48024 Applications Programming Assignment 2

Topics:

OO Design, GUI, MVC, Tables, Lists

Learning Outcomes:

This assessment task addresses the following subject learning objectives (SLOs): 3, 4 and 5

Due date:

11:59PM Monday 19 October 2020

Weight:

30%

Platform:

PLATE

1. Individual work

All work is individual. You must write every line of code yourself except for code copied from study module sample code, lecture sample code, tutor demos or lab code.

You **MUST NOT** let another student see your solution code, and you **MUST NOT** look at another student's solution code. Sharing your code on public forums such as the discussion board, or Internet forums such as stackoverflow.com is not permitted. More information about Academic Misconduct can be found at: http://www.gsu.uts.edu.au/rules/student/section-16.html

2. Skeleton code

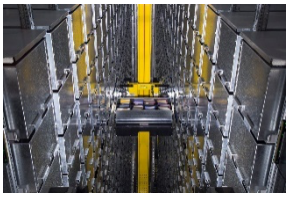
As a starting point for this assignment, you must use the skeleton code provided on PLATE (https://plate.it.uts.edu.au/) under Assessments-> Assignment 2. A plain version is available to import directly into NetBeans, which includes the all the structures required.

The skeleton code contains a file called progress.txt which you **must** fill in and submit with your project to PLATE as you progress on the assignment (read Section 6 of this document NOW!!!).

3. Expected workload

The time to do the assignment to a distinction level (i.e. a mark between 75% to 84%) has been estimated at 15 hours for a student of average ability who has completed all the tutorial and lab exercises.

4. Specification



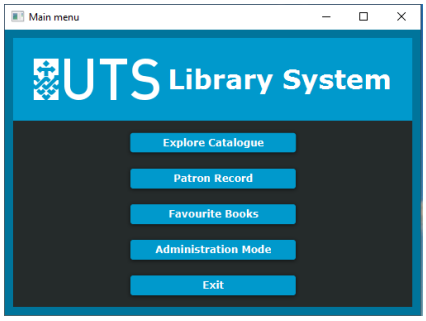
Library System

As Australia's newest university, UTS is reflecting the technology-focussed outlook with an exciting new Graphical User Interface (or "GUI") to go with the library's new system. Let's put those fancy new mice to work!

The specification is presented in several parts. In this document is given a series of screen shots and textual descriptions for visual reference. A demonstration is also presented in the video found on PLATE at Assessments-> Assignment 2. This demonstration video is considered part of the specification and contains important details about the dynamic function of the assignment.

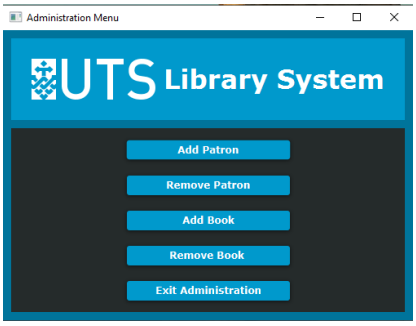
The screens presented below are given in no particular order, other than vague logical grouping, and in particular the order should not be construed as an indication of difficulty or recommended order of implementation.

Main Menu (10)



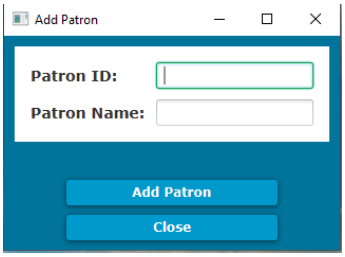
The main menu is opened when the application launches. It has buttons to access the catalogue, view patron records, view patron's favourites and the administration menu. It also has an exit button which shuts the entire application down. The main menu has a header section with the UTS logo and the title "Library System".

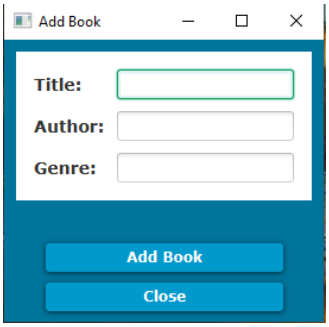
Administration Menu (5%)



The administration menu is launched from the main menu. It has buttons to add and remove patrons, and add and remove books. The close button closes the administration menu window. The administration menu includes the same header as the main menu (and catalogue menu).

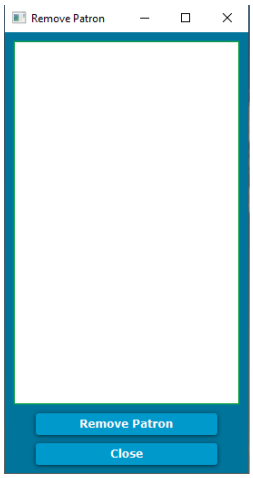
Add Patron and Add Book to Catalogue (10%)

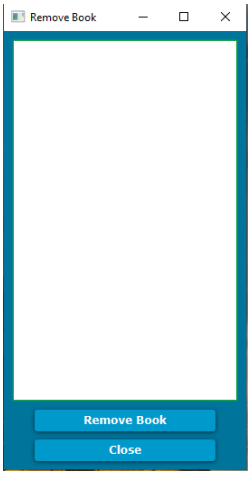




These two windows add a patron to the system, and a book to the catalogue respectively. Each has text fields for data entry, each with an appropriate label. The Patron ID field expects an integer, all others expect strings. The Add buttons handle the addition of the patron/book to the appropriate component of the model. The also both have areas to give feedback to the user if the action was successful or not.

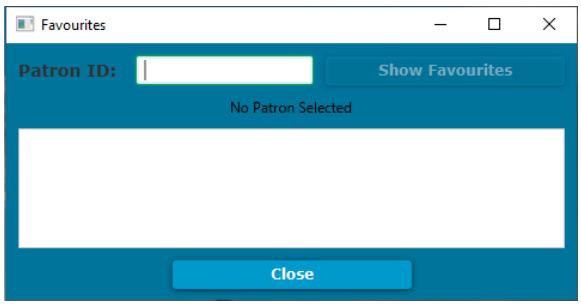
Remove Patron and Remove Book from Catalogue (10%)



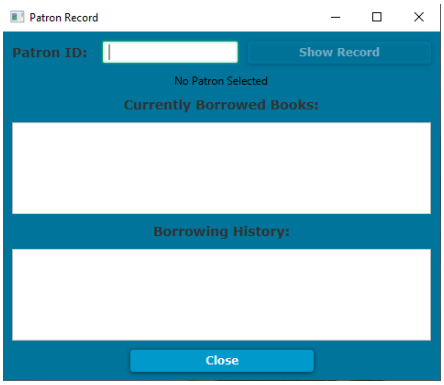


These two windows handle the removal of Patrons and Books. Remove Patron shows a list of the current Patrons, remove Book shows a list of the books available to be removed (which may not be all the books in the library). Each remove button removes the currently selected item in its window. The close buttons close their windows.

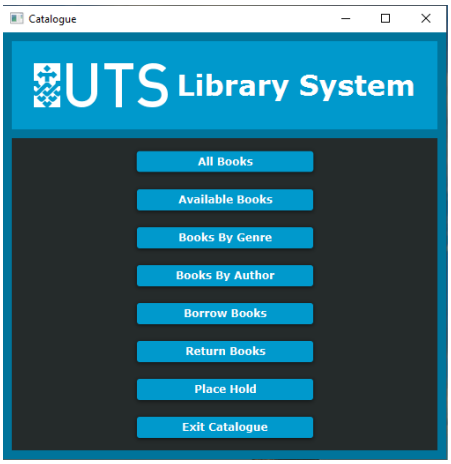
Patron Record and Patron Favourites (10%)



These two windows display the patron record and the patron's favourite books. The both include display areas for the related lists, suitably labelled. They both also include a text field for the entry of a patron ID, along with a label and a button to update the lists based on the entered patron ID. There is an area showing feedback text indicating which patron is selected.

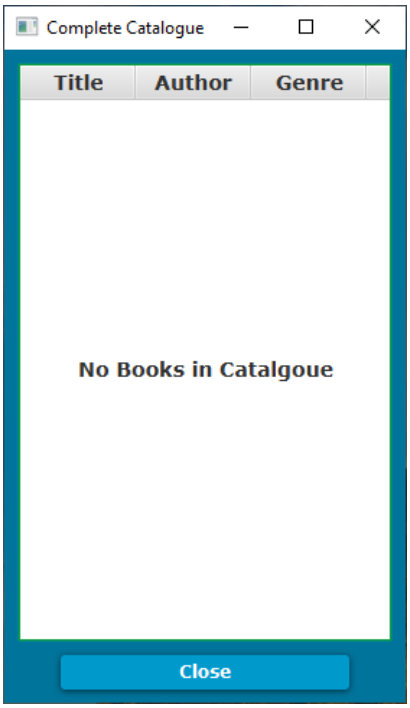


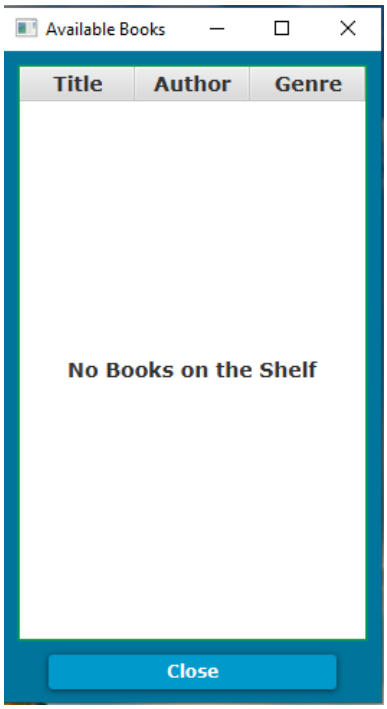
Catalogue Menu (5%)



The catalogue menu presents the options for interacting with the catalogue. It includes the menu header (as with the main menu and the administration menu).

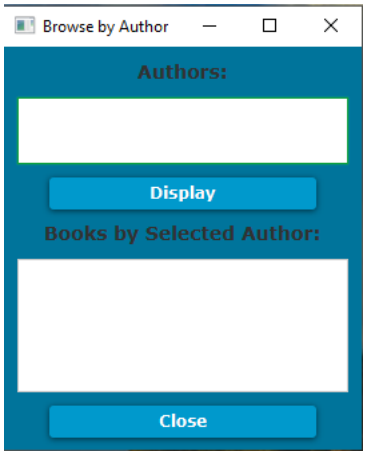
Show All Books and Show Available Books (10%)

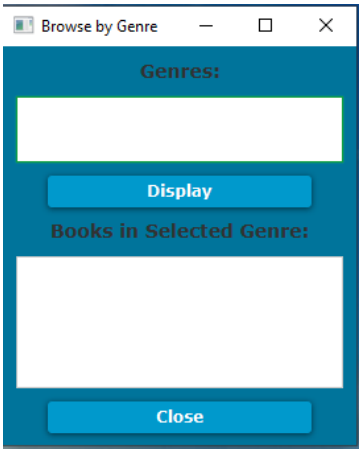




The complete catalogue (accessed from Show All Books) and the available books menu show the lists of all the books, and the books available for someone (but not necessarily everyone) to borrow, respectively.

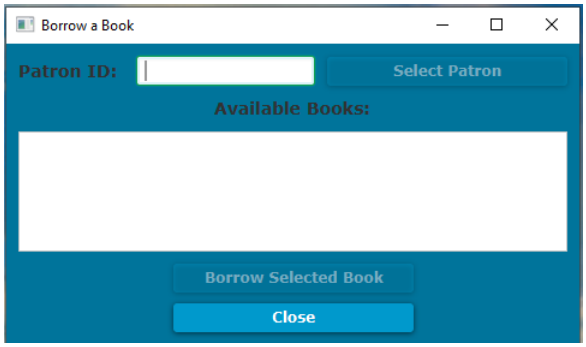
Browse By Genre and Browse By Author (10%)





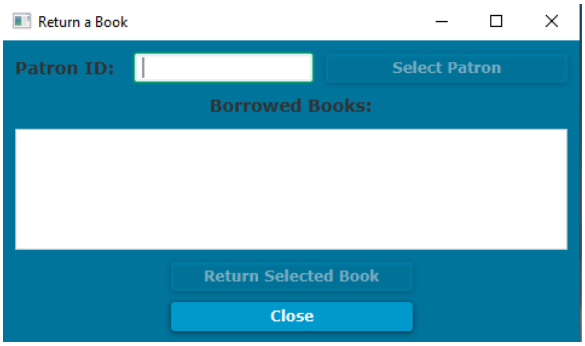
These two windows display labelled lists of authors/genres from which the selected item is used to display the suitably filtered list of books in the second list display area.

Borrow (10%)



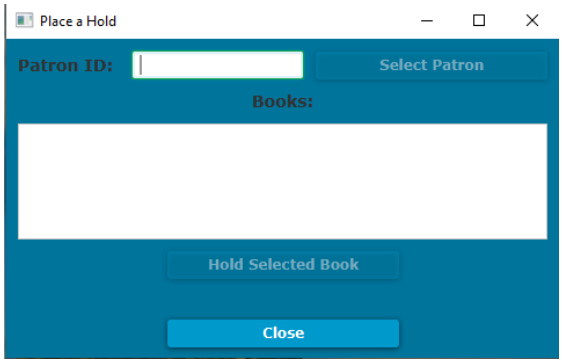
The borrow window allows the entry of a patron ID, which when the select patron button is clicked, shows a list of books that are available for that patron to borrow. This list should exclude books that are not on the shelf, or whose first hold is a different patron. Note that Select Patron button is disabled if there is no text in the Patron ID field, and the Borrow button is disabled if no book is selected.

Return (10%)



The return window operates similarly to the borrow window, but shows the books that the patron has currently borrowed (and are thus available to return). The buttons in this window have similar restrictions to the borrow window (text in Patron ID field, book selected).

Place Hold (10%)



The place hold window also operates similarly to the borrow and return windows, except the books available to be placed on hold includes all books. The window also additional includes an area for text feedback as to whether a hold was place, or if a hold already exists.

5. Requirements

Layout

To get full marks, you should layout your windows to look as close as possible to the screenshots. This means that you should try to duplicate the spacing between and around nodes that is shown in the screenshots, and the width and height of the nodes, and the alignment of the nodes. In the model solution, all hgap, vgap and spacing properties for GridPanes, HBoxes and VBoxes were set to 10 or 20.

Style

A CSS file is included in the skeleton code which provides all the styles used in the assignment.

Code

Your solution must satisfy the following code requirements:

- Your solution must follow the MVC architecture.
- Your solution must keep the package structure and class names that were provided in the skeleton code.
- The models must notify the views of changes by correctly applying the JavaFX property patterns and observable lists. Model data that can change must be observable. Model data that never changes need not be observable.
- The views must be laid out in FXML.

6. Submission to PLATE

READ THIS ENTIRE SECTION CAREFULLY

Included in the skeleton code is a file called progress.txt which you must fill out as you progress through the assignment. This file will contain lines such as these:

> [?] The Main menu window is at least partially done.
> [?] The Main menu window is done.
> [?] The Catalogue menu window is at least partially done.
> ...etc…

As you make progress on your assignment, you must edit this file by changing each [?] into a [y] and then submit your progress to PLATE. Don't forget to save this file before submitting. For example, after you get the main menu window partially done (even if you have only done a small amount), you edit this file as follows:

> [y] The Main menu window is at least partially done.
> [?] The Main menu window is done.
> [?] The Catalogue menu window is at least partially done.
> ...etc…

Then you submit your project to PLATE so that there is a record of what your code looked like when you first started to make progress on your Main menu window. After you complete the Main window feature, you should again update this file as follows:

[y] The Main menu window is at least partially done.
[y] The Main menu window is done.
[?] The Catalogue menu window is at least partially done.
...etc…

Then you submit your project to PLATE again so that there is a record of what your code looked like when you completed this feature.

It is not always required that you complete a feature before moving onto the next feature. For example, your progress.txt file may read:

[y] The Main menu window is at least partially done.
[?] The Main menu window is done.
[y] The Catalogue menu window is at least partially done.
...etc…

This would indicate that you partially completed the Main menu window, then moved on to the Catalogue menu window. This is allowed, as long as you have completed at least enough of the Main menu window that will allow you to correctly open the Catalogue menu window.

Important: If you don't submit your progress on a particular feature, then your marks for that feature won't count! That is, you are only marked for those features where you submit evidence of your progress. Be very careful to always submit your progress as soon as you make progress so that you don't lose any marks unnecessarily.

You are required to submit your project (including the updated progress.txt file) to PLATE regularly. Serious penalties apply if you do not submit your progress in small increments.

Penalty for lack of progress evidence: You must submit your progress to PLATE in increments of no larger than 25 marks. The penalty for making a submission to PLATE that is N marks higher than your previous best mark, where N > 25 is N marks. Here is an example:

Submission #1: 5 marks (OK. This is 5 marks higher than the previous best of 0)
Submission #2: 18 marks (OK. This is 13 higher than the previous best of 5)
Submission #3: 15 marks (OK)
Submission #4: 42 marks ("Just" OK. This is 24 higher than the previous best of 18)
Submission #5: 85 marks (Penalty of 43 . This jump is 43 marks higher than 42)

To avoid a penalty, submit your progress in smaller increments. That is, you should not introduce more than 25 new [y] answers in your progress.txt file in a single submission.

Your solution is to be submitted to PLATE at https://plate.it.uts.edu.au/ to Applications Programming / Assessments / Assignment 2. Your assignment should be submitted as a JAR file that includes:

- All Java source files required to compile your assignment.
- All FXML, CSS and image files required to run your assignment.
- The progress.txt file at the top level of your project directory structure.

Based on your submitted progress.txt file, PLATE will calculate a mark. This mark should NOT be considered in any way as your final mark. Rather, it should be considered as a "potential" mark. The final mark will be finalized by the subject coordinator after the deadline.

There is no scheduled late submission period as there is limited time left for peer marking of assignment2 . Any extension CANNOT be given after the due date.

You may also apply for special consideration for reasons including unexpected health, family or work problems. More information about how to apply for special consideration can be found at http://www.sau.uts.edu.au/assessment/consideration.html.

7. Peer marking and demonstration

In your scheduled week 12 lab class you must demonstrate your assignment to your tutor and be prepared explain parts of your code to your tutor if requested. If you are unable to explain your code, it may impact your marks. Your presence is required at this class. Any student who is not present without being granted prior permission may have up to 50% of their marks for this assignment deducted.

In addition to demonstrating your assignment, you will also be assigned two other students to peer mark, and two other students will be assigned to peer mark you. The purpose of this peer marking is to mark the functionality of your application which cannot be tested by PLATE. Your marks for functionality will be based on these peer marks after they are moderated by the subject coordinator. Aside from marks for the functionality, the subject coordinator will also mark your code to ensure that all code requirements have been met. Your final mark will be a combination of marks for functionality and marks for code (See "Marking scheme"). Note that you can only be marked for features that can be demonstrated to work.

8. Marking scheme

Task	Mark
Main Menu • All nodes are shown and FXML is used. - 4 • The layout is correct. - 2 • Fonts and colours are correct. - 1 • The buttons open the correct stages. - 1 • The exit button works. – 2	10
Administration Menu • All nodes are shown and FXML is used. - 1 • The layout is correct. - 1 • Fonts and colours are correct. - 1 • The buttons open the correct stages. - 1 • The close button works. - 1	5
Catalogue Menu • All nodes are shown and FXML is used. - 1 • The layout is correct. - 1 • Fonts and colours are correct. - 1 • The buttons open the correct stages. - 1 • The close button works. – 1	5
Add Patron and Add Book • All nodes are shown and FXML is used. - 4 • The layout is correct. - 4 • The close buttons work – 2	10
Remove Patron and Remove Book • All nodes are shown and FXML is used. - 1 • The layout is correct. - 1 • The patrons and books are displayed as a list. - 4 • The lists are displayed correctly using the observer pattern (*). - 3 • The close buttons work. - 1	10
Patron Record and Patron Favourites • All nodes are shown and FXML is used. - 1 • The layout is correct. - 1 • The buttons are correctly disabled and enable based on input. - 2 • The text feedback displays the correct information for valid and invalid patron IDs. - 2 • The patron record is correctly updated using the observer pattern (*). - 3 • The close buttons work. - 1	10

Show All Books and Show Available Books • All nodes are shown and FXML is used. – 1 • The layout is correct. – 1 • The columns are correctly labelled. - 4 • The lists are correctly updated using the observer pattern (*). - 3 • The close buttons work. - 1	10
Show Books By Genre and Show Books By Author • All nodes are shown and FXML is used. - 1 • The layout is correct. - 1 • The lists display the correct information (and only the correct information). - 3 • The lists are correctly updated using the observer pattern (*). - 4 • The close buttons work. – 1	10
Borrow • All nodes are shown and FXML is used. - 1 • The layout is correct. - 1 • The buttons are correctly disabled and enabled according to the conditions in the - 3 specification (including video). • The list is correctly refreshed when a book is borrowed. - 3 • The list displays the correct set of available books for that patron. - 1 • The close buttons work. - 1	10
Return • All nodes are shown and FXML is used. - 1 • The layout is correct. - 1 • The buttons are correctly disabled and enabled according to the conditions in the specification (including video). - 3 • The list is correctly updated using the observer pattern (*). - 4 • The close buttons work. - 1	10
Place Hold • All nodes are shown and FXML is used. - 1 • The layout is correct. - 1 • The buttons are correctly disabled and enabled according to the conditions in the specification (including video). - 3 • The text feedback correctly displays the result of attempting to place a hold. - 4 • The close buttons work. - 1	10

(*) The code will be checked by the subject coordinator in the 2 weeks following the due date.

9. Online support

The Assignment 2 discussion board has been set up in ED https://edstem.org/courses/4521/discussion/ so that students can ask questions, and other students can reply. A tutor may post a reply only if they think the student response was wrong, or in the case of correcting a mistake in the assignment specification.

You must not post Java code to the discussion board. The board is there to help you, not to provide the solution. Posting your code is academic misconduct and will reported. Each time this rule is violated, I will delete the code and post a comment of the form: “Strike 1: Posting code”. After 3 strikes, the discussion board will be deleted because it did not work.

FAQs (Frequently Asked Questions) and their answers are posted on PLATE alongside the assignment documentation. If you have a question, check the FAQ first, it may already be answered there. You should read the FAQ at least once before you hand in your solution, but to be safe check it every couple of days. Anything posted on the FAQ is considered to be part of the assignment specification. The FAQ will be frozen (no new entries) at the end of week 11; no questions will be answered after it is frozen.

If anything about the specification is unclear or inconsistent, contact the subject coordinator and Angela will try to make it clearer by replying to you directly and posting the common questions and answers to the FAQ. This is similar to working on the job, where you ask your client if you are unsure what has to be done, but then you write all the code to do the task. Email huan.huo@uts.edu.au to ask for any clarifications or corrections to the assignment.