources>

# strings\_activity\_settings.xml

<resources>

<string name="title\_activity\_settings">Settings</string>

<!-- Strings related to Settings -->

<!-- Example General settings -->

<string name="pref\_header\_general">General</string>

<string name="pref\_title\_social\_recommendations">Enable social recommendations</string>

<string name="pref\_description\_social\_recommendations">Recommendations for people to contact

based on your message history

</string>

<string name="pref\_title\_display\_name">Display name</string>

<string name="pref\_default\_display\_name">John Smith</string>

<string name="pref\_title\_add\_friends\_to\_messages">Add friends to messages</string>

<string-array name="pref\_example\_list\_titles">

<item>Always</item>

<item>When possible</item>

<item>Never</item>

</string-array>

<string-array name="pref\_example\_list\_values">

<item>1</item>

<item>0</item>

<item>-1</item>

</string-array>

<!-- Example settings for Data & Sync -->

<string name="pref\_header\_data\_sync">Data &amp; sync</string>

<string name="pref\_title\_sync\_frequency">Sync frequency</string>

<string-array name="pref\_sync\_frequency\_titles">

<item>15 minutes</item>

<item>30 minutes</item>

<item>1 hour</item>

<item>3 hours</item>

<item>6 hours</item>

<item>Never</item>

</string-array>

<string-array name="pref\_sync\_frequency\_values">

<item>15</item>

<item>30</item>

<item>60</item>

<item>180</item>

<item>360</item>

<item>-1</item>

</string-array>

<string name="pref\_title\_system\_sync\_settings">System sync settings</string>

<!-- Example settings for Notifications -->

<string name="pref\_header\_notifications">Notifications</string>

<string name="pref\_title\_new\_message\_notifications">New message notifications</string>

<string name="pref\_title\_ringtone">Ringtone</string>

<string name="pref\_ringtone\_silent">Silent</string>

<string name="pref\_title\_vibrate">Vibrate</string>

</resources>

# styles.xml

<resources>

<!-- Base application theme. -->

<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">

<!-- Customize your theme here. -->

</style>

<style name="Theme.UserDialog" parent="android:style/Theme.Dialog">

<item name="android:windowFrame">@null</item>

<item name="android:windowIsFloating">true</item>

<item name="android:windowIsTranslucent">true</item>

<item name="android:windowNoTitle">true</item>

<item name="android:background">@android:color/transparent</item>

<item name="android:windowBackground">@android:color/transparent</item>

</style>

<color name="button\_text\_blue">#33CCFF</color>

<color name="button\_text\_dark\_blue">#0000FF</color>

<color name="button\_text\_red">#FF3232</color>

<color name="button\_text\_purple">#551A8B</color>

<color name="button\_text\_green">#00eb46</color>

<color name="button\_text\_orange">#CC8400</color>

<color name="button\_text\_white">#ffffff</color>

<color name="black">#000000</color>

</resources>

Welcome to the end :D!