**COSC346 Assignment 2 Report – Media Manager Library**

**Joseph McManamon – 6021556**

**Henry – 0000000**

**Current functionality – and what is still to come \*still to implement**

**Main window**

🡪 Load file(s) via the load button or use the open menu item (command O).

🡪 Double click to open media type.

🡪 Right click and select ‘edit metadata’ to view a files metadata

🡪 Search metadata values via the search box.

🡪 Export entire library to JSON using the menu (command S).

🡪 Export subset of the library to JSON after performing a search.

🡪 Bookmark bar that allows a view for all bookmarks (and smaller views of each media type).

**Media window**

🡪 Window resizes by changing the frame size.

🡪 Left/right key to go back/forward to the next media type (if you adjust the item view in the main window then this changes the items you scroll through).

🡪 Zoom in/out of images.

**Metadata window**

🡪 Add metadata key values using the + button.

🡪 Remove selected metadata values using the – button.

🡪 Edit metadata values.

🡪 Notes for each media type.

**Design**

Our application makes use of a main ViewController that essentially acts as the parent controller of all other views. When a user performs an action (e.g. a double click on a file in the list), the view controller determines what media type was clicked on and opens one of four view controllers for each of the four media types (DisplayAudio, DisplayVideo, DisplayDocument and DisplayImage). These are initialised with all the required information they’ll need to load the file and some additional index information so that other navigation controls (left/right arrows) can determine what to open next.

We felt this was a reasonable design for our application as it ensured different views were sufficiently independent of each other to open multiple files without issue, while still allowing the freedom of navigation that comes from knowing the circumstances in which they were initialised.

In addition to the above views, a DisplayMetadata and AddMetadata view controllers are used to handle all metadata and notetaking functions.

**Testing**

Nada

**Contributions**

Overall, there has been an even contribution to the project. We have worked on some aspects of the application together and other aspects of it independently. Regardless, we have come together multiple times over the past couple weeks to iron out bugs, figure out solutions and implement new functionality.

**Bonus marks**

Just need to make a case for them.

**Acknowledgements**

Many thanks to the people of stackoverflow.