

Android Gym Application Utilizing Near Field Communications (NFC) CM0343

FULL CODE APPENDIX
BRETT STEVENS

Contents

Appendices.....	2
AddExerciseActivity.....	2
AddExerciseEntryActivity	5
AddExerciseEntryManualActivity	8
CalendarActivity	16
DBActivity.....	24
EditExerciseActivity	26
HelpActivity	31
MainActivity	32
NFCActivity.....	37
StopWatchActivity.....	43
ViewExercisesActivity	46
Db.....	48
NFCReader	56
activity_calendar.xml	58
activity_db_add_entry_manual.xml	59
activity_db_add_entry.xml	61
activity_db_add_exercise.xml.....	63
activity_db_edit_exercise.xml	64
activity_db_view_exercises.xml.....	65
activity_db.xml.....	66
activity_help.xml.....	67
activity_main.xml.....	68
activity_nfc.xml	70
activity_stopwatch.xml	71

Appendices

AddExerciseActivity

```
package com.brettstevens.gymdiary;

import com.brettstevens.gymdiary.R;
import com.brettstevens.gymdiary.db.Db;
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class AddExerciseActivity extends Activity {

    //initate database

    Db exerciseDb = new Db(this);

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_db_add_exercise);

        //Set add exercise button
        final Button addExercise = (Button) findViewById(R.id.addEntryButton);

        //add listener to button
        addExercise.setOnClickListener(
            new View.OnClickListener() {
                @Override
                public void onClick(View v) {
```

```

//here takes what is in the current Edit text box as a string
EditText exerciseName = (EditText)
findViewById(R.id.exerciseNameTextField);

String exerciseNameText = exerciseName.getText().toString();

//Check if text box is empty, if it is show warning, if not attempt
to add to database

if(!exerciseNameText.isEmpty()) {

    //If successful show added in toast pop up otherwise
    display error message

    if(exerciseDb.addExercise(exerciseNameText, -1)) {
        toast(exerciseNameText + " added");
    } else {
        toast("There was an error adding " +
exerciseNameText);
    }
} else {
    toast("Exercise name field is empty");
}
}

};
}

//Method to send toast, and clear text box to empty
protected void toast(String message) {
    Toast.makeText(this, new String(message), Toast.LENGTH_SHORT).show();
    EditText textField = (EditText) findViewById(R.id.exerciseNameTextField);
    textField.setText("", TextView.BufferType.EDITABLE);
}

```

```
@Override  
  
public boolean onCreateOptionsMenu(Menu menu) {  
    // Inflate the menu; this adds items to the action bar if it is present.  
    getMenuInflater().inflate(R.menu.main, menu);  
    return true;  
}  
}
```

AddExerciseEntryActivity

```
package com.brettstevens.gymdiary;

import java.util.List;

import com.brettstevens.gymdiary.db.Db;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.TextView;

public class AddExerciseEntryActivity extends Activity {
    //Initialize database
    private Db gymDiaryDb = new Db(this);

    @Override

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_db_add_entry);
        // Container that keeps the intent and various information
        final Bundle applicationBundle = getIntent().getExtras();
        //Sets up name label , and set the text of the label to what ever exercise was scanned
        TextView nameLabel = (TextView) findViewById(R.id.addEntryNameLabel);
        nameLabel.setText(applicationBundle.getString("name"));

        //Creates a new array to store 10 strings
        String[] sets = new String[10];
        //add 1 - 10
        for (int i = 1; i < 11; i++) {
            sets[i-1] = String.valueOf(i);
        }
        //add 1-25
        String[] repetitions = new String[25];

        for (int i = 1; i < 26; i++) {
            repetitions[i-1] = String.valueOf(i);
        }
        //add 0.5-300
```

```

String[] weight = new String[600];

Double weightTotal = 0.5;

for (int i = 1; i < 601; i++) {
    Double tempWeight = weightTotal;
    weight[i - 1] = String.valueOf(tempWeight);

    //increment weight total by 0.5
    weightTotal += 0.5;
}
//Sets up the dialogue ready to be shown
final AlertDialog.Builder dialog = new AlertDialog.Builder(this);
dialog.setTitle("Success");
dialog.setMessage("Entry was successfully added\nPress OK to return to the home
screen");
dialog.setPositiveButton("OK", new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int which) {
        Intent goHome = new Intent(getApplicationContext(), MainActivity.class);
        //clearing the current stack on return to home screen(clears history)
        goHome.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(goHome);
        //complete this activity
        finish();
    }
});
//Assigning the spinner to variables that we can use later
final Spinner setsSpinner = (Spinner) findViewById(R.id.addEntrySetsSpinner);
final Spinner repsSpinner = (Spinner) findViewById(R.id.addEntryRepsSpinner);
final Spinner weightSpinner = (Spinner) findViewById(R.id.addEntryWeightSpinner);

//Containers that hold the values of the spinners
ArrayAdapter<String> setsAdapter = new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item, sets);
ArrayAdapter<String> repsAdapter = new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item, repetitions);
ArrayAdapter<String> weightAdapter = new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item, weight);

//Setting the drop down view of the spinner
setsAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
repsAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
weightAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

//Adding the values to the spinners
setsSpinner.setAdapter(setsAdapter);
repsSpinner.setAdapter(repsAdapter);

```

```

weightSpinner.setAdapter(weightAdapter);

//assign the button to a variable (add entry)
final Button addExercise = (Button) findViewById(R.id.addEntryButton);

//Sets listener on the button
addExercise.setOnClickListener(
    new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            //Uses TagID to get exercise ID from Database
            List<String> exerciseld =
gymDiaryDb.getIdFromTagId((applicationBundle.getInt("tagId")));

//If the database returns one entry, add entry to the database
based on selected values
            if(exerciseld.size() == 1) {

                //If an entry is added to the database, display dialog
                if(gymDiaryDb.addEntry(exerciseld.get(0),
setsSpinner.getSelectedItem().toString(), repsSpinner.getSelectedItem().toString(),
weightSpinner.getSelectedItem().toString(), applicationBundle.getString("dateAdded"))) {
                    dialog.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.main, menu);
    return true;
}
}

```


AddExerciseEntryManualActivity

```
package com.brettstevens.gymdiary;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
import java.util.HashMap;
import java.util.List;

import com.brettstevens.gymdiary.db.Db;

import android.annotation.SuppressLint;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.Toast;

//
public class AddExerciseEntryManualActivity extends Activity {
```

```

//Initialize database and global variables
private HashMap<String, String> listOfAllExercises;

private Db gymDiaryDb = new Db(this);

private String date = "";

@SuppressLint("SimpleDateFormat")
@Override

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_db_add_entry_manual);

    // Container that keeps the intent and various information
    final Bundle applicationBundle = getIntent().getExtras();

    String from = applicationBundle.getString("from");

    //Checks to see if from is equal to "cal"
    if(from.equals("cal")) {

        //this is error messages used for debugging
        Log.d("GymDiary", "Sent from CalendarView");
        Calendar selectedDate = Calendar.getInstance();

        //Take date year, month and day from calendar view
        selectedDate.set(Calendar.YEAR, applicationBundle.getInt("year"));
        selectedDate.set(Calendar.MONTH, applicationBundle.getInt("month"));
        selectedDate.set(Calendar.DAY_OF_MONTH, applicationBundle.getInt("day"));
    }
}

```

```

//Formats date for database insertion , if not sent from calendar view , adds
with current date

```

```

        date = new SimpleDateFormat("dd-MM-yyyy
HH:mm:ss").format(selectedDate.getTime());

```

```

    } else {

```

```

        date = new SimpleDateFormat("dd-MM-yyyy HH:mm:ss").format(new Date());

```

```

    }

```

```

//debug message

```

```

Log.d("GymDiary", date);

```

```

//Setting string 1- 10

```

```

String[] sets = new String[10];

```

```

for (int i = 1; i < 11; i++) {

```

```

    sets[i-1] = String.valueOf(i);

```

```

}

```

```

//setting string size 1 - 25

```

```

String[] repetitions = new String[25];

```

```

for (int i = 1; i < 26; i++) {

```

```

    repetitions[i-1] = String.valueOf(i);

```

```

}

```

```

//add 0.5-300

```

```

String[] weight = new String[600];

```

```

Double weightTotal = 0.5;

```

```

for (int i = 1; i < 601; i++) {
    Double tempWeight = weightTotal;
    weight[i - 1] = String.valueOf(tempWeight);
    //increment weight total by 0.5
    weightTotal += 0.5;
}

//Sets up the dialogue ready to be shown
    final AlertDialog.Builder dialog = new AlertDialog.Builder(this);
    dialog.setTitle("Success");
    dialog.setMessage("Entry was successfully added\nPress OK to return to the home
screen");

    dialog.setPositiveButton("OK", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int which) {
            Intent goHome = new Intent(getApplicationContext(), MainActivity.class);

            //clearing the current stack on return to home screen(clears history)
            goHome.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(goHome);

            //complete this activity
            finish();
        }
    });

    //get all exercises from database
listOfAllExercises = gymDiaryDb.getExercises();

```

```
//Creates list of exercises

List<String> exerciseList = new ArrayList<String>();

//debug message
Log.d("GymDiary", "Enumerating values from keys");

for (String key : listOfAllExercises.keySet()) {
    exerciseList.add(key);
}

//debug message
Log.d("GymDiary", "Setting spinners");

//Assigning the spinner to variables that we can use later
final Spinner setsSpinner = (Spinner) findViewById(R.id.addEntryManualSetsSpinner);
final Spinner repsSpinner = (Spinner) findViewById(R.id.addEntryManualRepsSpinner);
final Spinner weightSpinner = (Spinner) findViewById(R.id.addEntryManualWeightSpinner);
final Spinner exercisesSpinner = (Spinner) findViewById(R.id.addEntryManualExerciseSpinner);

//debug message
Log.d("GymDiary", "Setting adapters");

//Containers that hold the values of the spinners

ArrayAdapter<String> setsAdapter = new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item, sets);

ArrayAdapter<String> repsAdapter = new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item, repetitions);
```

```

    ArrayAdapter<String> weightAdapter = new ArrayAdapter<String>(this,
    android.R.layout.simple_spinner_item, weight);

```

```

    ArrayAdapter<String> exerciseAdapter = new ArrayAdapter<String>(this,
    android.R.layout.simple_spinner_item, exerciseList);

```

```

//debug message

```

```

Log.d("GymDiary", "Setting drop down views");

```

```

//Setting the drop down view of the spinner

```

```

setsAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

```

```

repsAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

```

```

weightAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

```

```

exerciseAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

```

```

//debug message

```

```

Log.d("GymDiary", "Adding adapters to spinners");

```

```

//Adding the values to the spinners

```

```

setsSpinner.setAdapter(setsAdapter);

```

```

repsSpinner.setAdapter(repsAdapter);

```

```

weightSpinner.setAdapter(weightAdapter);

```

```

exercisesSpinner.setAdapter(exerciseAdapter);

```

```

//assign the button to a variable (add entry manual)

```

```

final Button addExercise = (Button) findViewById(R.id.addEntryManualButton);

```

```

//add listener to button

```

```

addExercise.setOnClickListener(

```

```

    new View.OnClickListener() {

```

```

@Override

public void onClick(View v) {

    String exerciseNameText =
exercisesSpinner.getSelectedItem().toString();

    //If true populate the spinner else display error message

    if(gymDiaryDb.addEntry(listOfAllExercises.get(exerciseNameText),
setsSpinner.getSelectedItem().toString(), repsSpinner.getSelectedItem().toString(),
weightSpinner.getSelectedItem().toString(), date)) {

        dialog.show();

    } else {

        toast("Something went wrong whilst adding the entry to
the database");

    }

}

};
}

protected void toast(String message) {

    //setup toast

    Toast.makeText(this, new String(message), Toast.LENGTH_SHORT).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.main, menu);

```

```
        return true;
    }
}
```


CalendarActivity

```
package com.brettstevens.gymdiary;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.List;

import android.annotation.SuppressLint;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.View;
import android.view.View.OnLongClickListener;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemLongClickListener;
import android.widget.ArrayAdapter;
import android.widget.CalendarView;
import android.widget.CalendarView.OnDateChangeListener;
import android.widget.ListView;
import android.widget.Toast;

import com.brettstevens.gymdiary.db.Db;
```

```

public class CalendarActivity extends Activity {

    //Initialize database and various variables
    protected Db gymDiaryDb = new Db(this);
    protected ArrayList<HashMap<String, String>> exerciseOnDayArrayList;
    protected int currentlySelectedDay;
    protected int currentlySelectedMonth;
    protected int currentlySelectedYear;
    protected String idToDelete = "";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_calendar);

        //Get list of all entries
        ArrayList<HashMap<String, String>> allEntries = gymDiaryDb.getEntries();

        //Prints out list of entries in debug
        for (HashMap<String, String> hash : allEntries) {
            Log.d("GymDiary", "id: " + hash.get("id") + " exercise_id: " + hash.get("exercise_id") + " sets: " +
            hash.get("sets") + " repetitions: " + hash.get("repetitions") + " weight: " + hash.get("weight") + "
            date_added: " + hash.get("date_added"));
        }

        //create calendar
        CalendarView calendar = (CalendarView) findViewById(R.id.calendarView);
        Calendar currentDate = Calendar.getInstance();
    }
}

```

```

//populate list with todays exercises

populateList(currentDate.get(currentDate.DAY_OF_MONTH), currentDate.get(currentDate.MONTH),
currentDate.get(currentDate.YEAR), false);

//sets up the dialog ready to be shown

final AlertDialog.Builder addEntryDialog = new AlertDialog.Builder(this);

    addEntryDialog.setTitle("Add Entry");

    addEntryDialog.setMessage("Do you want to add an entry for " +
intToDate(currentlySelectedYear, currentlySelectedMonth, currentlySelectedDay) + "?");

    addEntryDialog.setPositiveButton("Yes", new DialogInterface.OnClickListener() {

        public void onClick(DialogInterface dialog, int which) {

            Intent changeToManualEntryView = new Intent(getApplicationContext(),
AddExerciseEntryManualActivity.class);

            //sends currently selected date

            changeToManualEntryView.putExtra("from", "cal");

            changeToManualEntryView.putExtra("year", currentlySelectedYear);

            changeToManualEntryView.putExtra("month", currentlySelectedMonth);

            changeToManualEntryView.putExtra("day", currentlySelectedDay);

            startActivityForResult(changeToManualEntryView, 0);

        }

    });

//On no , do nothing

addEntryDialog.setNegativeButton("No", new DialogInterface.OnClickListener() {

    public void onClick(DialogInterface dialog, int which) {

        }

    });

```

```

//creates listener on long click(in calendar view)
calendar.setOnLongClickListener(new OnLongClickListener() {

    @Override

    public boolean onLongClick(View v) {

        //On long click display dialog included selected date
        addEntryDialog.setMessage("Do you want to add an entry for " +
intToDate(currentlySelectedYear, currentlySelectedMonth, currentlySelectedDay) + "?");

        addEntryDialog.show();

        return true;

    }

});

//whens date changes re-populate list with current selected date
calendar.setOnDateChangeListener(new OnDateChangeListener() {

    public void onSelectedDayChange(CalendarView view, int year, int month, int dayOfMonth) {

        populateList(dayOfMonth, month, year, false);

    }

});

//Sets dialog if row is deleted returns message "exercise deleted" if not "failed to delete exercise"
final AlertDialog.Builder dialog = new AlertDialog.Builder(this);

dialog.setTitle("Delete Entry");

dialog.setMessage("Are you sure you want to delete the selected entry?");

dialog.setPositiveButton("Yes", new DialogInterface.OnClickListener() {

    public void onClick(DialogInterface dialog, int which) {

        if(gymDiaryDb.removeEntry(idToDelete) > 0) {

            toast("Exercise deleted");

            populateList(currentlySelectedDay, currentlySelectedMonth,
currentlySelectedYear, true);

```

```

        } else {

            toast("Failed to delete exercise");

        }

    }

});

//If no, do nothing
dialog.setNegativeButton("No", new DialogInterface.OnClickListener() {

    public void onClick(DialogInterface dialog, int which) {

        }

    });

//If long click in calendar view
final ListView list = (ListView) findViewById(R.id.dailyExerciseList);
list.setLongClickable(true);

//Sets Listener for long click
list.setOnItemLongClickListener(new OnItemLongClickListener() {

    public boolean onItemLongClick(AdapterView<?> arg0, View arg1, int position, long id) {

        //gets ID from the array for the record that needs to be deleted , and shows in the dialog
        HashMap<String, String> tempHash = exerciseOnDayArrayList.get(position);
        idToDelete = tempHash.get(list.getItemAtPosition(position));
        dialog.show();

        return true;

    }

});
}

```

```

@SuppressLint("SimpleDateFormat")

//if the day has'nt changed and no over ride don't change, else re-populate the list
protected void populateList(int day, int month, int year, Boolean override) {
    if(day == currentlySelectedDay && month == currentlySelectedMonth && year ==
currentlySelectedYear && !override) {

        } else {
            currentlySelectedDay = day;
            currentlySelectedMonth = month;
            currentlySelectedYear = year;

            Calendar date = Calendar.getInstance();

            date.set(Calendar.YEAR, year);
            date.set(Calendar.MONTH, month);
            date.set(Calendar.DAY_OF_MONTH, day);

            String selectedDate = new SimpleDateFormat("dd-MM-yyyy").format(date.getTime());

            //debug message
            Log.d("GymDiary", selectedDate);

            ListView list = (ListView) findViewById(R.id.dailyExerciseList);

            //add entries from database added that day
            exerciseOnDayArrayList = gymDiaryDb.getEntriesOnDate(selectedDate);
            List<String> exerciseOnDay = new ArrayList<String>();

```

```

//loops through all the hash's and gets the keys
for(HashMap<String, String> hash : exerciseOnDayArrayList) {
    for(String key : hash.keySet()) {
        exerciseOnDay.add(key);
    }
}

//debug message
Log.d("GymDiary", String.valueOf(exerciseOnDay.size()));

//if records added set container if not set to null (display shows nothing)
if(exerciseOnDay.size() > 0) {
    list.setAdapter(new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1, exerciseOnDay));
} else {
    list.setAdapter(null);
}
}

}

//If 3 numbers are parsed it returns in simple date format
protected String intToDate(int year, int month, int day) {

    Calendar date = Calendar.getInstance();

    date.set(Calendar.YEAR, year);
    date.set(Calendar.MONTH, month);

```

```

        date.set(Calendar.DAY_OF_MONTH, day);

        String selectedDate = new SimpleDateFormat("dd/MM/yyyy").format(date.getTime());

        //debug messages
        Log.d("GymDiary", String.valueOf(year));
        Log.d("GymDiary", String.valueOf(month));
        Log.d("GymDiary", String.valueOf(day));
        Log.d("GymDiary", selectedDate);

        return selectedDate;
    }

    //toast method
    protected void toast(String message) {
        Toast.makeText(this, new String(message), Toast.LENGTH_SHORT).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.main, menu);
        return true;
    }
}

```


DBActivity

```
package com.brettstevens.gymdiary;

import com.brettstevens.gymdiary.R;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
import android.widget.Button;

public class DBActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_db);

        //Set button to go to exercises
        final Button goToAddExercise = (Button) findViewById(R.id.addEntryButton);
        final Button goToEditExercises = (Button) findViewById(R.id.editExercisesButton);
        final Button goToViewExercises = (Button) findViewById(R.id.viewExercisesButton);

        //sets listener for button
        goToAddExercise.setOnClickListener(
            new View.OnClickListener() {
                @Override
                public void onClick(View v) {
```

```

        Intent changeToAddExerciseView = new Intent(v.getContext(),
AddExerciseActivity.class);

        startActivityForResult(changeToAddExerciseView, 0);
    }
}

);

//Creates listner for go to edit exercises button
goToEditExercises.setOnClickListener(
    new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent changeToEditExercisesView = new Intent(v.getContext(),
EditExerciseActivity.class);

            startActivityForResult(changeToEditExercisesView, 0);
        }
    }
);

//Sets listener for go to view exercises
goToViewExercises.setOnClickListener(
    new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent changeToViewExercisesView = new Intent(v.getContext(),
ViewExercisesActivity.class);

            startActivityForResult(changeToViewExercisesView, 0);
        }
    }
);
}

```

```

@Override

public boolean onCreateOptionsMenu(Menu menu) {

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

    getMenuInflater().inflate(R.menu.main, menu);

    return true;

}

}

```

[EditExerciseActivity](#)

```

package com.brettstevens.gymdiary;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import com.brettstevens.gymdiary.R;
import com.brettstevens.gymdiary.db.Db;
import android.os.Bundle;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.util.Log;
import android.view.Menu;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.Toast;

```

```

public class EditExerciseActivity extends Activity {

    HashMap<String, String> listOfAllExercises;

    //initialize database

    Db exerciseDb = new Db(this);

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_db_edit_exercise);

        populateSpinner();

        //Set button to delete exercise

        final Button deleteExercise = (Button) findViewById(R.id.deleteSelectedItemButton);

        //Set dialog to show caution message when deleting exercise entry else if not "failed to delete"

        final AlertDialog.Builder dialog = new AlertDialog.Builder(this);

        dialog.setTitle("Caution");

        dialog.setMessage("Removing an exercise deletes all related entries");

        dialog.setPositiveButton("OK", new DialogInterface.OnClickListener() {

            public void onClick(DialogInterface dialog, int which) {

                Spinner exerciseName = (Spinner) findViewById(R.id.editExerciseSpinner);

                String exerciseNameText =
exerciseName.getSelectedItem().toString();

                if(exerciseName.getAdapter().getCount() > 2) {

                    if(exerciseDb.removeExercise(Integer.parseInt(listOfAllExercises.get(exerciseNameText))) > 0) {

```

```

        toast(exerciseNameText + " deleted");
        populateSpinner();
    } else {
        toast("Failed to delete " + exerciseNameText);
    }

    //Error message if user attempts to delete all the records(there
must be two minimum
    } else {
        toast("Spinner needs to have at least two items in it to
populate");
    }
}
});
//dialog message to cancel action
dialog.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int which) {

    }
});

// Listener to delete exercise button
deleteExercise.setOnClickListener(
    new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            dialog.show();
        }
    }
}

```

```

    );

}

protected void toast(String message) {
    Toast.makeText(this, new String(message), Toast.LENGTH_SHORT).show();
}

protected void populateSpinner() {
    final Spinner exercisesContainer = (Spinner) findViewById(R.id.editExerciseSpinner);

    listOfAllExercises = exerciseDb.getExercises();
    List<String> exerciseList = new ArrayList<String>();

    Log.d("GymDiary", "Enumerating values from keys");
    for (String key : listOfAllExercises.keySet()) {
        exerciseList.add(key);
        Log.d("GymDiary", key + " added to list");
    }

    Log.d("GymDiary", "Creating adapter");
    if(exerciseList.size() == 1) {

    }

    ArrayAdapter<String> spinnerAdapter = new ArrayAdapter<String>(this,
    android.R.layout.simple_spinner_item, exerciseList);

    spinnerAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

```

```
Log.d("GymDiary", spinnerAdapter.getItem(1).toString());

Log.d("GymDiary", "Adding to spinner");
exercisesContainer.setAdapter(spinnerAdapter);
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}
}
```

HelpActivity

```
package com.brettstevens.gymdiary;
```

```
import com.brettstevens.gymdiary.R;
```

```
import android.os.Bundle;
```

```
import android.app.Activity;
```

```
import android.view.Menu;
```

```
public class HelpActivity extends Activity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_help);
```

```
    }
```

```
    @Override
```

```
    public boolean onCreateOptionsMenu(Menu menu) {
```

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

```
        getMenuInflater().inflate(R.menu.main, menu);
```

```
        return true;
```

```
    }
```

```
}
```


MainActivity

```
package com.brettstevens.gymdiary;

import com.brettstevens.gymdiary.R;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
import android.widget.Button;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        //Set button to go to scan NFC tag page
        final Button goToScanTagButton = (Button) findViewById(R.id.goToScanTagButton);

        //Add click listener to button
        goToScanTagButton.setOnClickListener(
            new View.OnClickListener() {
                @Override

                public void onClick(View v) {
                    //on click generate new intent, starts NFC activity
```

```

                                Intent changeToScanTagView = new Intent(v.getContext(),
NFCActivity.class);
                                startActivityResult(changeToScanTagView, 0);
                                }
                                }

);

//Set button to go to database page
final Button goToDbButton = (Button) findViewById(R.id.goToDbButton);
//Add click listener to button
goToDbButton.setOnClickListener(
    new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            //on click generate new intent, starts Database activity
            Intent changeToDbView = new Intent(v.getContext(),
DBActivity.class);
            startActivityResult(changeToDbView, 0);
            }
        }
);

//Set button to go to manual entry page
final Button goToManualEntryButton = (Button) findViewById(R.id.goToManualEntryButton);

goToManualEntryButton.setOnClickListener(
    new View.OnClickListener() {
        @Override

```

```

        public void onClick(View v) {
            //on click generate new intent and set from tag, starts manual
exercise entry activity

            Intent changeToManualEntryView = new Intent(v.getContext(),
AddExerciseEntryManualActivity.class);

            changeToManualEntryView.putExtra("from", "home");
            startActivityForResult(changeToManualEntryView, 0);
        }
    }
};

```

```

//Set button to go to calendar page
final Button goToCalendarButton = (Button) findViewById(R.id.goToCalendarButton);

goToCalendarButton.setOnClickListener(
    new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent changeToCalendarView = new Intent(v.getContext(),
CalendarActivity.class);

            startActivityForResult(changeToCalendarView, 0);
        }
    }
);

```

```

//Set button to go to stopwatch page
final Button goToStopWatchButton = (Button) findViewById(R.id.goToStopWatchButton);

//adds listner to help page button
goToStopWatchButton.setOnClickListener(

```

```

        new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //on click generate new intent stop watch activity
                Intent changeToStopWatchView = new Intent(v.getContext(),
StopWatchActivity.class);

                startActivityForResult(changeToStopWatchView, 0);
            }
        }

    );

    //Set button to go to help page
    final Button goToHelpButton = (Button) findViewById(R.id.goToHelpButton);
    //adds listener to help page button
    goToHelpButton.setOnClickListener(
        new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //on click generate new intent, starts activity for help view
                Intent changeToHelpView = new Intent(v.getContext(),
HelpActivity.class);

                startActivityForResult(changeToHelpView, 0);
            }
        }
    );
}

```

```
@Override  
  
public boolean onCreateOptionsMenu(Menu menu) {  
    // Inflate the menu; this adds items to the action bar if it is present.  
    getMenuInflater().inflate(R.menu.main, menu);  
    return true;  
}  
}
```

NFCActivity

```
package com.brettstevens.gymdiary;

import java.util.HashMap;
import java.util.List;

import com.brettstevens.gymdiary.R;
import com.brettstevens.gymdiary.db.Db;
import com.brettstevens.gymdiary.nfc.*;
import android.nfc.NfcAdapter;
import android.os.Bundle;
import android.app.Activity;
import android.app.PendingIntent;
import android.content.Intent;
import android.util.Log;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class NFCActivity extends Activity {
    //Set up variable for use later
    private TextView textView;
    private NfcAdapter NFCAdapter;
    private PendingIntent NFCTagPendingIntent;
    private Db gymDiaryDb = new Db(this);
    private int tagId = -1;
    private String name = "";
```

```

private String dateAdded = "";

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_nfc);

    //Set textView to our label
    textView = (TextView) findViewById(R.id.checkNfcAdapterTextMain);

    //Get and initialise NFC adapter from phone
    NFCAdapter = NfcAdapter.getDefaultAdapter(this);

    //Check if we have got an adapter and set text to indicate
    if(NFCAdapter == null) {
        textView.setText("Unable to enable NFC adapter");
    }

    // Create and intent for tag data
    NFCTagPendingIntent = PendingIntent.getActivity(this, 0, new Intent(this,
    getClass()).addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP), 0);

    final Button goToAddExerciseEntry = (Button) findViewById(R.id.addActivityFromTagButton);

    goToAddExerciseEntry.setOnClickListener(
        new View.OnClickListener() {
            @Override
            public void onClick(View v) {

```

```

        Intent changeToAddExerciseEntryView = new
Intent(v.getContext(), AddExerciseEntryActivity.class);

        changeToAddExerciseEntryView.putExtra("tagId", tagId);
        changeToAddExerciseEntryView.putExtra("name", name);
        changeToAddExerciseEntryView.putExtra("dateAdded",
dateAdded);

        startActivityForResult(changeToAddExerciseEntryView, 0);

    }

}

);
}

@Override
public void onNewIntent(Intent intent) {

    //Initialise reader and pass it the intent to parse
    NFCReader reader = new NFCReader();
    HashMap<String, String> result = reader.getTag(intent);
    Boolean showButton = false;

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

    if(result.get("payload") == "-1") {
        textView.setText("Empty or unrecognisable tag scanned -- Please try again");
        addActivityFromTagButton.setVisibility(View.INVISIBLE);
    } else {
        String[] values = result.get("payload").split(";");
        if(values.length == 2) {
            textView.setText(values[1]);

```



```

        List<String> checkForExercise =
gymDiaryDb.checkForTagId(Integer.valueOf(values[0]));

        Log.d("GymDiary-Size", String.valueOf(checkForExercise.size()));

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

            toast("New exercise scanned, adding " + values[1] + " to database");

            if(gymDiaryDb.addExercise(values[1], Integer.valueOf(values[0]))) {

                toast("Successfully added " + values[1] + " to database");

                showButton = true;

                tagId = Integer.valueOf(values[0]);

                name = values[1];

                dateAdded = result.get("dateRead");

            } else {

                toast("Something went wrong whilst adding " + values[1] + " to
the database");

            }

        } else if (checkForExercise.size() == 1) {

            showButton = true;

            tagId = Integer.valueOf(values[0]);

            name = values[1];

            dateAdded = result.get("dateRead");

        }

    } else {

        showButton = false;

        textView.setText(result.get("payload"));

        toast("Malformed Gym Diary tag detected.\nPayload: " + result.get("payload"));

    }

}

```

```

        if(showButton) {
            addActivityFromTagButton.setVisibility(View.VISIBLE);
        } else {
            addActivityFromTagButton.setVisibility(View.INVISIBLE);
        }
    }

    protected void toast(String message) {
        Toast.makeText(this, new String(message), Toast.LENGTH_SHORT).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.main, menu);
        return true;
    }

    @Override
    public void onResume() {
        super.onResume();

        //If we have NFC capability, enable foreground dispatch with the pending intent
        if (NFCAdapter != null) {
            NFCAdapter.enableForegroundDispatch(this, NFCTagPendingIntent, null, null);
        }
    }

```

```
}

@Override
public void onPause() {
    super.onPause();

    //If we have NFC capability, disable foreground dispatch
    if (NFCAdapter != null) {
        NFCAdapter.disableForegroundDispatch(this);
    }
}
}
```

StopWatchActivity

```
package com.brettstevens.gymdiary;
```

```
import android.app.Activity;
```

```
import android.content.Intent;
```

```
import android.graphics.Color;
```

```
import android.os.Bundle;
```

```
import android.os.SystemClock;
```

```
import android.view.Menu;
```

```
import android.view.View;
```

```
import android.widget.Button;
```

```
import android.widget.Chronometer;
```

```
public class StopWatchActivity extends Activity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_stopwatch);
```

```
        //sets chronometer to a variable
```

```
        final Chronometer stopWatch = (Chronometer) findViewById(R.id.stopWatchChronometer);
```

```
        //Set start/stop button
```

```
        final Button startStopButton = (Button) findViewById(R.id.stopStartWatchButton);
```

```
        //Sets listener for start / stop button
```

```
        startStopButton.setOnClickListener(
```

```
            new View.OnClickListener() {
```

```

@Override
public void onClick(View v) {

    String button = startStopButton.getText().toString();

    if(button.equals("Start")) {
        stopWatch.setBase(SystemClock.elapsedRealtime());
        stopWatch.start();
        //On button press changes colour of button to red

startStopButton.setBackgroundColor(Color.parseColor("#bd362f"));
        startStopButton.setText("Stop");
    } else {
        stopWatch.stop();
        //On button press changes colour of button to green

startStopButton.setBackgroundColor(Color.parseColor("#51a351"));
        startStopButton.setText("Start");
    }
}

}

);

//Set button to go to manual entry
final Button resetButton = (Button) findViewById(R.id.resetStopWatchButton);

//Sets listener for reset button
resetButton.setOnClickListener(

```

```
        new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                stopWatch.setBase(SystemClock.elapsedRealtime());  
            }  
        }  
    );  
  
}  
  
@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
    // Inflate the menu; this adds items to the action bar if it is present.  
    getMenuInflater().inflate(R.menu.main, menu);  
    return true;  
}  
}
```

ViewExercisesActivity

```
package com.brettstevens.gymdiary;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

import com.brettstevens.gymdiary.R;
import com.brettstevens.gymdiary.db.Db;

import android.os.Bundle;
import android.app.Activity;
import android.util.Log;
import android.view.Menu;
import android.widget.AdapterView;
import android.widget.ListView;

public class ViewExercisesActivity extends Activity {

    //Initializes database
    protected Db gymDiaryDb = new Db(this);

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_db_view_exercises);

        //Selects all exercises and returnsd hash with name and id
```

```

HashMap<String, String> listOfAllExercises = gymDiaryDb.getExercises();
List<String> exerciseList = new ArrayList<String>();

//debug message
Log.d("GymDiary", "Enumerating values from keys");

//loop through to get key
for (String key : listOfAllExercises.keySet()) {
    exerciseList.add(key);
    Log.d("GymDiary", key + " added to list");
}

ListView list = (ListView) findViewById(R.id.exercisesInDatabaseList);

    ArrayAdapter<String> listAdapter = new ArrayAdapter<String>(this,
    android.R.layout.simple_list_item_1, exerciseList);

//debug message
Log.d("GymDiary", "Adding to list");
list.setAdapter(listAdapter);
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

```



```
}
```

Db

```
package com.brettstevens.gymdiary.db;
```

```
import java.util.ArrayList;
```

```
import java.util.HashMap;
```

```
import java.util.List;
```

```
import android.content.ContentValues;
```

```
import android.content.Context;
```

```
import android.database.Cursor;
```

```
import android.database.sqlite.SQLiteDatabase;
```

```
import android.database.sqlite.SQLiteOpenHelper;
```

```
import android.util.Log;
```

```
public class Db extends SQLiteOpenHelper {
```

```
    private static final String GymDiary = null;
```

```
    public Db(Context application) {
```

```
        super(application, "gymdiary.db", null, 1);
```

```
        Log.d(GymDiary, "Created");
```

```
    }
```

```
    @Override
```

```
    public void onCreate(SQLiteDatabase gymDiaryDatabase) {
```

```
        String query;
```

```
        query = "CREATE TABLE \"exercises\" (\"id\" INTEGER PRIMARY KEY AUTOINCREMENT  
NOT NULL UNIQUE , \"name\" TEXT NOT NULL UNIQUE , \"tag_id\" INTEGER UNIQUE )";
```

```
gymDiaryDatabase.execSQL(query);
```

```
Log.d(GymDiary,"Exercises table added");
```

```
query = "CREATE TABLE \"entries\" (\n\"id\" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL\nUNIQUE ,\n\"exercise_id\" INTEGER NOT NULL ,\n\"sets\" INTEGER NOT NULL ,\n\"repetitions\" INTEGER\nNOT NULL ,\n\"weight\" DOUBLE NOT NULL ,\n\"date_added\" DATETIME NOT NULL );
```

```
gymDiaryDatabase.execSQL(query);
```

```
ContentValues values = new ContentValues();
```

```
values.put("name", "Bench press");
```

```
values.put("tag_id", "1");
```

```
gymDiaryDatabase.insert("exercises", null, values);
```

```
values = new ContentValues();
```

```
values.put("name", "Deadlift");
```

```
values.put("tag_id", "2");
```

```
gymDiaryDatabase.insert("exercises", null, values);
```

```
Log.d(GymDiary,"Entries table added");
```

```
}
```

```
@Override
```

```
public void onUpgrade(SQLiteDatabase gymDiaryDatabase, int old, int current) {
```

```
    String query;
```

```
    query = "DROP TABLE IF EXISTS exercises";
```

```
    gymDiaryDatabase.execSQL(query);
```

```
    query = "DROP TABLE IF EXISTS entries";
```

```
    gymDiaryDatabase.execSQL(query);
```

```
onCreate(gymDiaryDatabase);
```

```
}
```

```
public Boolean addExercise(String name, int tagId) {
    SQLiteDatabase database = this.getWritableDatabase();
    ContentValues values = new ContentValues();

    Boolean success = false;

    values.put("name", name);
    if(tagId != -1) {
        values.put("tag_id", tagId);
    } else {
        values.put("tag_id", (String) null);
    }

    if(database.insert("exercises", null, values) != -1) {
        success = true;
    }
    database.close();
    Log.d("GymDiary", "Added " + name);

    return success;
}
```

```
public Boolean addEntry(String exerciseId, String sets, String reps, String weight, String
dateAdded) {
    SQLiteDatabase database = this.getWritableDatabase();
    ContentValues values = new ContentValues();
```

```

        Boolean success = false;

        values.put("exercise_id", exerciseId);
        values.put("sets", sets);
        values.put("repetitions", reps);
        values.put("weight", weight);
        values.put("date_added", dateAdded);

        if(database.insert("entries", null, values) != -1) {
            success = true;
        }
        database.close();
        Log.d("GymDiary", "Added entry");

        return success;
    }

    public HashMap<String, String> getExercises() {
        Log.d("GymDiary", "Getting all exercises");
        HashMap<String, String> listOfAllExercises = new HashMap<String, String>();
        String query = "SELECT * FROM exercises";
        SQLiteDatabase database = this.getWritableDatabase();
        Cursor cursor = database.rawQuery(query, null);

        if (cursor.moveToFirst()) {
            do {
                listOfAllExercises.put(cursor.getString(1), cursor.getString(0));
            } while (cursor.moveToNext());
        }
    }

```

```

        Log.d("GymDiary", "Returning all exercises");
        return listOfAllExercises;
    }

    public HashMap<String, String> getExercise(int id) {
        HashMap<String, String> exercise = new HashMap<String, String>();
        SQLiteDatabase database = this.getReadableDatabase();
        String selectQuery = "SELECT * FROM exercises where id='" + Integer.toString(id) + "'";
        Cursor cursor = database.rawQuery(selectQuery, null);
        if (cursor.moveToFirst()) {
            do {
                //HashMap<String, String> map = new HashMap<String,
String>();

                exercise.put("id", cursor.getString(0));
                exercise.put("name", cursor.getString(1));
                exercise.put("tag_id", cursor.getString(2));
                //wordList.add(map);
            } while (cursor.moveToNext());
        }

        return exercise;
    }

    public List<String> checkForTagId(int id) {
        List<String> exercise = new ArrayList<String>();
        SQLiteDatabase database = this.getReadableDatabase();
        String selectQuery = "SELECT tag_id FROM exercises where tag_id='" +
Integer.toString(id) + "'";
        Cursor cursor = database.rawQuery(selectQuery, null);

```

```

        if (cursor.moveToFirst()) {
    do {
        exercise.add(cursor.getString(0));
    } while (cursor.moveToNext());
    }

    return exercise;
}

public List<String> getIdFromTagId(int id) {
    List<String> exercise = new ArrayList<String>();
    SQLiteDatabase database = this.getReadableDatabase();
    String selectQuery = "SELECT id FROM exercises where tag_id='" + Integer.toString(id) +
    """;

    Cursor cursor = database.rawQuery(selectQuery, null);
    if (cursor.moveToFirst()) {
    do {
        exercise.add(cursor.getString(0));
    } while (cursor.moveToNext());
    }

    return exercise;
}

public ArrayList<HashMap<String, String>> getEntriesOnDate(String date) {
    ArrayList<HashMap<String, String>> exercise = new ArrayList<HashMap<String,
String>>();

    SQLiteDatabase database = this.getReadableDatabase();

    String selectQuery = "SELECT exercises.name, entries.sets, entries.repetitions,
entries.weight, entries.id FROM entries, exercises WHERE entries.date_added BETWEEN '" + date + "
00:00:00.00' AND '" + date + " 23:59:59.999' AND entries.exercise_id = exercises.id";

    Cursor cursor = database.rawQuery(selectQuery, null);

```

```

        if (cursor.moveToFirst()) {
            do {
                HashMap<String, String> hash = new HashMap<String, String>();

                hash.put(cursor.getString(0) + "\n\tSets: " + cursor.getString(1) + "\n\tReps: " +
cursor.getString(2) + "\n\tWeight: " + cursor.getString(3) + "Kg", cursor.getString(4));

                exercise.add(hash);
            } while (cursor.moveToNext());
        }

        return exercise;
    }

    public ArrayList<HashMap<String, String>> getEntries() {
        Log.d("GymDiary", "Getting all entries");

        ArrayList<HashMap<String, String>> listOfAllEntries = new ArrayList<HashMap<String,
String>>();

        String query = "SELECT * FROM entries";
        SQLiteDatabase database = this.getWritableDatabase();
        Cursor cursor = database.rawQuery(query, null);

        if (cursor.moveToFirst()) {
            do {
                HashMap<String, String> hash = new HashMap<String, String>();

                hash.put("id", cursor.getString(0));
                hash.put("exercise_id", cursor.getString(1));
                hash.put("sets", cursor.getString(2));
                hash.put("repetitions", cursor.getString(3));
                hash.put("weight", cursor.getString(4));
                hash.put("date_added", cursor.getString(5));

                listOfAllEntries.add(hash);
            } while (cursor.moveToNext());
        }
    }
}

```

```
        } while (cursor.moveToNext());
    }

    Log.d("GymDiary", "Returning all exercises");
    return listOfAllEntries;
}

public int removeExercise(int id) {
    SQLiteDatabase database = this.getWritableDatabase();
    String query = "DELETE FROM entries WHERE exercise_id = '" + Integer.toString(id) + "'";
    database.rawQuery(query, null);
    return database.delete("exercises", "id = " + Integer.toString(id), null);
}

public int removeEntry(String id) {
    SQLiteDatabase database = this.getWritableDatabase();
    return database.delete("entries", "id = " + id, null);
}
}
```


NFCReader

```

package com.brettstevens.gymdiary.nfc;

import java.text.SimpleDateFormat;
import java.util.Arrays;
import java.util.Date;
import java.util.HashMap;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.nfc.NdefMessage;
import android.nfc.NdefRecord;
import android.nfc.NfcAdapter;
import android.os.Parcelabele;
import android.util.Log;

public class NFCReader {
    @SuppressLint("SimpleDateFormat")
    public HashMap<String, String> getTag(Intent intent) {

        HashMap<String, String> result = new HashMap<String, String>();

        //Initialise string
        String tagText = "";

        //Get array of data to parse
        Parcelable[] data =
intent.getParcelableArrayExtra(NfcAdapter.EXTRA_NDEF_MESSAGES);

        //Check if we have data payload, if not it is an empty but formatted
tag
        if (data != null) {
            try {
                //Loop through data and get stored records
                for (int i = 0; i < data.length; i++) {
                    NdefRecord [] NDEFRecords =
((NdefMessage)data[i]).getRecords();

                    //Loop through record and get
                    for (int j = 0; j < NDEFRecords.length; j++) {
                        if (NDEFRecords[j].getTnf() ==
NdefRecord.TNF_WELL_KNOWN && Arrays.equals(NDEFRecords[j].getType(),
NdefRecord.RTD_TEXT)) {
                            byte[] tagPayload =
NDEFRecords[j].getPayload();

                                /* Build string, stripping of the language
identifier at the start
                                *
                                * tagPayload[0] & 0777 gets the length of the
country code identifier from the control byte
                                * tagPayload[0] & 0200 gets the bit identifying
whether it is stored as UTF-8 or UTF-16 from the control byte
                                *

```

```

        * new String(the whole payload, start string
from country code identifier +1, finish at length - country code identifier
length - 1, set encoding to UTF-8 or UTF-16 depending on control byte
        */
        tagText = new String(tagPayload, (tagPayload[0] &
0077) + 1, tagPayload.length - (tagPayload[0] & 0077) - 1, ((tagPayload[0] &
0200) == 0) ? "UTF-8" : "UTF-16");
    }
}
} catch (Exception e) {
    Log.e("TagDispatch", e.toString());
}
} else {
    tagText = "";
}

result.put("payload", (tagText.isEmpty() || tagText == null) ? "-1" :
tagText);
result.put("dateRead", new SimpleDateFormat("dd-MM-yyyy
HH:mm:ss").format(new Date()));

return result;
}
}

```

activity_calendar.xml

```

<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/GridLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="0dp"
    android:background="#ffffff"
    android:columnCount="1"
    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"
    tools:context=".MainActivity" >

    <Space />

    <TableLayout
        android:id="@+id/tableGrid"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <CalendarView
            android:id="@+id/calendarView"
            android:layout_width="match_parent"
            android:layout_height="240dp"
            android:layout_column="0"
            android:layout_gravity="left|top"
            android:layout_row="0" />

        <ListView
            android:id="@+id/dailyExerciseList"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_gravity="fill_horizontal" >
        </ListView>
    </TableLayout>

</GridLayout>

```

activity_db_add_entry_manual.xml

```

<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/TableLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="0dp"
    android:background="#ffffff"
    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=".MainActivity" >

    <TextView
        android:id="@+id/addEntryManualExerciseLabel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Exercise"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <Spinner
        android:id="@+id/addEntryManualExerciseSpinner"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <TextView
        android:id="@+id/addEntryManualSetsLabel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Sets"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <Spinner
        android:id="@+id/addEntryManualSetsSpinner"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <TextView
        android:id="@+id/addEntryManualRepetitionsLabel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Repetitions"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <Spinner
        android:id="@+id/addEntryManualRepsSpinner"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <TextView
        android:id="@+id/addEntryManualWeightLabel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Weight (Kg)"
        android:textAppearance="?android:attr/textAppearanceMedium" />

```

```
<Spinner
    android:id="@+id/addEntryManualWeightSpinner"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

<Button
    android:id="@+id/addEntryManualButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom"
    android:layout_marginTop="10dp"
    android:background="#51a351"
    android:text="Add Entry"
    android:textColor="#ffffff" />

</TableLayout>
```

activity_db_add_entry.xml

```

<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/TableLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="0dp"
    android:background="#ffffff"
    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=".MainActivity" >

    <TableRow
        android:id="@+id/tableRow1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginBottom="10dp" >

        <TextView
            android:id="@+id/addEntryNameLabel"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_weight="1"
            android:text="Name"
            android:textAppearance="?android:attr/textAppearanceLarge" />

    </TableRow>

    <TextView
        android:id="@+id/addEntrySetsLabel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Sets"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <Spinner
        android:id="@+id/addEntrySetsSpinner"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <TextView
        android:id="@+id/addEntryRepetitionsLabel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Repetitions"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <Spinner
        android:id="@+id/addEntryRepsSpinner"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

```

```

<TextView
    android:id="@+id/addEntryWeightLabel"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Weight (Kg)"
    android:textAppearance="?android:attr/textAppearanceMedium" />

<Spinner
    android:id="@+id/addEntryWeightSpinner"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

<Button
    android:id="@+id/addEntryButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom"
    android:layout_marginTop="10dp"
    android:background="#51a351"
    android:text="Add Entry"
    android:textColor="#ffffff" />

</TableLayout>

```

activity_db_add_exercise.xml

```

<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/TableLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="0dp"
    android:background="#ffffff"
    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=".MainActivity" >

    <TextView
        android:id="@+id/exerciseNameLabel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="left|top"
        android:text="Exercise Name"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <EditText
        android:id="@+id/exerciseNameTextField"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="left"
        android:ems="10" />

    <Button
        android:id="@+id/addEntryButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom"
        android:layout_marginTop="10dp"
        android:background="#51a351"
        android:text="Add Exercise"
        android:textColor="#ffffff" />

</TableLayout>

```


activity_db_edit_exercise.xml

```

<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/TableLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="0dp"
    android:background="#ffffff"
    android:orientation="horizontal"
    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=".MainActivity" >

    <Spinner
        android:id="@+id/editExerciseSpinner"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="left|top" />

    <Button
        android:id="@+id/deleteSelectedItemButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:background="#bd362f"
        android:text="Delete Record"
        android:textColor="#ffffff" />

</TableLayout>

```

activity_db_view_exercises.xml

```
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/GridLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="0dp"
    android:background="#ffffff"
    android:columnCount="1"
    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"
    tools:context=".MainActivity" >

    <ListView
        android:id="@+id/exercisesInDatabaseList"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_column="0"
        android:layout_gravity="left|top"
        android:layout_row="1" >
    </ListView>

    <TextView
        android:id="@+id/exercisesInDatabaseListLabel"
        android:layout_column="0"
        android:layout_gravity="left|top"
        android:layout_row="0"
        android:text="Exercises"
        android:textAppearance="?android:attr/textAppearanceLarge" />

</GridLayout>
```

activity_db.xml

```

<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/TableLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="0dp"
    android:background="#ffffff"
    android:orientation="horizontal"
    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=".MainActivity" >

    <Button
        android:id="@+id/addEntryButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="left|top"
        android:layout_marginBottom="5dp"
        android:background="#51a351"
        android:text="Add Exercise"
        android:textColor="#ffffff" />

    <Button
        android:id="@+id/viewExercisesButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="5dp"
        android:background="#0044cc"
        android:text="View Exercises"
        android:textColor="#ffffff" />

    <Button
        android:id="@+id/editExercisesButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="left|top"
        android:background="#bd362f"
        android:text="Delete Exercise"
        android:textColor="#ffffff" />

</TableLayout>

```

activity_help.xml

```
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/GridLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="0dp"
    android:background="#ffffff"
    android:columnCount="1"
    android:orientation="horizontal"
    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=".MainActivity" >

</GridLayout>
```

activity_main.xml

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/RelativeLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="5dp"
    android:background="#ffffff"
    android:columnCount="2"
    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"
    tools:context=".MainActivity" >

    <Space
        android:id="@+id/Space1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="fill_vertical" />

    <Space
        android:id="@+id/Space1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_column="1"
        android:layout_gravity="fill"
        android:layout_row="6" />

    <TableLayout
        android:id="@+id/tableGridLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true" >

        <Button
            android:id="@+id/goToScanTagButton"
            android:layout_marginBottom="5dp"
            android:background="#0044cc"
            android:text="Scan Tag"
            android:textColor="#ffffff" />

        <Button
            android:id="@+id/goToManualEntryButton"
            android:layout_marginBottom="5dp"
            android:background="#0044cc"
            android:text="Manual Entry"
            android:textColor="#ffffff" />

        <Button
            android:id="@+id/goToDbButton"
            android:layout_column="0"
            android:layout_gravity="fill_horizontal"
            android:layout_marginBottom="5dp"

```

```

        android:background="#0044cc"
        android:text="Database"
        android:textColor="#ffffff" />

<Button
    android:id="@+id/goToCalendarButton"
    android:layout_column="0"
    android:layout_gravity="fill_horizontal"
    android:layout_marginBottom="5dp"
    android:background="#0044cc"
    android:text="Calendar"
    android:textColor="#ffffff" />

<Button
    android:id="@+id/goToStopWatchButton"
    android:layout_column="0"
    android:layout_gravity="fill_horizontal"
    android:layout_marginBottom="5dp"
    android:background="#0044cc"
    android:text="Stop Watch"
    android:textColor="#ffffff" />

<Button
    android:id="@+id/goToHelpButton"
    android:layout_column="0"
    android:layout_gravity="fill_horizontal"
    android:background="#0044cc"
    android:text="Help"
    android:textColor="#ffffff" />
</TableLayout>

<ImageView
    android:id="@+id/mainActivityImage"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="5dp"
    android:src="@drawable/gymdiarylogo" />

</RelativeLayout>

```

activity_nfc.xml

```

<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/GridLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="0dp"
    android:background="#ffffff"
    android:columnCount="1"
    android:orientation="horizontal"
    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=".NFCActivity" >

    <ImageView
        android:id="@+id/scanNFCTagImageView"
        android:layout_width="match_parent"
        android:layout_height="330dp"
        android:layout_column="0"
        android:layout_gravity="left|top"
        android:layout_marginTop="30dp"
        android:layout_row="0"
        android:src="@drawable/phonescan" />

    <TextView
        android:id="@+id/checkNfcAdapterTextMain"
        android:layout_column="0"
        android:layout_gravity="center_horizontal|bottom"
        android:layout_marginBottom="110dp"
        android:layout_row="0"
        android:text="Please Scan Tag"
        android:textAlignment="center"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textSize="25sp"
        android:visibility="visible" />

    <Button
        android:id="@+id/addActivityFromTagButton"
        android:layout_width="fill_parent"
        android:layout_column="0"
        android:layout_gravity="left|bottom"
        android:layout_marginBottom="10dp"
        android:layout_row="0"
        android:background="#51a351"
        android:text="Add Entry"
        android:textAllCaps="true"
        android:textColor="#ffffff"
        android:visibility="invisible" />

</GridLayout>

```

activity_stopwatch.xml

```

<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/TableLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="0dp"
    android:background="#ffffff"
    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:textAlignment="center"
    tools:context=".MainActivity" >

    <RelativeLayout
        android:id="@+id/relativeStopWatchContainer"
        android:layout_width="fill_parent"
        android:layout_height="217dp"
        android:layout_marginTop="25dp" >

        <ImageView
            android:id="@+id/stopWatchImage"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_alignParentLeft="true"
            android:layout_alignParentRight="true"
            android:layout_alignParentTop="true"
            android:scaleType="fitCenter"
            android:src="@drawable/stopwatch" />

        <Chronometer
            android:id="@+id/stopWatchChronometer"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="95dp"
            android:gravity="center"
            android:text="StopWatch"
            android:textAlignment="center"
            android:textSize="50sp" />
    </RelativeLayout>

    <Button
        android:id="@+id/stopStartWatchButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="left|center_vertical"
        android:layout_marginBottom="5dp"
        android:layout_marginTop="85dp"
        android:background="#51a351"
        android:text="Start"
        android:textColor="#ffffff" />

    <Button
        android:id="@+id/resetStopWatchButton"

```



```
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:background="#0044cc"  
        android:text="Reset"  
        android:textColor="#ffffff" />  
</TableLayout>
```