COSC346 Assignment 2: Lecture notes organiser

Submission information

Due date: Friday, 7th October 2016, at 5 PM.

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.

Marking: You are permitted to do this assignment individually or in pairs, but the marking will be different in each case.

Submissions from individuals: The *essential functionality* will make up 80 points of your assignment mark. Up to 20 points will be awarded from the items of *additional functionality*. You can accumulate points up to a maximum of 100.

Submissions from a pair: 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.

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: In the same manner as for assignment 1, collect everything that you need to submit into a directory, and then within MacOS on a Lab machine, run:
`/home/cshome/coursework/bin/submit346` *directory_name*

A submission from one member of a pair is sufficient if you choose to work in a pair. Late submissions will be handled in the same manner as for assignment one.

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.

Problem Description

In this assignment you will design and implement a tool, using Cocoa, for viewing a set of PDF lecture notes. In addition to viewing the lecture note PDFs, the purpose of this application is to allow the user to record short notes and bookmarks about the lecture notes with the aim of assisting their learning.

Some means should be provided to record notes in plain text about particular pages of any of the PDF documents, but as discussed below, you do not need to implement persistent storage of this data.

Essential functionality

Your lecture notes organiser must provide all of the essential functionality items shown in the list below.

- Users can navigate through a given PDF document, including moving to the next page, moving to the previous page, and jumping to a given page.
- Users can navigate between different PDF documents in a set, providing both "local" navigation, such as "next document" and "previous document", but also the ability to focus on a specific document.
- An indication is provided of the current document being viewed, and the current page of that document that is being viewed.
- Your application provides functions for zooming in, zooming out and zooming to fit the PDF document contents in the window.
- The application's controls resize sensibly when its containing window is resized.
- Users can record brief notes that are related to a particular page of a particular PDF document.
- Users can bookmark particular pages within particular PDFs and then later use these bookmarks to jump back to the appropriate page within the given PDF.
- A "useful" menu structure is implemented, that complements your other user interface controls.
- 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.

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.

The core document display functions of your PDF lecture notes organiser will be provided by Apple's PDF Kit. The PDF display functions will mostly involve calling into the appropriate methods of a `PDFViewer` class. PDF Kit documentation is available via the following URL: `https://developer.apple.com/library/mac/documentation/graphicsimaging/Conceptual/PDFKitGuide` although it focuses on Objective-C use. API documentation at `https://developer.apple.com/library/mac/documentation/GraphicsImaging/Reference/ImageKitReferenceCollection/` gives details for both Swift and Objective-C.

Additional functionality

- Implement persistent storage of the notes made on the PDF 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 PDF documents.

- 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: the text within the note must support having formatting applied to it.
- Provide support for use of PDF annotations. **[20 points]**
  - Subclasses of the `PDFAnnotations` class are likely to be very useful.
  - Implementing two types of annotation is sufficient.
  - You should provide undo and redo functions for annotations.
- Add an introductory startup page to your application **[10 points]**
  - This page should guide first-time users into the use of your software.
  - It should also provide useful functionality to more experienced users without getting in their way.
- Add document search capabilities. **[10 points]**
  - PDF Kit can provide the search facilities that you would need to use.
- Integrate help into your application. **[10 points]**
  - Your built-in help should operate in a manner similar to other MacOS applications, and should explain at least all of the main controls on your GUI, and menu items that you have implemented.

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.

Report

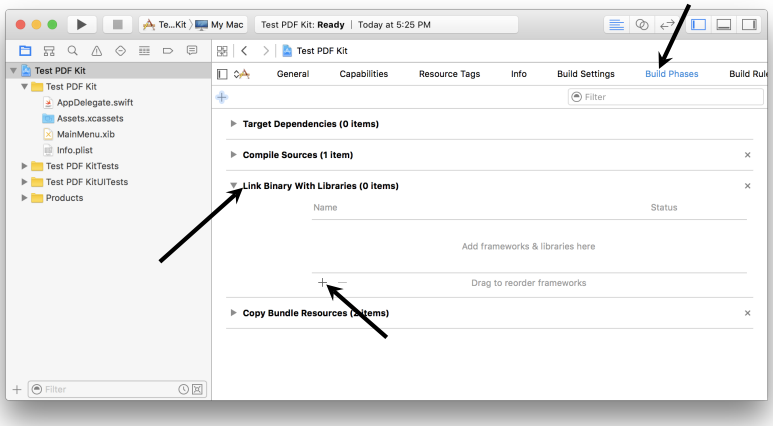
You must provide a report in PDF format that lists the features that you implemented, and how users should use them—particularly where you have implemented keyboard controls or short-cuts. This is important for the marking of the assignment: you want to ensure that none of your hard work is missed. You 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 one page. You will lose marks if the report has obvious typos, spelling or grammatical errors. Submissions from pairs must also explain the role taken by each member of the pair in completing the assignment.

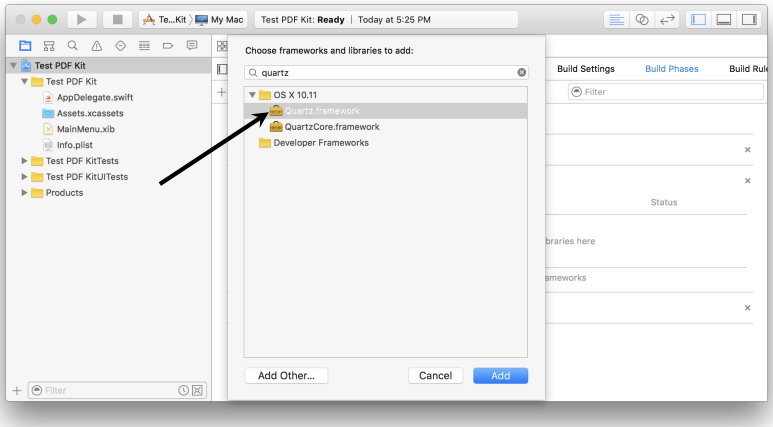
Using PDF Kit in your project

These directions are not part of the assignment specification. The steps below aim to help you get started with PDF Kit.

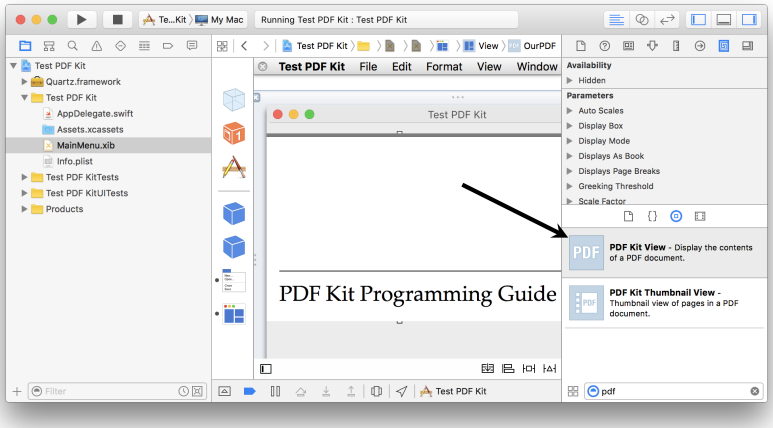
1. Create a Cocoa application project.
2. In the project settings select the "Build Phases" tab and expand the "Link Binary With Libraries" window.



3. Click on the "+" sign at the bottom left of that window and type `quartz` in the search box.
4. Select `Quartz.framework` and click "Add".



5. You can grab the PDFView object from the object library (as shown by the black arrow) and place it in your window (not shown).



6. Whenever referring to PDF Kit classes, ensure that your file contains the directive to `import Quartz` as shown in window below.