****

**Computer & Electronic Engineering**

**Final Year Project "Music Host Interface"**

**Sprint 6: Week 3 Log**

**Thomas Flynn**

**Project Supervisor: Brian O'Shea**

**07/03/16 - 13/03/16**

**Entry 1: 10/03/16**

Today I accomplished a major goal in my project. Instead of having my custom client-server protocol based on reading the first char in the inputstream and then determining what to do next with the package. I instead decided to use read and writeInt() in order to determine operations.

A simple switch statement of 1-5 cases will be given to both the host and client. This will essentially be how the two systems interface.

Either the client or the host will send an intention. This specific intention will first write an int to the outputstream. The receiver will readInt() from inputstream and depending on the intention integer. The receiver will then determine what to do next.

The implications for thread safety with this approach is probably... erh...not good, especially considering how the single connection thread on the server side will have to do a lot of work and also dip it's hand into many pies in the Model (ie. unsynchronised helper functions, variables being accessed without semaphore locks and race conditions) .

As of this moment in time my FX GUI can initialize a list of songs from the remote SQL database into the selection list and can then add a specific song to the song queuelist from the selection list.

With this initialization done followed by clicking "start server", the Android client will receive a JSON array of songs that originated from the remote cloud SQL database. The client gets a listView of songs after aysyncTask() finishes processing the song Beans from the JSON song array. From there the Android user can select a song and send that clickable song to the FX GUI screen as a String. This has also been all laid out with the foundation of read/writeInt that my improvised communication will use.

**Entry 2: 11/03/16**

Today I worked on the login screen for my FX GUI. I managed to get the login success/error messages working. Now upon logging in, the unique ID of the user is stored in the model. The ID that is stored will be used in SQL statements.

**Task completed:**

ReadInt working with Android and writeInt working for FX host.

Server sends JSON array of songs upon connecting to client.

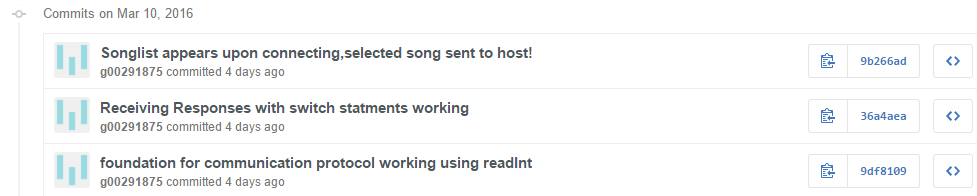
Android's song Listview clickable is sent to host.

Login works with database on FX GUI.

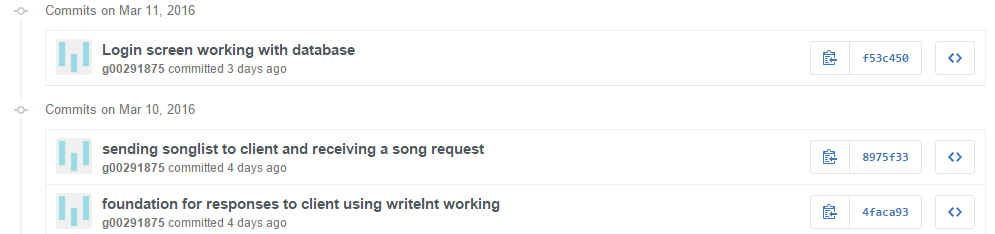
Added songs to the SQL database for the second user for testing later.

**Github Repositories:**

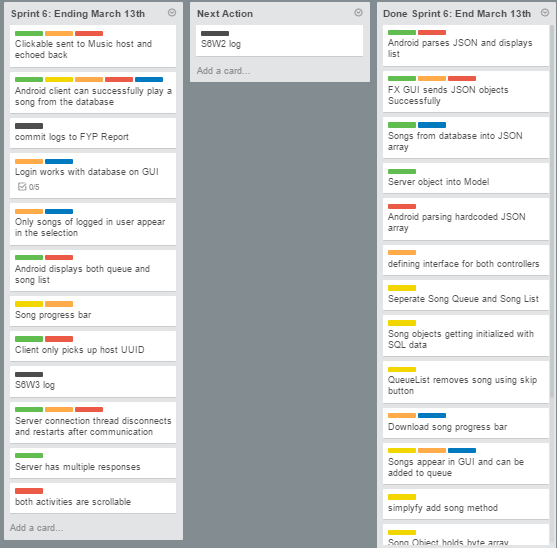
**FYP-Android:**

****

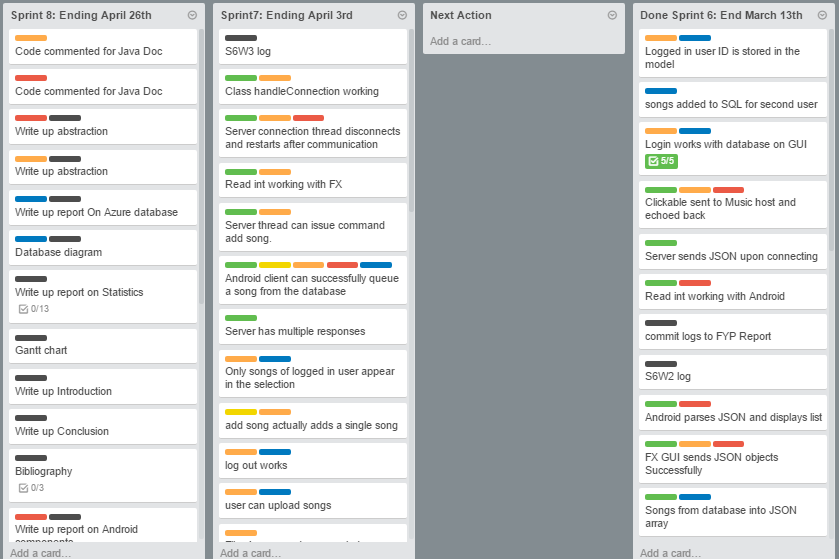
**FYP-GUI:**

****

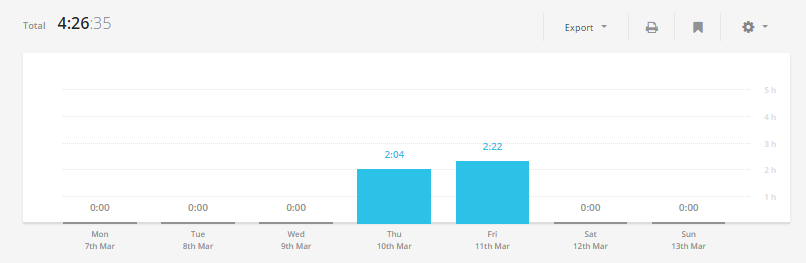
**Board at the start of the week:**

****

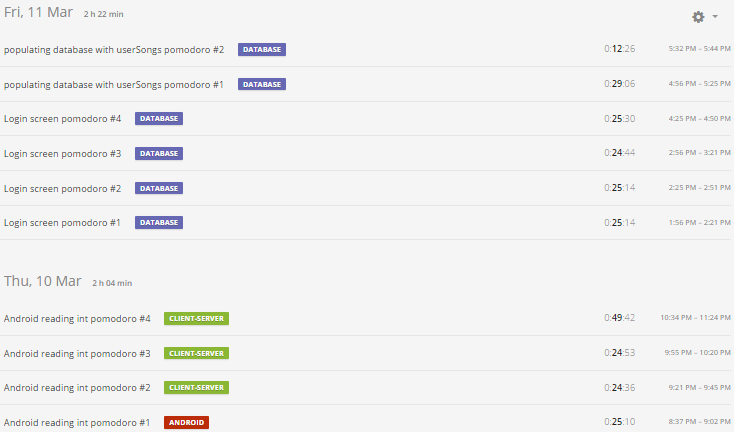
**Board at the end of the week:**

****

**Weekly time Log bar chart:**

****

**Weekly Time Log:**

****

**Weekly log Pie Charts:**

****