1 Assignment 2 COSC346 - SPECIFICATIONS

This assignment is a continuation of the first assignment. You may use your own code or the skeleton provided or both.

- **Due Date**: Friday, 5th October 2018, at 23:59.
- **Weight**: This assignment is worth 20% of the mark for the paper. This mark will be formed from a maximum of 100 points that are awarded as explained below. The default late penalties apply.
- **Submissions**: During the lab session on 3rd of October, you will have a small presentation of your work. You will have up to 5 minutes each pair. This presentation is assessed in global scheme. Fill in your preferred time for the presentation in the following link Presentations Timetable. Unless specifically requested by **both** members of a pair, the same mark will be awarded to both pair members. The essential functionality will make up 60 points of your assignment mark. Up to 40 points will be awarded from the items of additional functionality. You can accumulate points up to a maximum of 100.
- **Marking**: This is a **pair** assignment. Fill in your preferred time for the presentation and send your Student ID, firstname, and family name, for both of you, with your presentation time to Hamza (hamza@hamzabennani.com). Do it as soon as possible.
- **What to submit**:
  - A complete Xcode project including the source code and all the resources that you use.
  - A report in PDF format. (See instructions below.)
- **How to submit**:
  - Commit and push your changes to your github repository created for the assignment (if you're using branches we'll be marking the master branch).
  - On the due date/time we will automatically collect all the submissions from each of your github repositories

Be careful to ensure that your submission only relies on files that you have actually submitted. For example, in the past students have submitted projects that referred to images using file paths that were not included in the collection of files that they submitted. Take note of the University's policy on Academic Integrity and Misconduct below.

2 Table of Contents

2.1 Learning Objectives

The goal of this assignment is for you to design and implement the GUI application described below. The specification is not entirely strict and precise: you will need to develop your own design to meet the stated requirements.

Successfully implementing the application is important, but you should also pay careful attention to the object oriented principles used within your code, and the usability of your GUI. These factors should be discussed in your report, and any other factors you found worthy to implement and discuss.

2.2 Problem Description

In this assignment you will design and implement a tool, using Cocoa, for managing the media library from Assignment 1. You must allow the user of your software to open multiple media files together.

In addition to displaying the media files, the purpose of this application is to allow the user using the software to record short notes connected to particular media file. While the means should be provided to record notes in plain text about particular media file, persistent storage of this data is not essential functionality but additional.

Also, the user needs to be able to record and jump to bookmarks so that they can move between different topics of media file and the different media files themselves. When presenting the media files, consider in your design that it should be possible to view a window showing just the media content. For example, if using a laptop that is plugged into a video projector, this would facilitate moving the media window onto the projector's display, while keeping notes and control windows on the laptop's display. Those watching the media content should not see unnecessary controls. However, you do not need to test your code on a multimonitor system: it would be sufficient to allow the user to drag the windows to the appropriate displays. It may be useful to be able to open more than one window containing the media content, e.g., so that one copy can be placed on a laptop's display, and another copy shown through a connected data projector.

You are welcome to design an application that has separate organising and presenting modes of operation, or to have one mode that allows for both presenting media material and organising it. However, when presenting the media content it must be posible to at least view the notes, and use bookmarks, even if at that time notes cannot be edited and bookmarks cannot be changed.

2.2.1 Academic Integrity and Academic Misconduct

Academic integrity means being honest in your studying and assessments. It is the basis for ethical decision-making and behaviour in an academic context. Academic integrity is informed by the values of honesty, trust, responsibility, fairness, respect and courage. Students are expected to be aware of, and act in accordance with, the University's Academic Integrity Policy.

Academic Misconduct, such as plagiarism or cheating, is a breach of Academic Integrity and is taken very seriously by the University. Types of misconduct include plagiarism, copying, unauthorised collaboration, taking unauthorised material into a test or exam, impersonation, and assisting someone else's misconduct. A more extensive list of the types of academic misconduct and associated processes and penalties is available in the University's Student Academic Misconduct Procedures.

It is your responsibility to be aware of and use acceptable academic practices when completing your assessments. To access the information in the Academic Integrity Policy and learn more, please visit the University's Academic Integrity website or ask at the Student Learning Centre or Library. If you have any questions, ask your lecturer.

2.3 Essential Functionality

Your media presentation tool must use a design that allows you to provide all of the essential functionality items shown in the list below.

1. Users can navigate through a given media file, including moving forward and backward for video file, moving to next image in the same folder, moving to the previous image, and jumping to a specific media file while showing a list or a grid.
2. Users can navigate between different media files within a set, providing both "local" navigation, such as "next media" and "previous media".
3. An indication is provided of the current media file being viewed, with the specific notes, and the current time for example for the video type, or page for the text file that is being viewed.
4. Your application provides functions for zooming in, zooming out and zooming to fit the media document contents in the window.

5. The application's controls should resize in a sensible manner when its containing window is resized.
6. Users can record brief textual notes that are related to a particular media file.
7. Users can bookmark particular topics of media or bookmark a oarticular media file then later use these bookmarks to jump back to the appropriate media file or list of media file.
8. Search within a set of media file related to a particular topic or particular type or media file.
9. A "useful" menu structure is implemented, that complements your other user interface controls.
10. Your "About" panel must be customised to include some relevant information about your project, e.g., giving credit to the creators of any resources that you acquire from elsewhere, such as icons.
11. The same import/export functionality for assignment one should be a feature.
12. For document preview, the previous extensions are a must to implement, other extensions are by choice. If the extension is not supported you can always show an error message image for example instead of a preview. However, in both cases (supported or not) the metadata should be previewed as well. As there are many different extensions of media file we will be testing the following ones that are easy to create on mac machines:
    - for images, a screenshot will provide a png extension;
    - for audio files, quicktime records them into m4a extention;
    - for video files, quicktime records them into mov extension;
    - for documents, plain text files.

2.3.1 Important Notes

There are many ways that you can achieve the navigation and display functions discussed above. While you are encouraged to be creative with your user interface, do remember that you will be marked on the usability of the system, so your design choices must be justified.

If you use any external kit provide it with your deliverable, discuss what has been used, how it has been used and keep in mind the copyright (variants of the Creative Commons licences are good options – just take note of the commercial aspects of the license).

You may want to look at Appkit Framewrok Documentation for document preview. Or use any other possible library for this purpose.

2.4 Additional Functionality

- Implement persistent storage of the notes made on media files. [10 points]

  - Your application should be able to write all of the required application state to disk and to be able to restore the notes from disk.
  - Loading notes back from disk should (try to) open the associated meida file.
- Implement rich formatting of the notes made on the PDF files. [10 points]
  - The formatting must not only apply to a whole note at a time: parts of the text within each formatted note must support having formatting applied to it.
- Fast search processing of the media files by topic and by type. [10 points]
- Add large displays of the time of day and the elapsed time within a video media file. [10 points]
  - Appropriate controls will need to be provided to start, pause and reset the elapsed time.
- Anything you think is useful and discussed in detail in your report [10 points]

2.5 Use of External Resources

If you wish, you can design your own icon for your application, but you are also welcome to use any freely-licensed external graphical content, such as the set that is available at http://www.gentleface.com/free_icon_set.html#geticons. You must include relevant credits and licensing information in your "About" panel. Also, be sure to test that these resources are included and correctly linked into the Xcode project that you submit.

2.6 Report

You must provide a report in PDF format that justifies your key GUI and object-oriented design decisions. Your report must also list the features that you chose to implement, and how users should use them—particularly where you have implemented keyboard controls or short-cuts. This is important for the marking of the assignment: you want to ensure that none of your hard work is missed. You do not need to go into detail explaining the internals of your implementation, as the comments in your code are likely to provide a useful guide on that front.

The report usually should not need to be longer than two pages, but should be clear and concise. You will lose marks if the report has obvious typos, spelling or grammatical errors. It must also explain the role taken by each member of the pair in completing the assignment.

2.7 Assignment Deliverables

Along with the source code, you must provide a report in PDF format. It is an important part of the assignment. Your report should indicate:

- the way in which object-oriented concepts were used in your design and implementation;
- how you tested your code;
- you must explain the role taken by each member of your pair; and
- if you implemented any extensions, how many bonus marks (up to 40 marks) you believe you should be awarded, and why. Note that the bonus marks can only be used to reach the maximum mark of 100.

2.8 Marking Guidelines

The assignment is worth 20% of your COSC346 mark and will be marked out of 100. There is not a strict marking scheme, as having one tends to disadvantage students, but the marking principles will include the following considerations:

- Elegant code will receive higher marks than ugly code (i.e., try to avoid ugly hacks).
- Being unable to implement each requirement does not mean that you will fail the assignment, but you will need to clearly indicate what is and is not working, and why.
- Code that makes good use of object-oriented principles will receive higher marks than code that does not.
- Do not forget to document and justify what you did in the code and in the report.
- Uncommented code will receive a maximum mark of 75 for the assignment.
- Assignments that do not provide a testing framework will receive a maximum mark of 60. It should also be possible to use your code as a library that does not run tests, i.e., your testing should be cleanly separated from your implementation.
- An assignments that is submitted without a report will receive a maximum mark of 70.
- You should make a case for how many bonus marks you deserve in your report, and argue in a small paragraph how many marks do you deserve for this asignment.