Lab 5: GUI

Instruction

1. Click the provided link on MyCourseVille to create your own repository.
2. Create a new Project **Progmeth_Lab5_2024_1_{ID}_{FIRSTNAME}**
   - Example: **Progmeth_Lab5_2024_1_6631234521_Hayase**
3. Setup JavaFX in the created project.
4. Initialize git in your project directory.
   - Add ***.gitignore***
   - Commit and push initial codes to your GitHub repository.
5. Implement all the classes and methods following the details given in the problem **(some files or part of the files are already given, please see the *src* folder).**
   - You should create commits with meaningful messages when you finish each part of your program.
   - Don't wait until you finish all the features to create a commit.
6. Export your project into a **JavaFX JAR file (with resources)** called **Lab5_2024_1_{ID}** and place it at the root directory of your project. Your jar file must contain source code.
   - Example: **Lab5_2024_1_6631234521.jar**
7. Push all other commits to your GitHub repository.

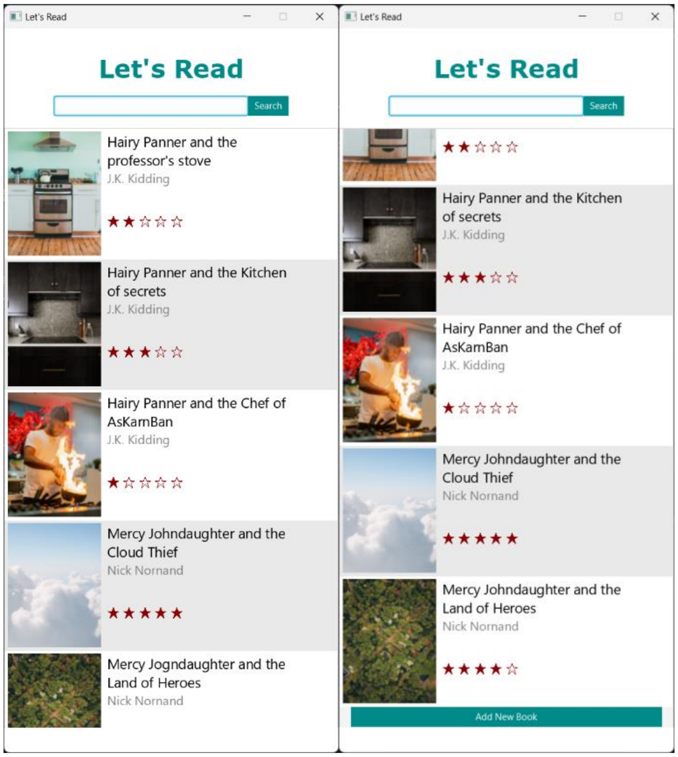
Problem Statement: Let’s Read!

Once upon a time, a clever programmer built a magical bookshelf app named "Let's Read". It began with an empty shelf, waiting to be filled with books.

Whenever a user finds a book, he/she can place it on the bookshelf by simply tapping on the app. Each book could be given a star rating, from one to five, based on how much the user likes it. Users can also write a short description for each book, sharing their thoughts and experiences.

If a user wants to find a book, he/she can just type the name into the search bar, and the book will appear on the screen!

Now, it's your turn to make this magic happen. Read the implementation details below and bring "Let's Read" to life!



Let's Read

Let's Read

Search

Hairy Panner and the professor's stove
J.K. Kidding
★★☆☆☆

Hairy Panner and the Kitchen of secrets
J.K. Kidding
★★★☆☆

Hairy Panner and the Chef of AsKamBan
J.K. Kidding
★☆☆☆☆

Mercy Johndaughter and the Cloud Thief
Nick Nornand
★★★★★

Mercy Jogndaughter and the Land of Heroes
Nick Nornand

Let's Read

Let's Read

Search

★★☆☆☆

Hairy Panner and the Kitchen of secrets
J.K. Kidding
★★★☆☆

Hairy Panner and the Chef of AsKamBan
J.K. Kidding
★☆☆☆☆

Mercy Johndaughter and the Cloud Thief
Nick Nornand
★★★★★

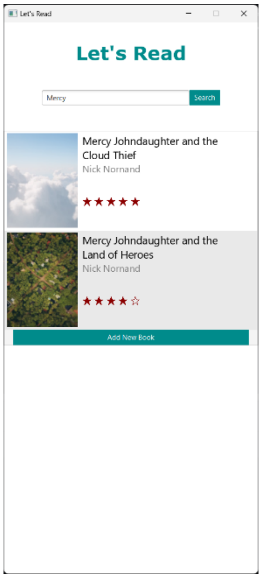
Mercy Johndaughter and the Land of Heroes
Nick Nornand
★★★★☆

Add New Book

Figure 1: Main screen

The main screen displays a booklist and a search pane. A button for adding a new book is located at the bottom of the booklist.

A book can be searched by name.



Let's Read

Let's Read

Mercy

Search

Mercy Johndaughter and the Cloud Thief
Nick Nornand
★★★★★

Mercy Johndaughter and the Land of Heroes
Nick Nornand
★★★★☆

Add New Book

Figure 2: Searching for books in the main page

Upon clicking the add-new-book button, you will be redirected to the add-new-book page. This page prompts you to input a title, an author, a rating, a cover image, and a description to create a new book entry. All input fields must be filled in order to create the book.   A back button is also provided, which will return you to the main page.

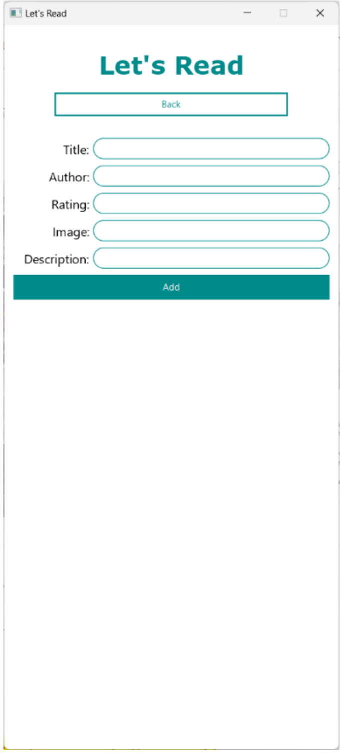
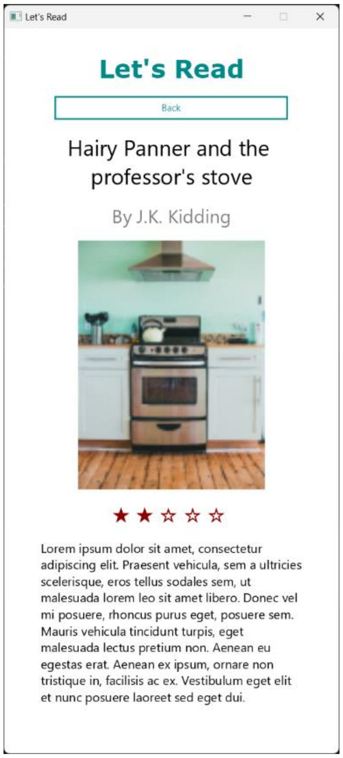


Figure 3: Add new book page.



Hairy Panner and the professor's stove

By J.K. Kidding

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent vehicula, sem a ultricies scelerisque, eros tellus sodales sem, ut malesuada lorem leo sit amet libero. Donec vel mi posuere, rhoncus purus eget, posuere sem. Mauris vehicula tincidunt turpis, eget malesuada lectus pretium non. Aenean eu egestas erat. Aenean ex ipsum, ornare non tristique in, facilisis ac ex. Vestibulum eget elit et nunc posuere laoreet sed eget dui.

Figure 4: Book's detail page.

Lastly, the book's detail page contains a brief description of the book, including its name, author, cover image, rating, and a short description. This page only has one interactive "back" button, which is used to return to the main screen.

Implementation Detail

1. Package item

1.1. Class Book /* ALREADY PROVIDED */

This class represents the book.

Field

Name	Description
- String name	The book’s name. (Cannot be blank)
- String author	The book’ author. (Cannot be blank)
- int stars	The book’s rating which represents as stars. (only in range of 0 to 5)
- Image image	The book’s cover.
- String description	The book’s description

Constructor

Name	Description
+ Book(String name, String author, int stars, String imagePath, String description)	Initialize all fields with the given value.
+ Book(String name, String author, String stars, String imagePath, String description)	Initialize all fields with the given value. If the 'stars' parameter is not in the correct numeric format, set it to 0.

Method

Name	Description
+ void setImageByPath(String imagePath)	Set the image of the book by image’s path.
+ Getters/Setters for everything	Except for the image. To set an image, use the 'setImageByPath' method instead.

2. Package utils /* You must implement this package from scratch */

2.1. Class GetDisplay /* You must implement this class from scratch */

This class contains utility methods to make displaying books in the GUI easier.

Method

Name	Description
+ Text Description(Book book, int fontSize, int wrappingWidth)	Create new Text with book's description. Then, - Set font size to fontSize - Set wrapping width to wrappingWidth Lastly, return the created Text.
+ ImageView image(Book book, int fitHeight)	Create a new ImageView with the book's image. Then, - Set preserve ratio to true - Set fit height to fitHeight Lastly, return the created ImageView.
+ Text name(Book book, int fontSize, int wrappingWidth, TextAlignment textAlignment)	Create a new Text with the book's name. Then, - Set font size to fontSize - Set wrapping width to wrappingWidth - Set text alignment to textAlignment Lastly, return the created Text.
+ Text author(Book book, int fontSize, int wrappingWidth, TextAlignment textAlignment)	Create a new Text with the book's author. Then, - Set fill to Color.GRAY - Set font size to fontSize - Set wrapping width to wrappingWidth - Set text alignment to textAlignment Lastly, return the created Text.
+ Text stars(Book book, int fontSize)	Create a new Text that shows the book's rating out of 5.

	Use "★" for each star the book has, and "☆" for the remaining out of 5. For instance, if the book has a rating of 3 stars, the resulting Text would be "★ ★ ★ ☆ ☆". Then, - Set fill to Color.DARKRED - Set font size to fontSize Lastly, return the created Text. **NOTE**: You can write '★' as "\u2605" and '☆' as "\u2606". **HINT:** You can make the string "xxx" by writing "x".repeat(3).

2.2. Class Goto /* You must implement this class from scratch */

Field

Name	Description
- RootPane rootPane	RootPane's instance

Method

Name	Description
+ void setRootPane(RootPane rootPane)	Setter for rootPane field.
+ void clear()	If the RootPane's instance has one child or none, do nothing. Otherwise, remove all children except the first one from the RootPane's children.
+ void mainPage()	NOTE: Please utilize clear()

	Remove all children except the first one from the RootPane's children. Create new ScrollPane with BookListPane’s instance. Then, - Set HBarPolicy and VBarPolicy to never. RootPane’s instance add new SearchPane and the created ScrollPane to children.
+ Button backToMainPageButton()	**NOTE:** Please utilize Goto’s methods Create a new Button with the string “Back”. Then, - Set border to border with stroke that - color: Color.DARKCYAN - style: SOLID - width: 2 - Set background to Color.WHITE. - Set text fill to Color.DARKCYAN. - Set preferred width to 300. - Set when the button is clicked **go to mainPage**. Return the created button.
+ void bookPage(Book book)	**NOTE:** Please utilize GetDisplay’s methods **NOTE:** Please utilize clear() Remove all children except the first one from the RootPane's children. 1. Create a new Text with the book’s name.

	○ Set font size to 28. ○ Set wrapping width to 336. ○ Set text alignment to center. 2. Create a new Text with the book’s author. ○ Set font size to 24. ○ Set wrapping width to 336. ○ Set text alignment to center. ○ Set text to “By {author}” (for example: “By A.J. Toe”) 3. Create a new ImageView with the book’s image. ○ Set fit height to 320. 4. Create a new Text with the book’s stars. ○ Set font size to 24. 5. Create a new Text with the book’s description. ○ Set font size to 16. ○ Set wrapping width to 336. Add a "Back to Main Page" button to RootPane’s children. Then, add all the created objects to RootPane’s children in this order: name, author, ImageView, stars, and description.
+ void addNewBookPage()	**NOTE:** Please utilize clear() Remove all children except the first one from the RootPane's children. Add a "Back to Main Page" button to the RootPane’s children.

	Then, create a new NewBookPane and add it to the RootPane's children.

3. Package application /* This package is partially given */

3.1. Class Main extends Application /* This class is partially given */

Constructor

Name	Description
+ void main(String[] args)	Launch the JavaFX application

Method

Name	Description
+ void start(Stage stage)	/* FILL CODE */ 1. Create a new Scene. ○ Set parent to RootPane's instance. ○ Set width to 430. ○ Set height to 932. 2. Set stage's scene to the created scene. 3. Set stage's title to "Let's Read" 4. Set stage's resizable to false 5. Show stage.

4. Package pane /* This package is partially given */

4.1. Class RootPane extends VBox /* This class is partially given */

Field

Name	Description
- RootPane instance	RootPane's instance

Constructor

Name	Description
- RootPane()	/* FILL CODE */ 1. Set background to Color.WHITE

	2. Set alignment to top and center. 3. Set spacing to 16 4. Set padding to 32 for the top and bottom, and 0 for the left and right. 5. Create a new Text with the string "Let's Read". a. Set text fill to Color.DARKCYAN b. Set font to i. Font family: Verdana ii. Font weight: BOLD iii. Font size: 32 c. Add the created text to children /* ALREADY PROVIDED */ **6. Set Goto's rootPane to this** **7. Goto mainPage**

Method

Name	Description
+ RootPane getInstance()	Return the instance of RootPane

4.2. Class BookPane extends GridPane /* You must implement this class from scratch */

Constructor

Name	Description
+ BookPane(Book book)	**NOTE:** Please utilize GetDisplay's methods and Goto's methods. 1. Set preferred width to 428 2. Set Hgap to 8 3. Set padding to 4. 4. Create a new ImageView with the book's image. a. Set fit height to 160.

	b. Add it to the grid at column 0 and span from row 0 to 3. 5. Create a new Text with the book’s name. a. Set font size to 18. b. Set wrapping width to 250. c. Set text alignment to the left. d. Add it to the grid at column 1 and row 0. 6. Create a new Text with the book’s author. a. Set font size to 16. b. Set wrapping width to 250. c. Set text alignment to the left. d. Add it to the grid at column 1 and row 1 7. Create a new Text with the book’s stars. a. Set font size to 16. b. Add it to the grid at column 1 and row 3. 8. Set on mouse clicked to go to the book’s page.

4.3. Class SearchPane extends HBox /* You must implement this class from scratch */

Constructor

Name	Description
+ SearchPane()	1. Set Alignment to center. 2. Create a new TextField. a. Set prompt text to “Find the book”.

	b. Set preferred width to 250. 3. Create a new Button with the string "Search". a. Set background to Color.DARKCYAN b. Set text fill to white. c. Set on action to update the searched books in the instance of BookListPane. Follow these conditions: i. If the created TextField is blank, set the searched books to all books in BookListPane's books. ii. Otherwise, set the searched books to all books whose names contain the text in the TextField. 4. Add the created TextField to children, then add the created button. **HINT:** For button's action {type} {name} = new ArrayList<> ({BookListPane's instance}.getBooks()); {name}.removeIf(...)

4.4. Class BookListPane extends VBox /* This class is partially given */

Field

Name	Description
- BookListPane instance	BookListPane's instance.
- ArrayList<Book> books	BookListPane's books.

Constructor

Name	Description

- BookListPane()	Initialize books and add some books to it. /* FILL CODE */ 1. Set fill width to true. 2. Set alignment to center. 3. Set searched books to books.

Method

Name	Description
+ ArrayList<Book> getBooks()	Getter for books.
+ Button newBookButton()	/* FILL CODE */ Create a new Button with the string “Add New Book”. a. Set the button’s preferred width to 400. b. Set the button’s background to Color.DARKCYAN. c. Set the button’s text fill to white. d. Set the button’s on mouse clicked event to go to the Add New Book page. **NOTE:** Please utilize Goto’s methods.
+ setSearchedBooks(ArrayList<Book> searchedBooks)	/* FILL CODE */ Clear all BookListPane’s children. For each book in the searched books, create an instance of BookPane from that book. Next, for every even-numbered book, set the background of the created BookPane to white. For each odd-numbered book, set the background to Color.color(0, 0, 0, 0.05).

	Finally, add the newBookButton to the children. NOTE: The first book is considered the 0th.

4.5. Class NewBookPane extends GridPane /* You must implement this class from scratch */

Constructor

Name	Description
+ NewBookPane()	1. Set padding to 12. 2. Set Vgap to 8. 3. Create 5 TextField inputs for name, author, rating, image, and description. NOTE: Please utilize input(). 4. Create 2 ColumnConstraints. a. Set one’s percent width to 25 and Halignment to right. b. Set the other one’s percent width to 75. 5. Create a new Button with the string “Add”. a. Set max width to 430. b. Set preferred height to 32. c. Set text fill to white. d. Set background to Color.DARKCYAN. e. Set on mouse clicked event to handleClick(…) 6. Add the created ColumnConstraints to the NewBookPane’s ColumnConstraints in the order of 25%, then 75%. NOTE: Please utilize label().

	7. Add the label “Title: ” to the grid at column 0 and row 0. 8. Add the label “Author: “ to the grid at column 0 and row 1. 9. Add the label “Rating: “ to the grid at column 0 and row 2. 10. Add the label “Image: “ to the grid at column 0 and row 3. 11. Add the label “Description: “ to the grid at column 0 and row 4. 12. Add the created name’s input to the grid at column 1 and row 0. 13. Add the created author’s input to the grid at column 1 and row 1. 14. Add the created rating’s input to the grid at column 1 and row 2. 15. Add the created image’s input to the grid at column 1 and row 3. 16. Add the created description’s input to the grid at column 1 and row 4. 17. Finally, add the created Button to the grid spanning from column 0 to 1 and row 5.

Method

Name	Description
- Text label(String s)	Create a new Text with the string **s**. Set font size to 16. Return the created Text.
- TextField input()	Create a new TextField. Set the background to white with the CornerRadii 16. - Set the border to border with stroke that - color: Color.DARKCYAN - style: SOLID

	- CornerRadii: 16 Return the created TextField.
- void handleClick(String name, String author, String rating, String image, String description)	If any parameters of the method are blank, do nothing. Create a new Book with the provided parameters and add it to the BookListPane's books. Set the searched books to include all books. Go to mainPage.

5. Criteria

The total is ***27*** points, which will be scaled to ***2.5***.

5.1. Main (2 points)

- (1 point) The main stage size is set at 430 (w) x 932 (h) and cannot be resized.
- (1 point) The title of the main stage is "Let's Read".

5.2. RootPane (3 points)

- (2 points) The RootPane's style and spacing follow the instructions.
- (1 point) The text "Let's Read" always stays at the top of the pane, regardless of the page, and its style is correct.

5.3. Goto (5 points)

- (3 points) All pages are rendered correctly.
- (1 point) The "back to main page" button follows the style and spacing instructions and functions correctly.
- (1 point) The BookListPane on the main page is scrollable and doesn't show a scrollbar.

5.4. BookPane (4 points)

- (2 points) The BookPane's style and spacing follow the instructions.
- (1 point) All child elements are positioned correctly.
- (1 point) Clicking on the BookPane triggers a redirection.

5.5. SearchPane (4 points)

- (2 points) The SearchPane's style and spacing follow the instructions.
- (2 points) The book searching functionality works correctly.

5.6. BookListPane (4 points)

- (2 points) The BookListPane's style and spacing follow the instructions.
- (1 point) The "new book" button follows the style and spacing instructions and functions correctly.
- (1 point) The searched books are displayed as instructed.

5.7. NewBookPane (5 points)

- (2 points) The NewBookPane's style and spacing follow the instructions.
- (2 points) The "add" button follows the style and spacing instructions and functions correctly.
- (1 point) All child elements are positioned correctly.

5.8. GetDisplay - This class doesn't carry any score.

- The main purpose of this class is to simplify other classes.