****

# Computer & Electronic Engineering

# Final Year Project "Music Host Interface"

# Sprint 7: Week 1 Log

# Thomas Flynn

# Project Supervisor: Brian O'Shea

# 14/03/16 - 20/03/16

# 1 Log Entries

## 1.1.0 Entry 1: 18/03/16

Today whilst still being mildly intoxicated I researched java script basics and added a WebView and WebEngine to the DJ screen. The web engine object will perform my "downloadYoutubeSongRequestScript.JS". It is my hope to remove the WebView entirely from the DJ screen and leave only the WebEngine as a behind the scenes interface for running my java script. I might reconsider a permanent webView at a later date. One that will display the top 40 songs in the charts via a Java script.

Ideally the downloadYoutubeMp3 JS script will run within (or just before) the construction of the Youtube song object and initialize either the "songdatabyte array" field or directly feed the downloaded youtube mp3 file URI into the constructor of the MediaPlayer() which is another field within the song class.

## 1.1.1 Clarification:

The field songByteArray within the Song object get's initialized when it ordinarily goes from the "selectionList" to the "QueueList" via the add button. songByteArray is usually initialized with binary data from the SQL server when it transitions from selectionList to QueueList. This happens because when it's in the selection list, the song object just has it's ID, String songname, and String artistname in order to keep the object lightweight.

The field mediaPlayerSong is an object the song Class constructs (ideally) when it's just about to be played. Where the song byte[] data progresses from the "bytearray stage" to the "file I/O stage". And then (ideally) the mp3 file used to construct the MediaPlayer is deleted when the song is over. The reason for this is because MediaPlayer will only take file URI's into the constructor, therefore an mp3 file will be created for every song, so the files need to be deleted and IO streams need to be closed in a finally{} to prevent resource leaks.

1.1.2 Random idea:

There needs to be a clearly defined interface for the Song object for when it's in the queuelist or the selection list. (song object behaves differently when it's in the different queues). A song object in selection list should not be allowed to implement "play me" because it's not a "QueueSong object" it's a lightweight "SelectionSong object". I'll probably pass the selectionSong object into the constructor of the QueueSong object when I'm creating them (via the add song to queue button) in order to copy the ID,songname and artistname fields over quickly.

This might is probably irrelevant dribble and will not be implemented at all. Mainly just tracking my thoughts at this moment in time.

## 1.2.1 Entry 19: 28/02/16

Idea to use ANT to run javascripts.

Or ant runs a static void main that runs JS for "youtube download" , "security scan", and creates song object. At this moment in time I don't know even know how one runs a java script, just to give some perspective on the hopeful eventual product I will demonstrate.

came accross this

JSObject jdoc = (JSObject) webEngine.getDocument();

#### Mapping JavaScript values to Java objects

If the result is a JavaScript object, it is wrapped as an instance of the [JSObject](http://docs.oracle.com/javase/8/javafx/api/netscape/javascript/JSObject.html) class. (As a special case, if the JavaScript object is a JavaRuntimeObject as discussed in the next section, then the original Java object is extracted instead.) The JSObject class is a proxy that provides access to methods and properties of its underlying JavaScript object.

1.2.2 Random idea:

Java scipt security, a list of urls that are allowed to run

ie. youtube.com/blablabla

1.2.2 Random idea:

Platform.runlater in all controller constructors test.

#### Threading

WebEngine objects must be created and accessed solely from the JavaFX Application thread. This rule also applies to any DOM and JavaScript objects obtained from the WebEngineobject.

**http://docs.oracle.com/javase/8/javafx/api/javafx/scene/web/WebEngine.html**

**https://docs.oracle.com/javase/8/javafx/embedded-browser-tutorial/js-javafx.htm**

# 2 Task completed:

# 3 GIT Repositories:

No commits made.

# 4 Trello boards

## 4.1 Board at the start of the week:

## 4.2 Board at the end of the week:

# 5 Toggl Time Logs

## 5.1 Weekly time Log bar chart:

## 5.2 Weekly Time Log:

## 5.3Weekly log Pie Charts: