

Islington College



Emerging Programming Platforms and Technologies CS5004NI

GROUPWORK

Submitted By:

Biplav Malla 16033187

Prayash Bikram Shah 16033180

Shaurya Singh Deuja 16033181

Unik Gurung 16033191

Group C7 Year 2 Computing

Date: 8th January, 2018

Submitted To:

Mr. Dhurba Sen

Mr. Weenit Maharjan

Proposal

Pustakalaya is a small library run by a local community here in Kathmandu. It caters to a wide number of readers, from children of a local government school to elders who have been visiting the library ever since they held their first book. Since the library has been operating with the help of the community, it relies on volunteers to help manage the registers, ledgers and the books. Although the library has been running well for a number of years, the system is prone to a large number of errors. To minimize this, a proposed solution is to design a software that keeps track of all the books in the library.

Our system will introduce a software that will keep track of all the books in the library. The system will keep track of various information about the books, name, number, genre, status, condition and many more. It will show whether the book has been issued to someone else and whether it is available in the library.

DATA LIST

The Library Management System will contain the following:

Name	String
ISBN Number	Numeric
Genre	String
Year	Numeric
Edition	String
Available	Boolean

List of Features

1. The initial features that the system will have are as follows:
2. Will keep track of the number of books in the library.
3. Will show the details of all the books in the library, this includes the name, ISBN number, its edition, year published etc.
4. Will have a search function to show a specific book.

5. Will show if the book is available in the library or if it has been issued to a customer.
6. The software will be easy to use and the list can always be edited.

This is the initial and primary list of features, more features can and will be added if needed.

Tool Justification:

The Library Management System will be created using a high-level programming language, Java. The IDE that is going to be used is Netbeans. Netbeans is a versatile platform that is used by many all around the world. It has many advantages and for these reasons it is more superior to its competitors:

1. It is free and open source and yet it gets regular updates and is stable
2. It works with all kinds of operating systems so it is versatile and can be used anywhere by anyone on any system
3. It has more plugins compared to most of its competitors, because of that it provides more advanced support to the programmers than other platforms
4. It is more structured and helps run efficiently to ensure there are no crashes or problems when the code is large
5. It makes it easier for the programmer to use the software due to its easy to use interface
6. The drag and drop option to create GUI saves time and lets the programmer create the interface exactly the way they want to. (This feature is important and useful and a lot of another IDEs lack it)
7. It helps the programmer structure code so it is understandable to other programmers if they ever have to.

Table of Contents

1.Individual Tasks.....	1
2.Introduction	2
4.Description of Methods	3
5.Binary search.....	4
5.1 Binary Search Used in the system.....	4
6.System Testing.....	6
6.1 Running GUI for the system:	6
6.2 Adding Items to the tables:.....	7
6.3 Searching:.....	8
6.4 Open File:	9
6.5 Dialog Boxes:	10
7.Conclusion	12
8.References.....	13
9.Appendix	14
9.1 User Guide for the System.....	14
9.2 Source Code.....	15

Table of Figures:

Figure 1 Binary Search Algorithm (Geeksforgeeks, 2017)	4
Figure 2 Binary Search 1	5
Figure 3 Binary Search 2	5
Figure 4 GUI of the System	6
Figure 5 Adding Values in JTable.....	7
Figure 6 Search Result.....	8
Figure 7 Open File	9
Figure 8 User guide opened	9
Figure 9 Dialog Box 1	10
Figure 10 Dialog Box 2.....	10
Figure 11 Dialog Box 3.....	11

Table of Tables:

Table 1 Task Distribution	1
Table 2 Description of Method	3
Table 3 Testing Box 1	6
Table 4 Testing Box 2.....	7
Table 5 Testing Box 3.....	8
Table 6 Testing Box 4.....	10
Table 7 Testing Box 5.....	11

1. Individual Tasks

Group Member	Task Description
Biplav Malla	Coding of the program for the Search, Sort, Add, menu bar methods.
Prayash Bikram Shah	Coding for remove, clear, exit, GUI components and Documentation.
Shaurya Singh Deuja	Documentation and Coding of GUI, research on sort and search for the implementation
Unik Gurung	Documentation and Coding of GUI, research on sort and search for the implementation

Table 1 Task Distribution

2.Introduction

This group work was to create a GUI of a menu system based on an organization. We had to use Java Language on Netbeans IDE platform to create the GUI. The organization could be any kind of assumption or real time one. We choose a library menu system naming it as Pustakalya. This coursework displays the basic idea of the menu system for a library where you can add books and search books. In this coursework there were minimum requirements like there must be a J table which has at least 5 categories in it. Of those five categories there must be a radio button combo box. There must be buttons for clearing the table. Clearing the interactive mode, exit for closing GUI and add button to add values entered in interactive mode to the JTable in the GUI. A search button and a text filed was required too. Going through this basic knowledge we gained while making this GUI menu system and testing of the system.

4. Description of Methods

Method	Description
MenuInfo()	All the GUI components are inserted in it.
AddToTable()	Adds values to table
Remove()	It is used for removing an element of the specified index from a list. It the selected row from the column.
Open()	It opens user guide
Exit()	This method closes the GUI.
Search()	It is used for searching for a book form the JTable
Clear ()	This method clears the value from the Input fields.
getText()	This method gets the value.
isSelected()	This method sets the selection of item.
setRowCount()	This method sets the row count.
setText()	This method sets the text field's value.
setSelectedItem	This method sets the selected item.

Table 2 Description of Method

5.Binary search

Binary Search is an approach in which the program will search an item on a sorted array. This algorithm divided the interval of search in half repeatedly. It compares the value of the search key with the interval and if the value is less than of the middle of the interval then it narrows down to the that half. If it is not less than it narrows downs to the other half. This process goes on and on until the value of the is found on the array or the interval is empty. (Geeksforgeeks, 2017)

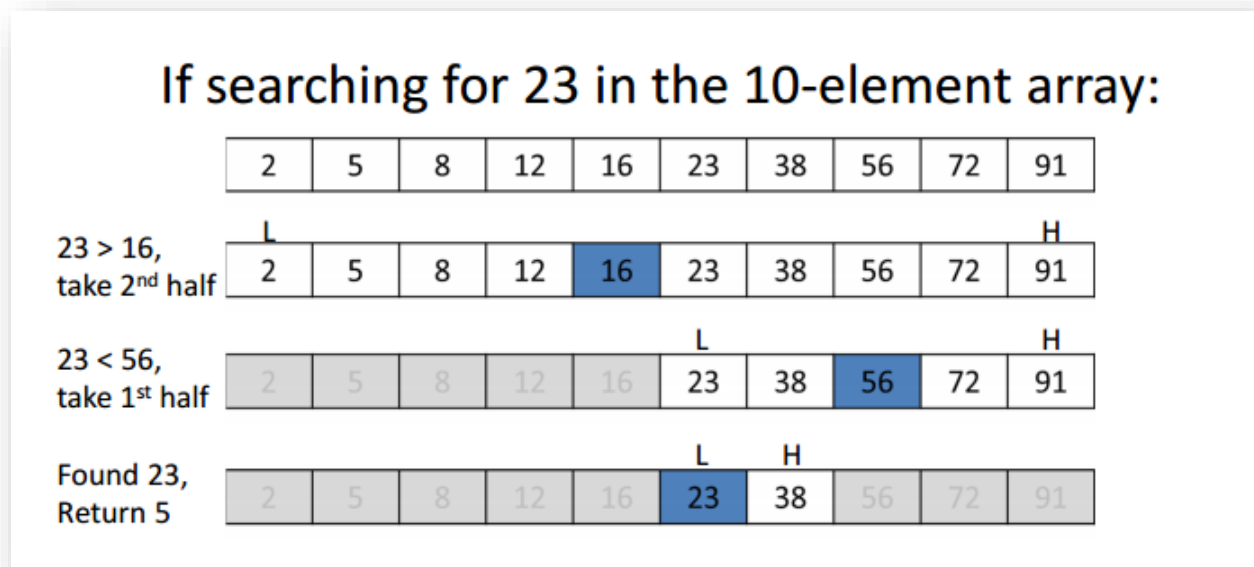


Figure 1 Binary Search Algorithm (Geeksforgeeks, 2017)

5.1 Binary Search Used in the system

The search function is worked based on the ISBN number of the books taking its integer value from the sorted array which is retrieved after performing bubble sort. The lower value is the index and the length of the array is subtracted by one is considered as the higher value. The sum of there is calculated to find a mid-value and that mid value and the key value is checked. If the mid value is the higher than the numbers then the mid value is subtracted by one and the output of that is taken as the latest higher value and the old mid value is added by one which becomes as the new lower value. This repeats until the value matches and returns the index number and if it does not match any value in the sorted array then the whole process is terminated.

```

public Pustaks getPustaks(int ISBN) {
    int minIndex = 0;

    int maxIndex = pustakList.size() - 1;

    pustakList = sortArrays(pustakList);
    while (minIndex <= maxIndex) {
        int midIndex = (minIndex + maxIndex) / 2;
        int tempISBN = 0;
        tempISBN = pustakList.get(midIndex).getISBN();
        if (tempISBN == ISBN) {
            return pustakList.get(midIndex);

        } else if (tempISBN < ISBN) {
            minIndex = midIndex + 1;
        } else if (tempISBN > ISBN) {
            maxIndex = midIndex - 1;
        }
    }

    return null;
}

```

Figure 2 Binary Search 1

```

public void Search(int ISBN) {
    model.setRowCount(0);
    Pustaks tempPustakList = getPustaks(ISBN);
    Object obj[] = {tempPustakList.getName(), tempPustakList.getISBN(),
        tempPustakList.getGenre(), tempPustakList.getYear(), tempPustakList.getEdition(), tempPustakList.getAvailable()};
    model.addRow(obj);
}

```

Figure 3 Binary Search 2

6. System Testing

6.1 Running GUI for the system:

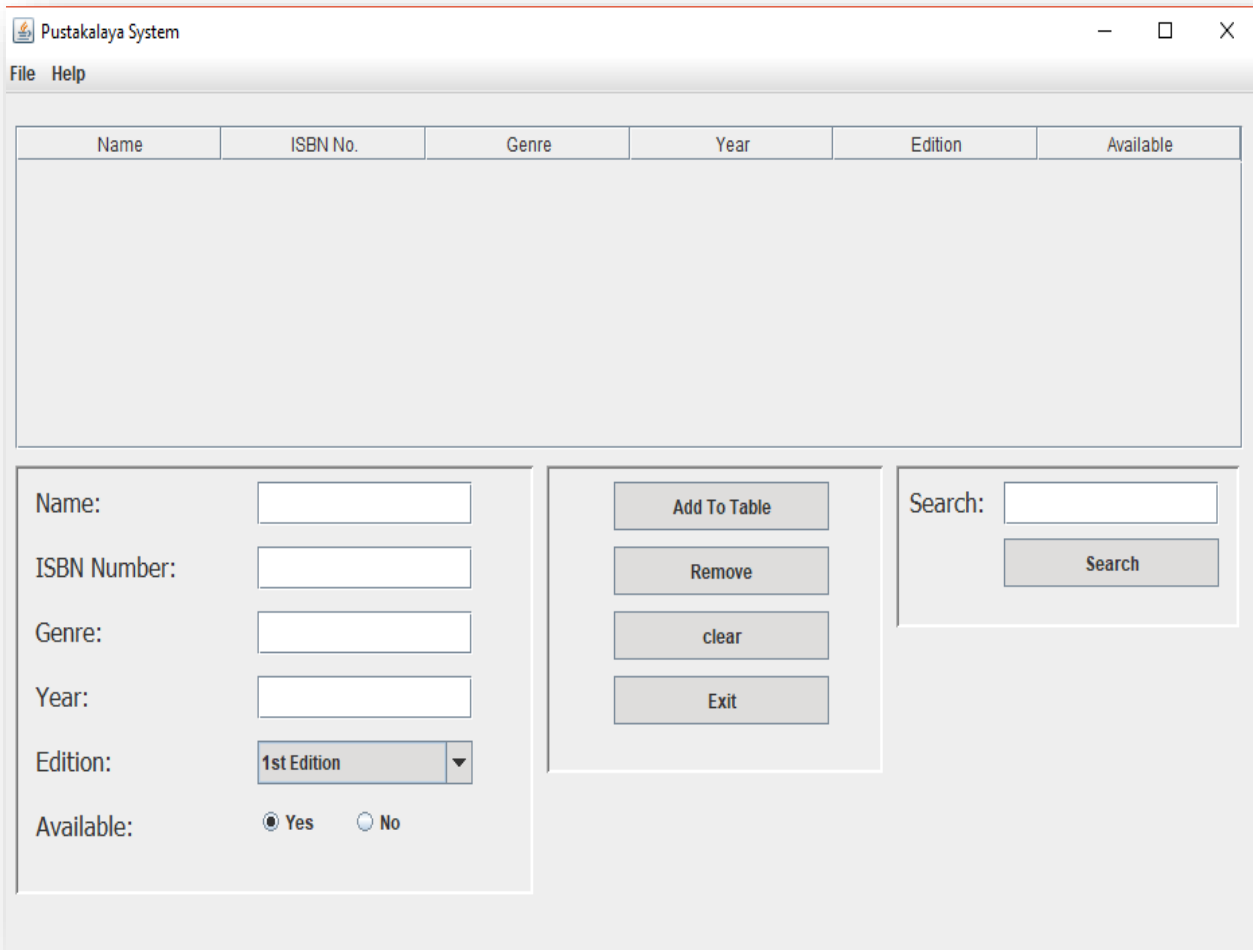


Figure 4 GUI of the System

Testing Description	
Purpose	Running the system.
Action	Display a Graphical User Interface.
Expected Result	GUI of the menu system.
Obtained Result	GUI of the menu system.
Conclusion	GUI was displayed.

Table 3 Testing Box 1

6.2 Adding Items to the tables:

The screenshot shows a Java Swing window titled "Pustakalaya System". It contains a table with the following data:

Name	ISBN No.	Genre	Year	Edition	Available
Charlie and the Chocolate...	9780141	Fiction	1964	2nd Edition	Yes
Two States	8770351	Fiction	2009	1st Edition	No
Game of Thrones	7751389	Fiction	1996	1st Edition	Yes
In Cold Blood	5634921	Non-Fiction	1996	1st Edition	Yes
Silent Spring	4124968	Non-Fiction	1962	3rd Edition	No
Into Thin Air	6783452	Non-Fiction	1997	4th Edition	Yes
Black Boy	1783672	Non-Fiction	1945	3rd Edition	No
Macbeth	2674893	Drama	1606	1st Edition	Yes
Merchant of Venice	3675894	Drama	1605	1st Edition	Yes
The Good Daughter	9853781	Mystery	2017	1st Edition	Yes
The Magpie Murders	9653783	Mystery	2016	1st Edition	Yes

Below the table is a form to add new items:

Name:

ISBN Number:

Genre:

Year:

Edition:

Available: ☐ Yes ☒ No

Buttons: Add To Table, Remove All, clear, Exit

Search:

Figure 5 Adding Values in JTable

Testing Description	
Purpose	Adding Values to the table
Action	The values were inserted into respective columns.
Expected Result	Values in the table in their own columns.
Obtained Result	Values in the table in their own columns.
Conclusion	Values were displayed in the JTable.

Table 4 Testing Box 2

6.3 Searching:

The screenshot shows a Java Swing window titled "Pustakalaya System". It has a menu bar with "File" and "Help". Below the menu bar is a table with the following data:

Name	ISBN No.	Genre	Year	Edition	Available
Romeo and Juliet	2459872	Drama	1567	2nd Edition	Yes

Below the table, there are input fields for adding a new book:

- Name:
- ISBN Number:
- Genre:
- Year:
- Edition:
- Available: ☒ Yes ☐ No

In the center, there are four buttons: "Add To Table", "Remove", "clear", and "Exit".

On the right, there is a search section with a label "Search:" followed by a text field containing "2459872" and a "Search" button.

Figure 6 Search Result

Testing Description	
Purpose	Searching Values in the table.
Action	The ISBN number of the book is typed on the search text field to search that particular book in the table.
Expected Result	The searched result is should be displayed.
Obtained Result	The searched result has been displayed.
Conclusion	The search result was displayed in the JTable.

Table 5 Testing Box 3

6.4 Open File:



Figure 7 Open File

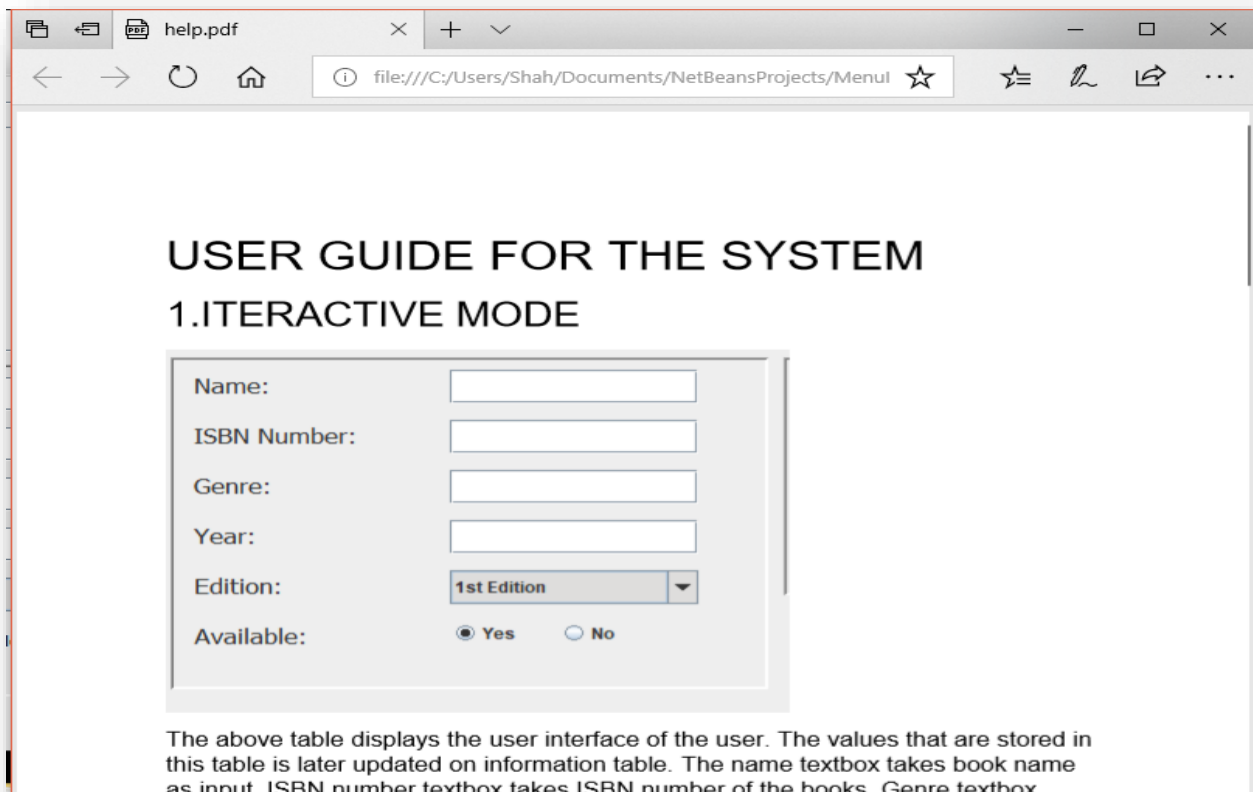


Figure 8 User guide opened

Testing Description	
Purpose	Opening User guide file named help.
Action	Open from menu bar is clicked.
Expected Result	A user guide manual named help.pdf opens.
Obtained Result	A user guide manual named help.pdf opens.
Conclusion	Help.pdf files open up.

Table 6 Testing Box 4

6.5 Dialog Boxes:

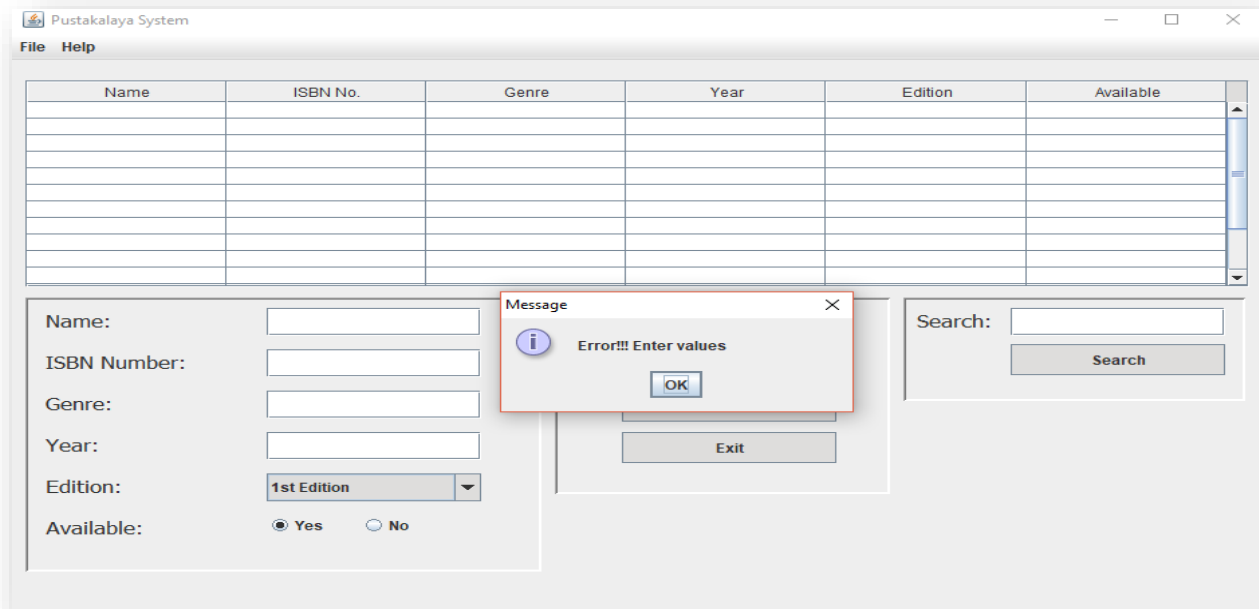


Figure 9 Dialog Box 1

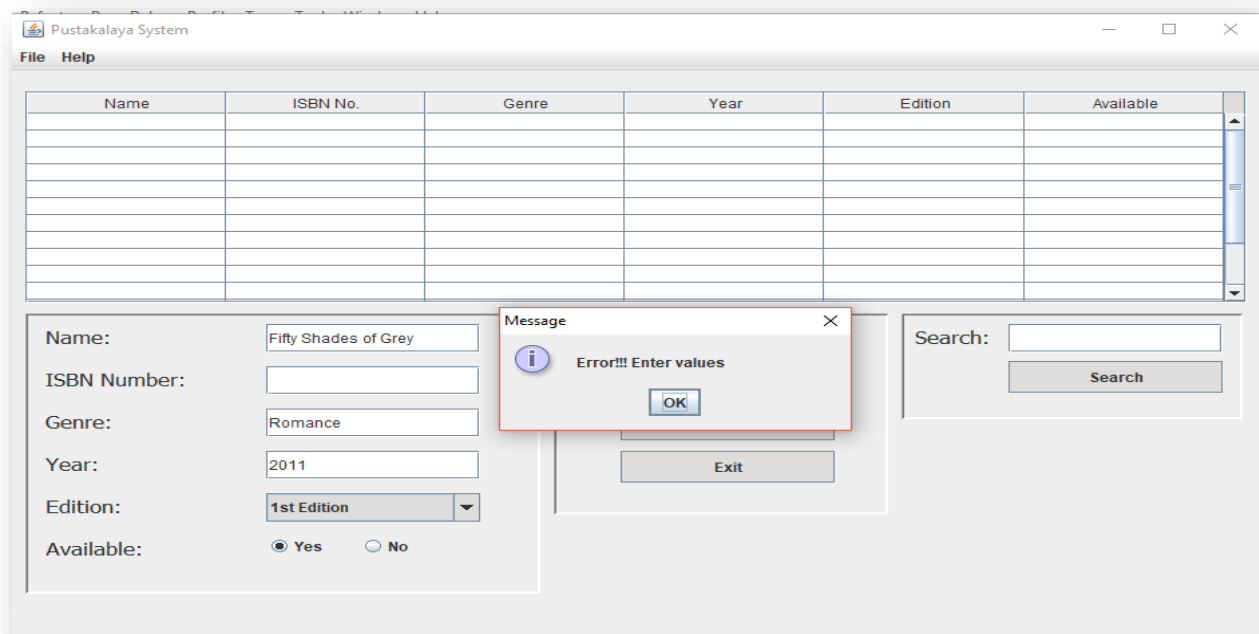


Figure 10 Dialog Box 2

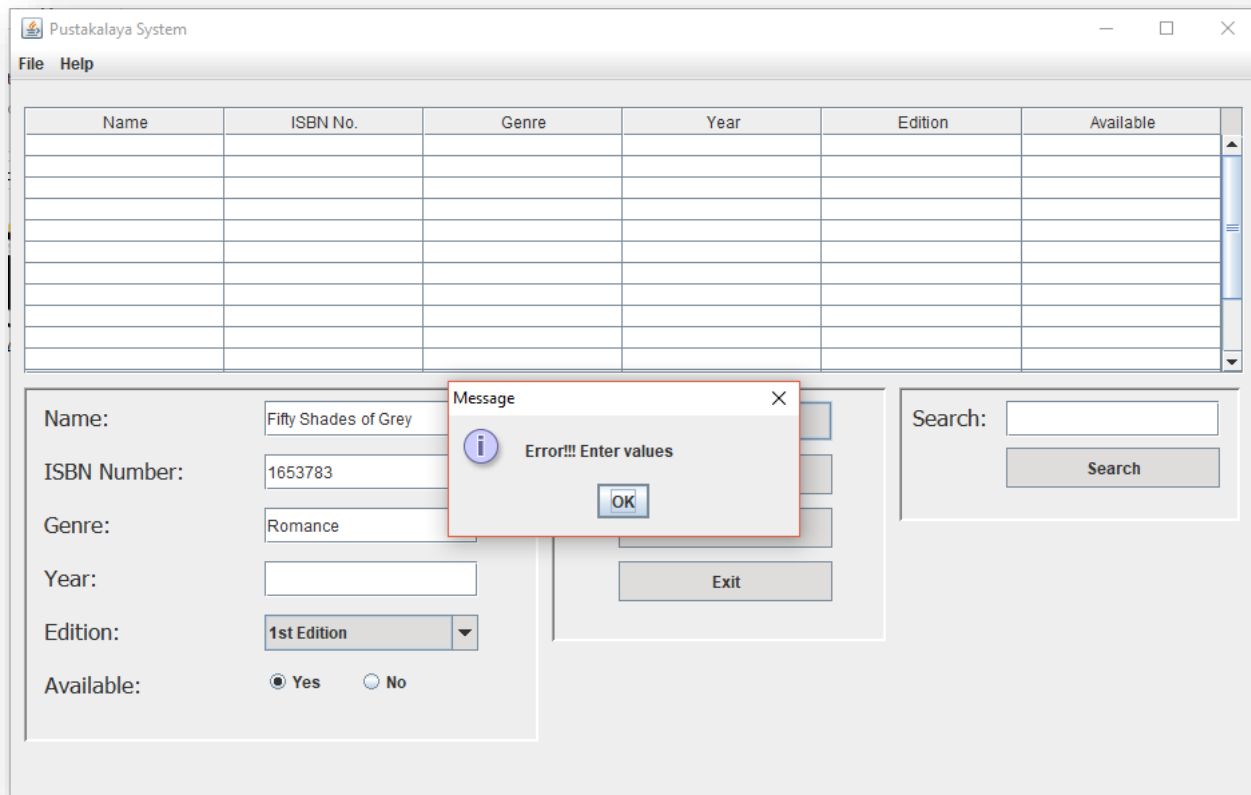


Figure 11 Dialog Box 3

Testing Description	
Purpose	Proper Error Dialog boxes while trying to add values in the table.
Action	Add values without inserting anything on the interactive mode.
Expected Result	Error Dialog Boxes.
Obtained Result	Error Dialog Boxes.
Conclusion	Error Dialog Box displayed while trying to add values without entering anything in ISBN, Name and Year text field.

Table 7 Testing Box 5

7. Conclusion

Completing the coursework has been a new learning experience for the whole team. With the guidance of our module leader and friends, creating a GUI has been a wide success. One of the most important lessons that the coursework taught us was learning from our mistakes. It helped us overcome our fear of coding. The topics we covered during this group work were, making GUI components of a menu system and adding values to the JTable and sorting them and searching certain values in that table. Fixing the errors that occurred while completing the coursework was a real rollercoaster ride. Facing the problems and patience to overcome them helped us in making a habit of researching the problems.

As Collin Powell once said “A dream doesn’t become reality through magic; it takes sweat, determination and hard work.” Personally, we have gotten through this ordeal, but still there are many obstacles that we have a lot to overcome along the way.

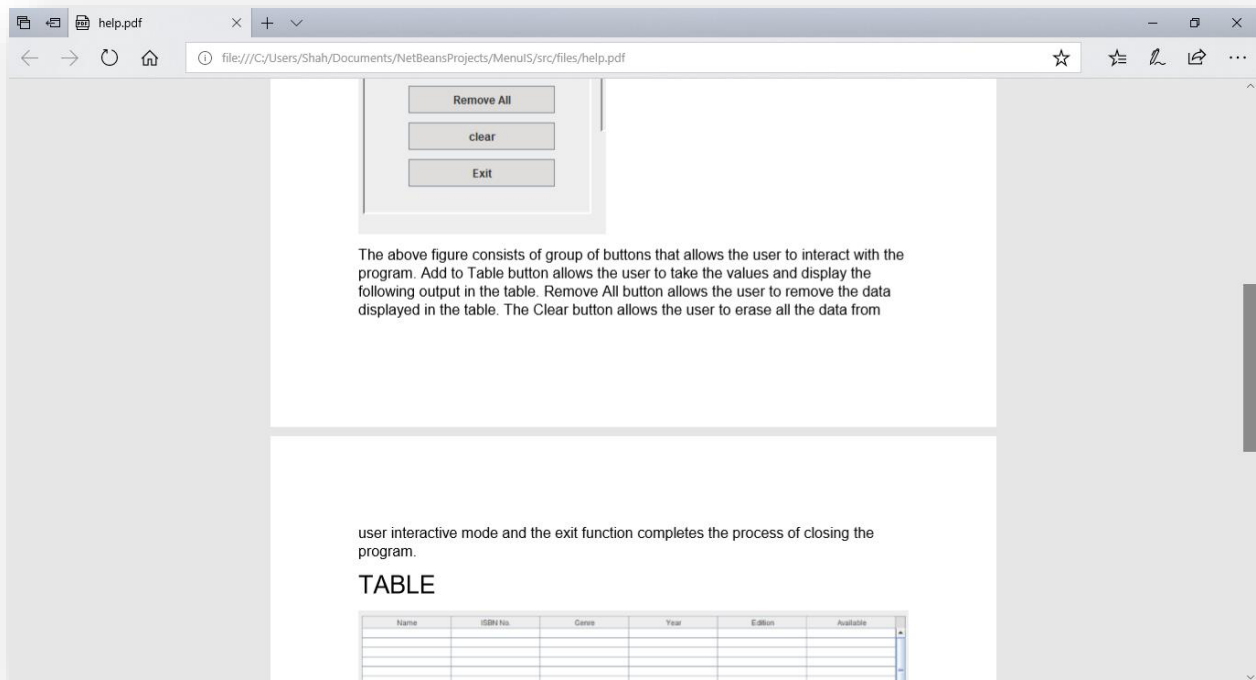
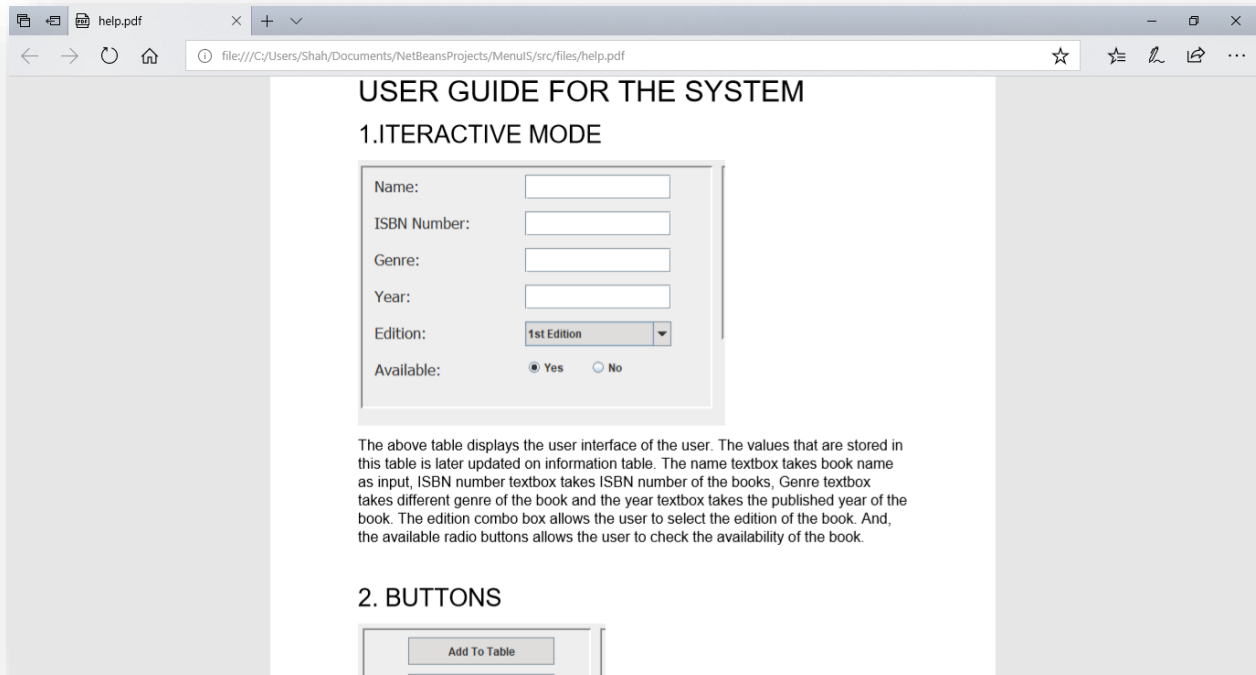
Being a programmer is all about being able to think logically and make a proper algorithm before coding so that there will not be any errors. This project not just helped us in the learning process of Java programming but also about the ideas and critical thinking importance while doing a project.

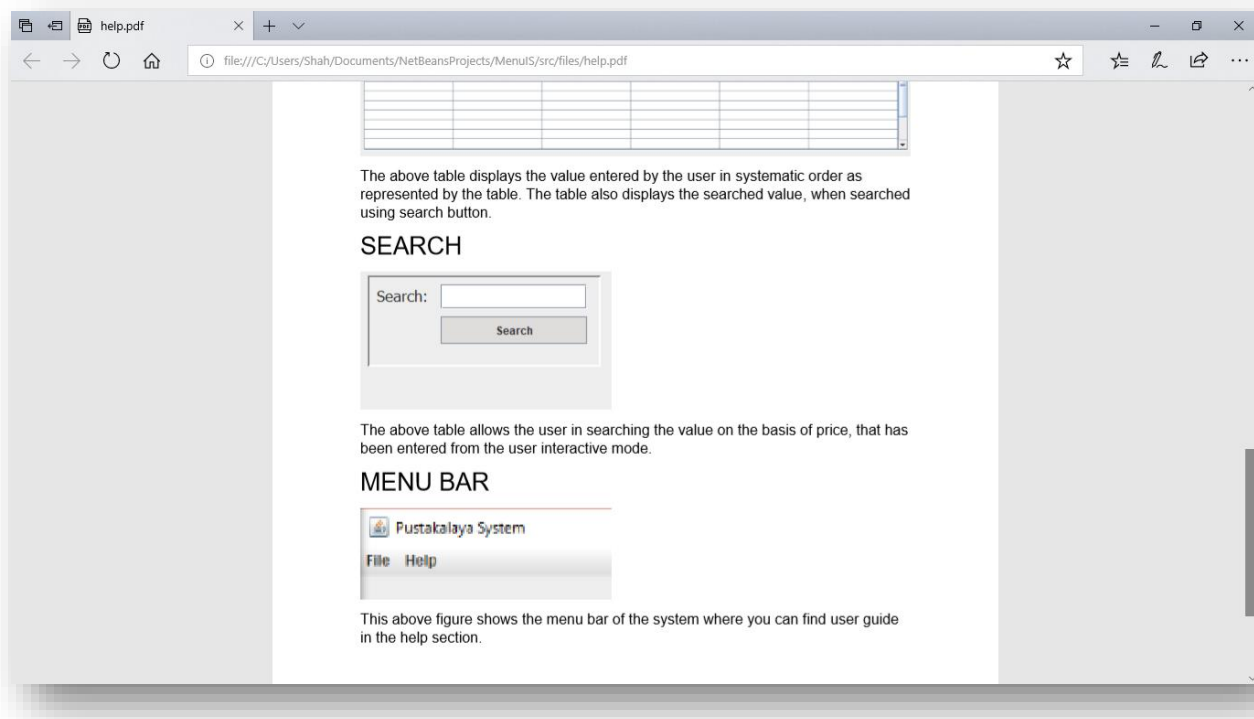
8. References

Geeksforgeeks, 2017. *Binary Search - GeeksforGeeks*. [Online]
Available at: <https://www.geeksforgeeks.org/binary-search/>
[Accessed 10 December 2017].

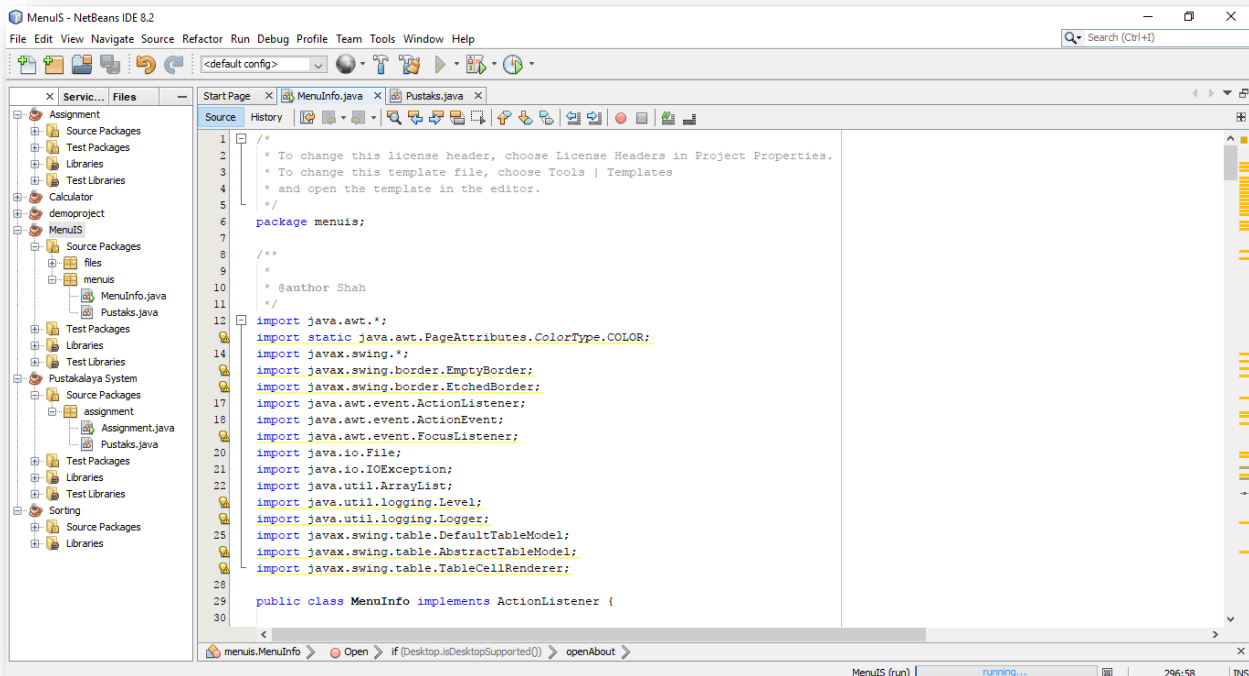
9. Appendix

9.1 User Guide for the System

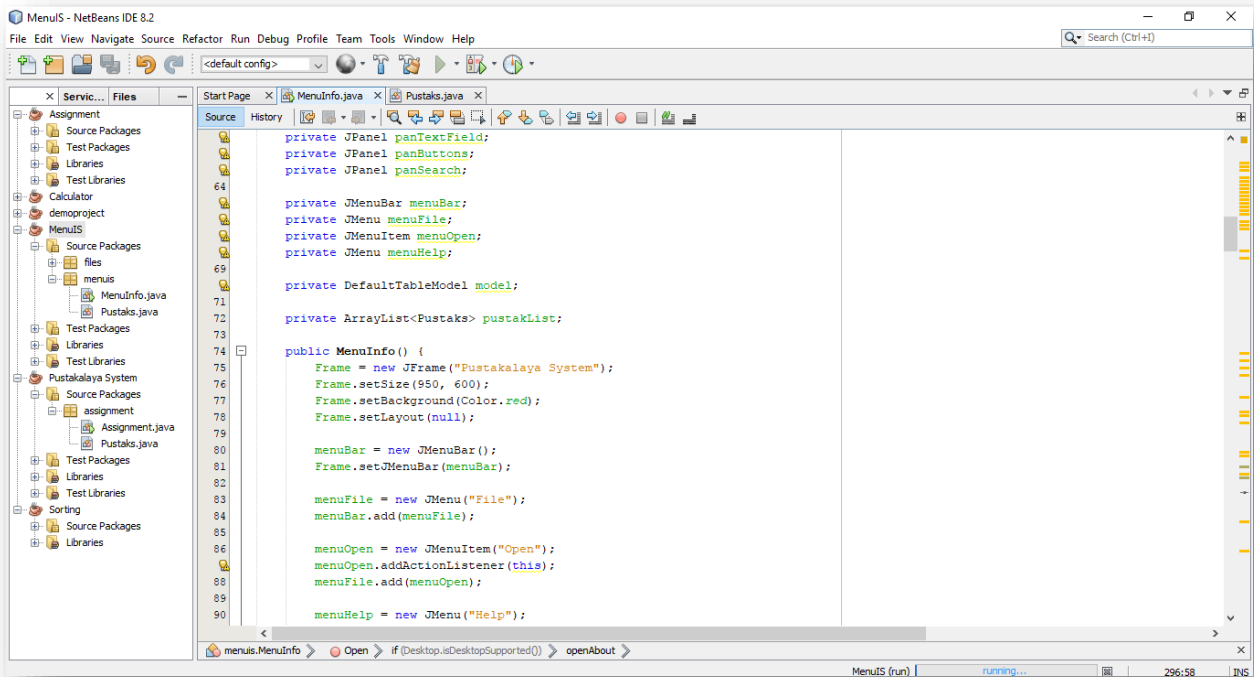
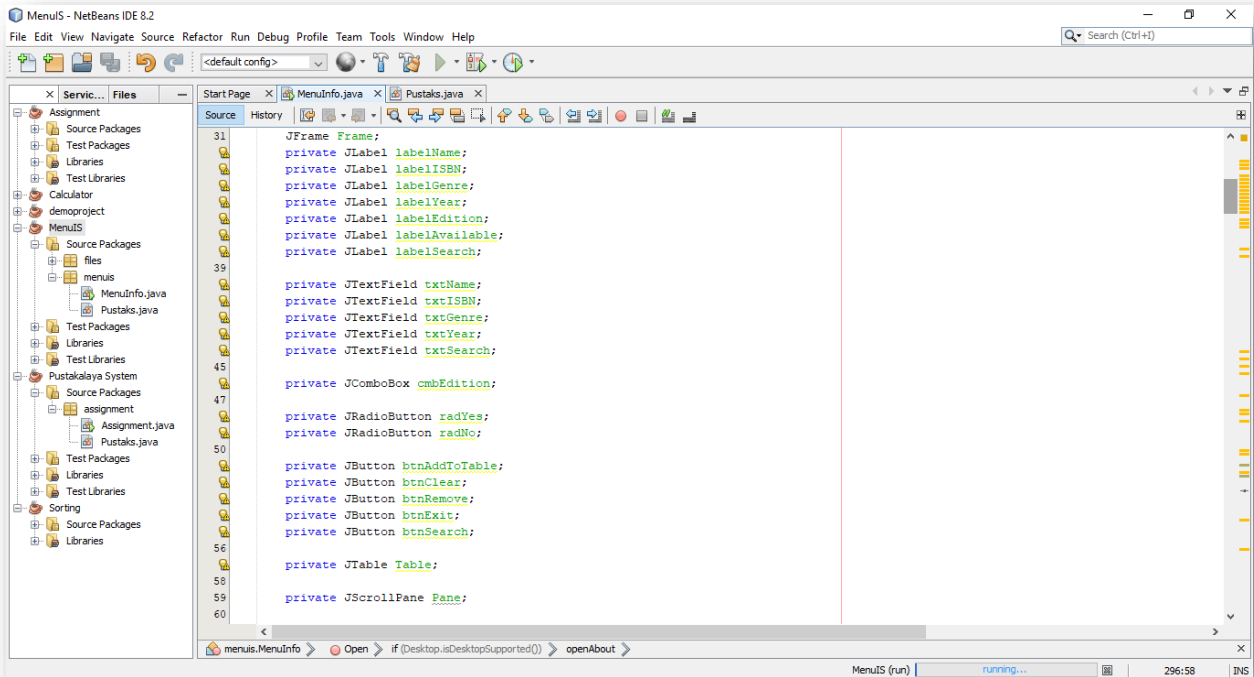




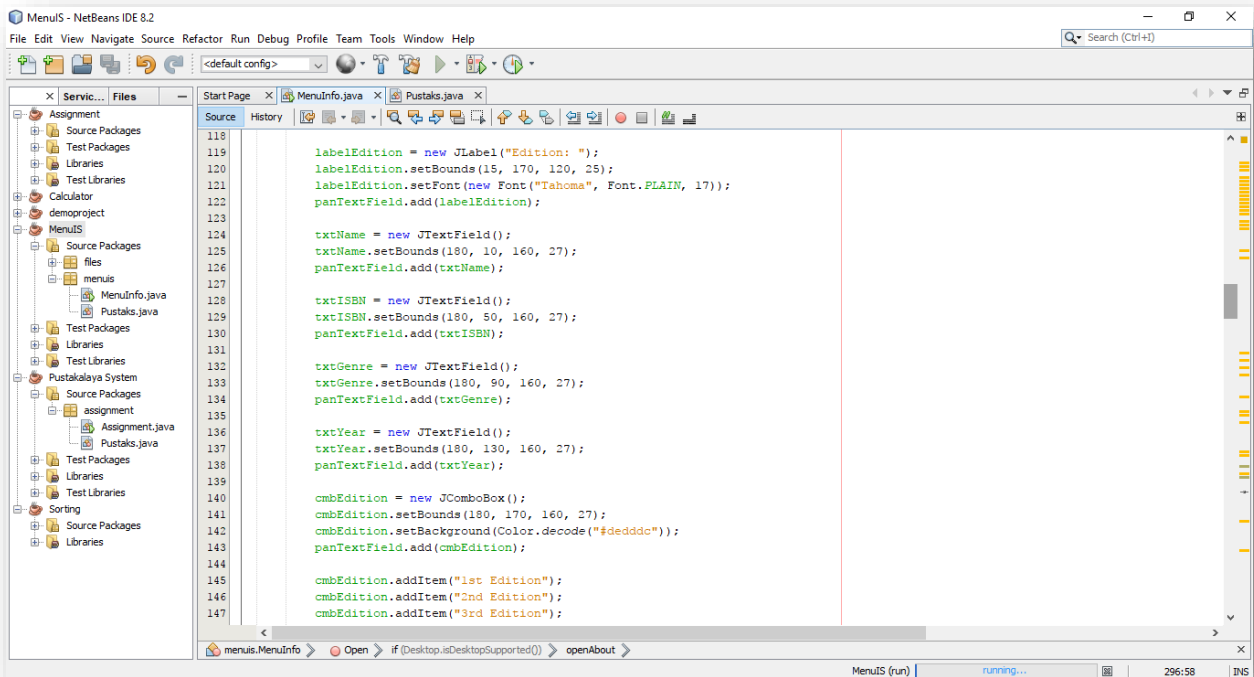
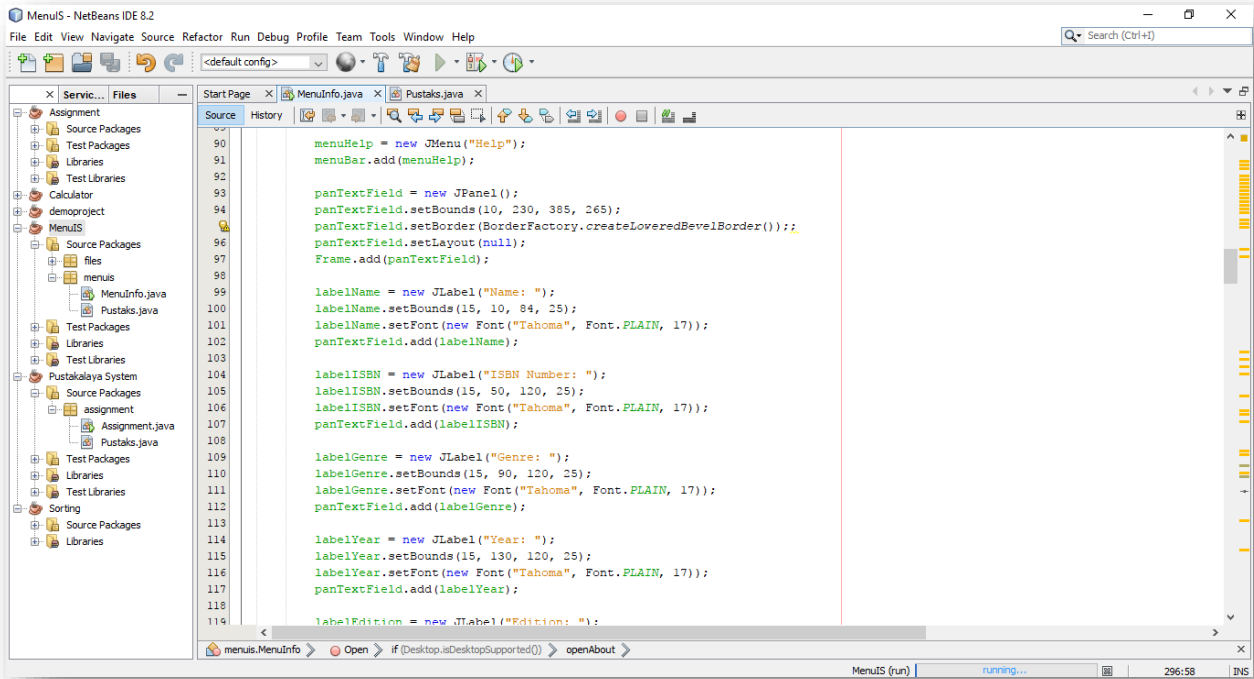
9.2 Source Code



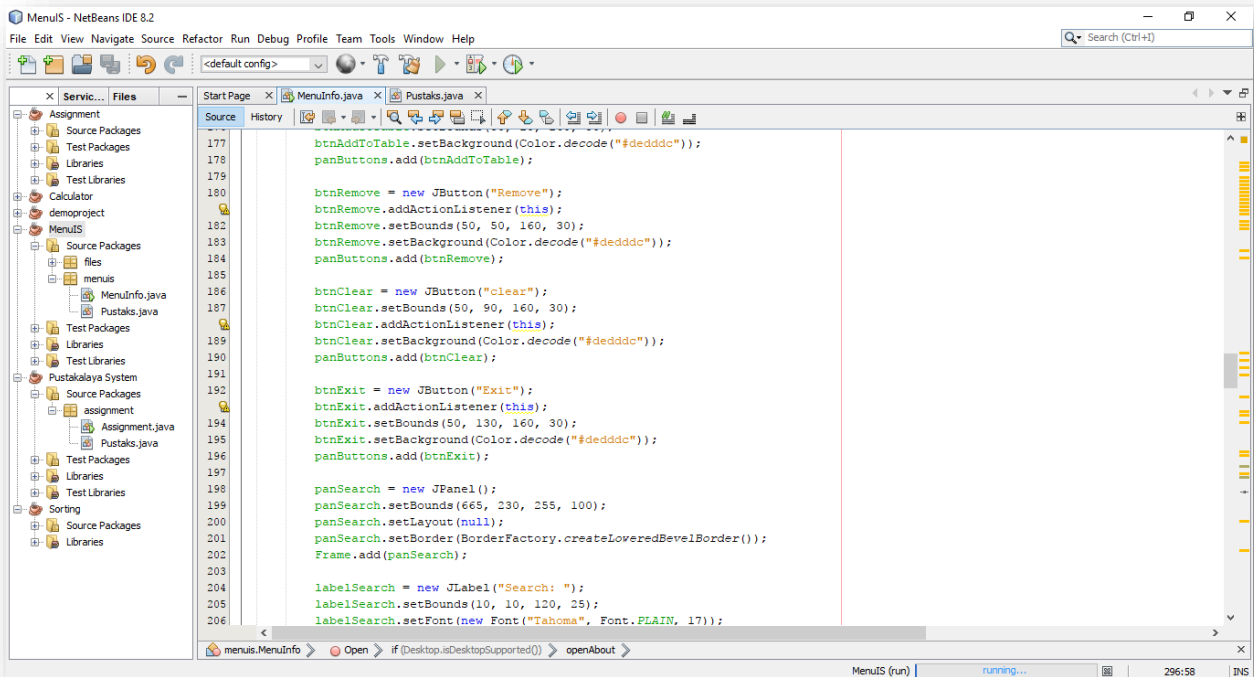
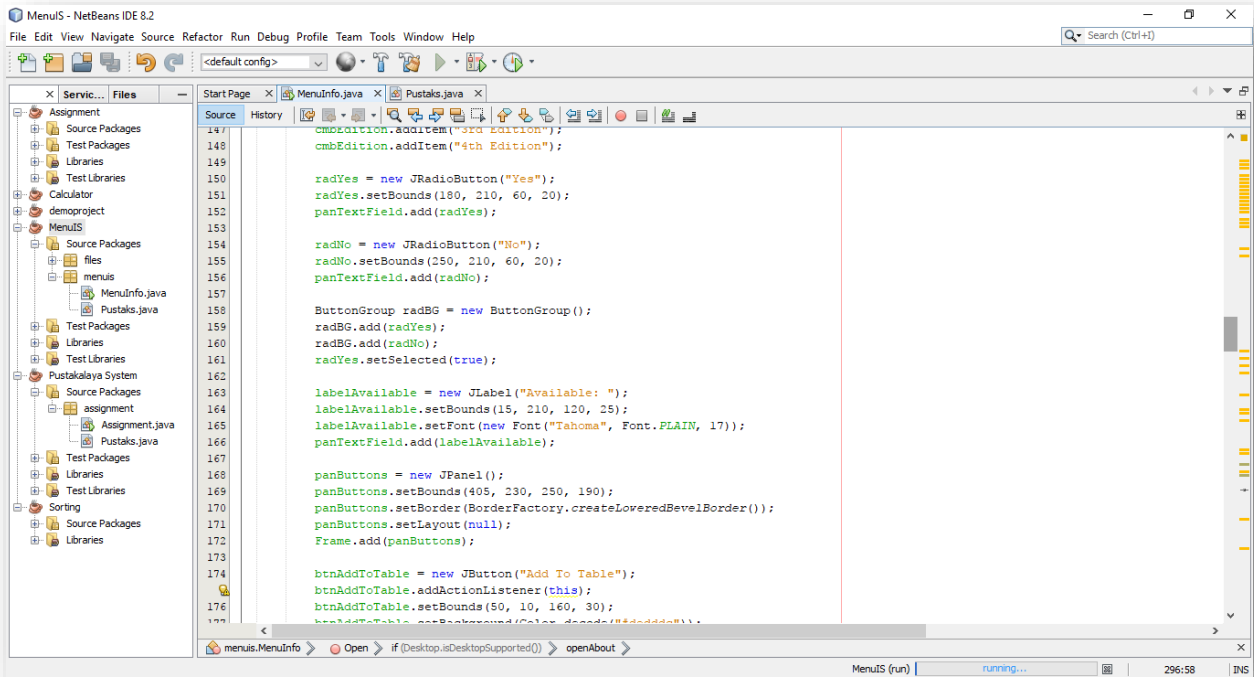
Emerging Programming Platforms and Technologies CS5004NI



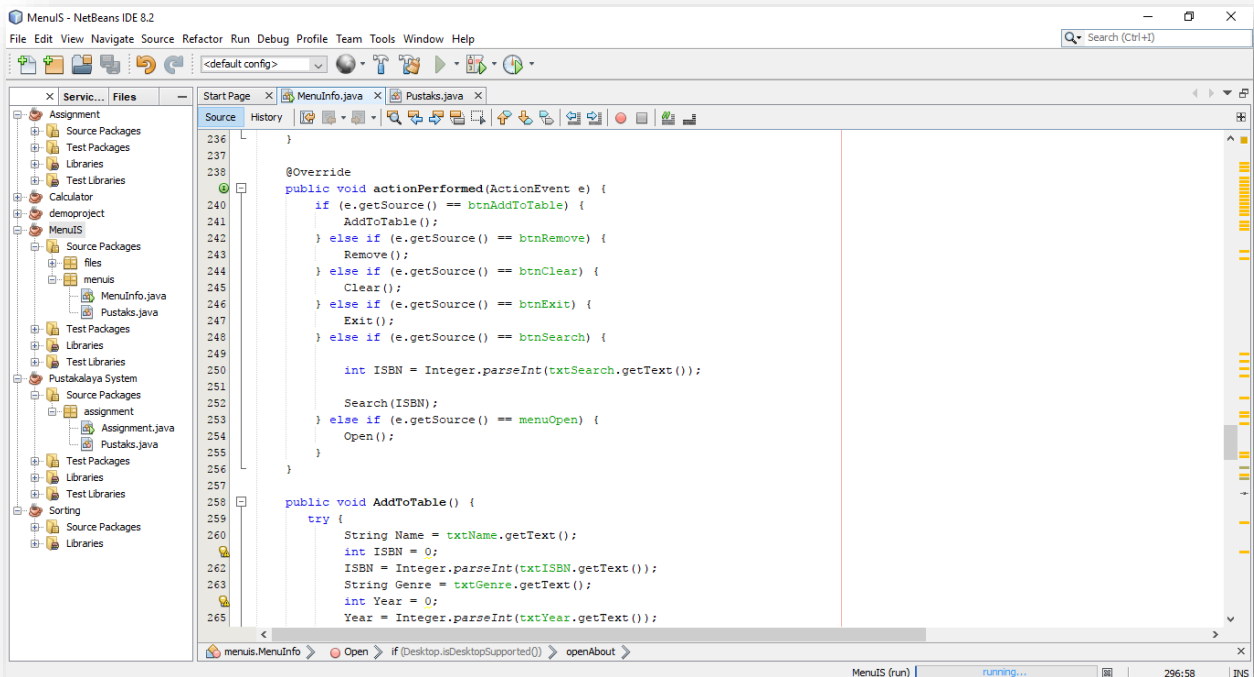
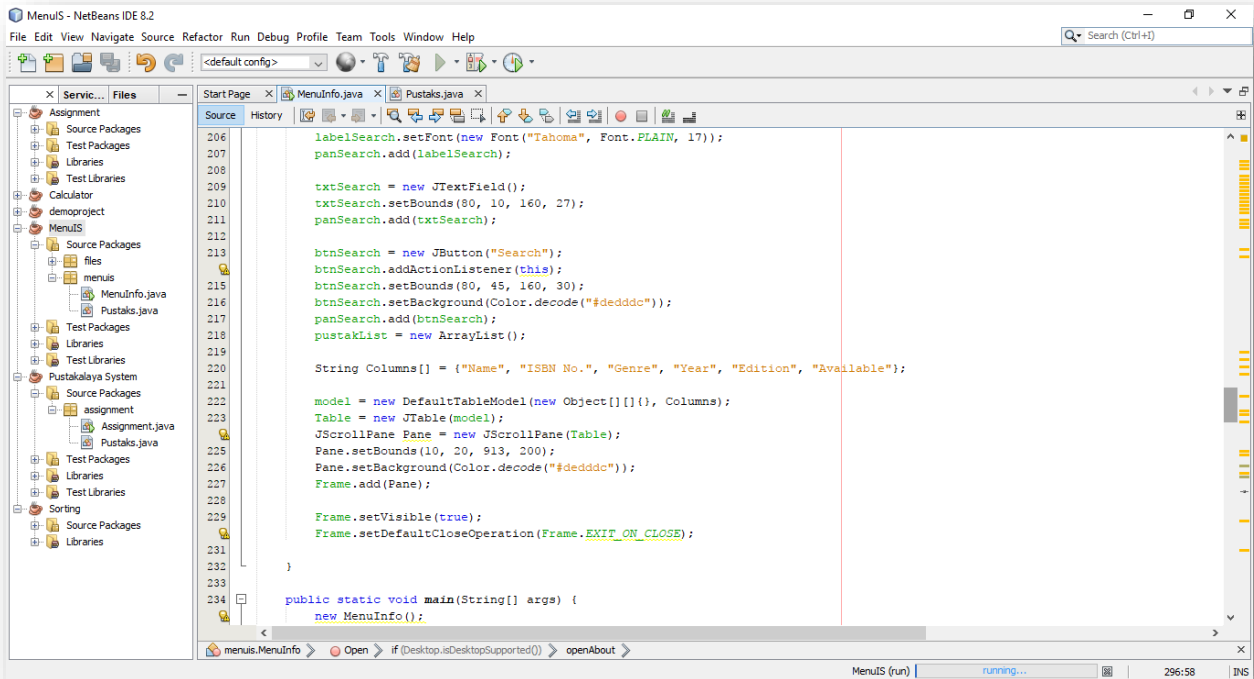
Emerging Programming Platforms and Technologies CS5004NI



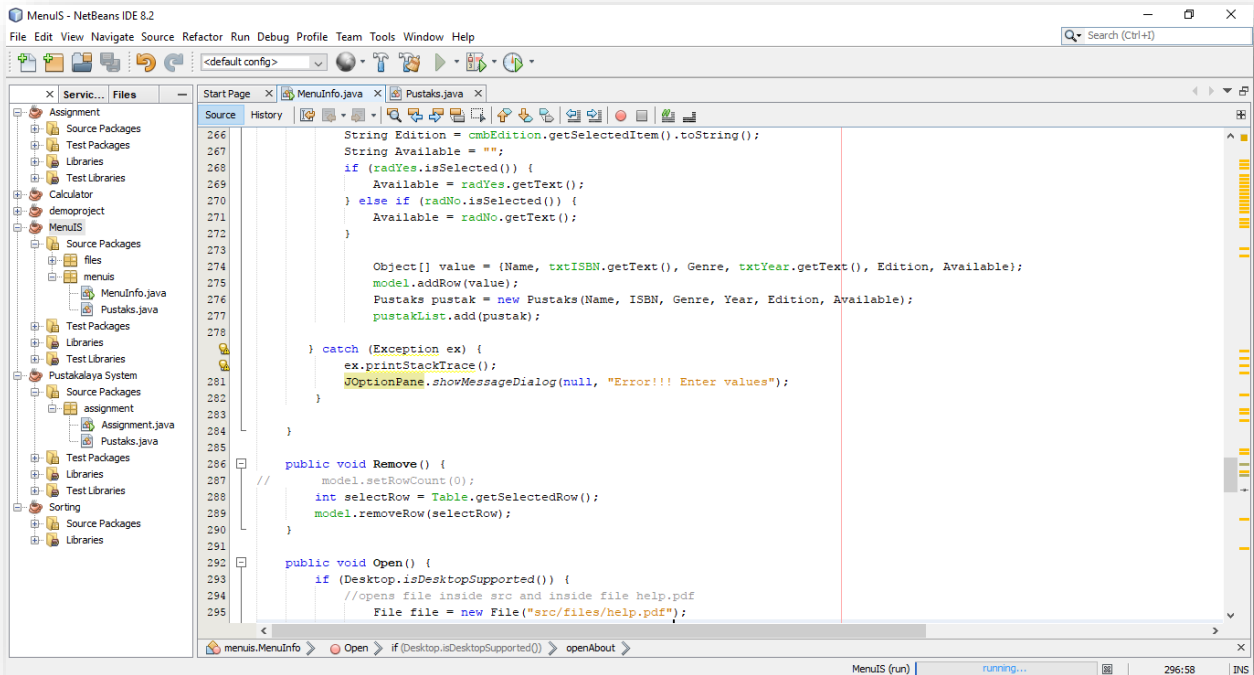
Emerging Programming Platforms and Technologies CS5004NI



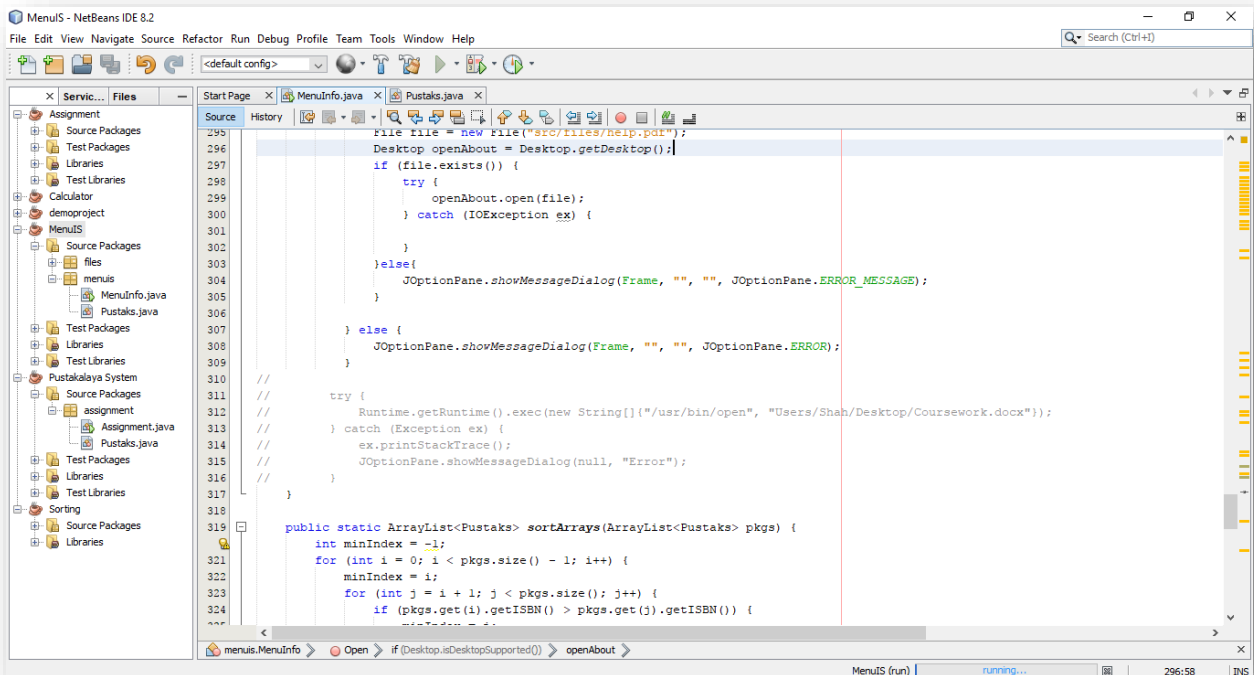
Emerging Programming Platforms and Technologies CS5004NI



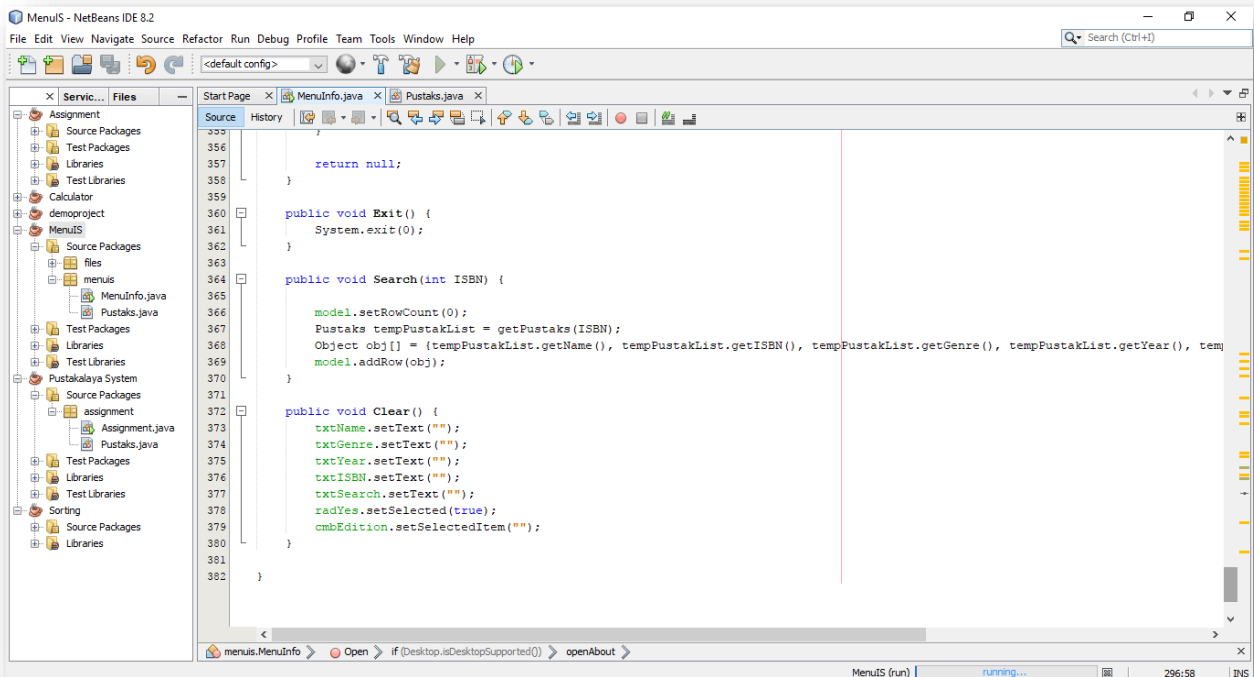
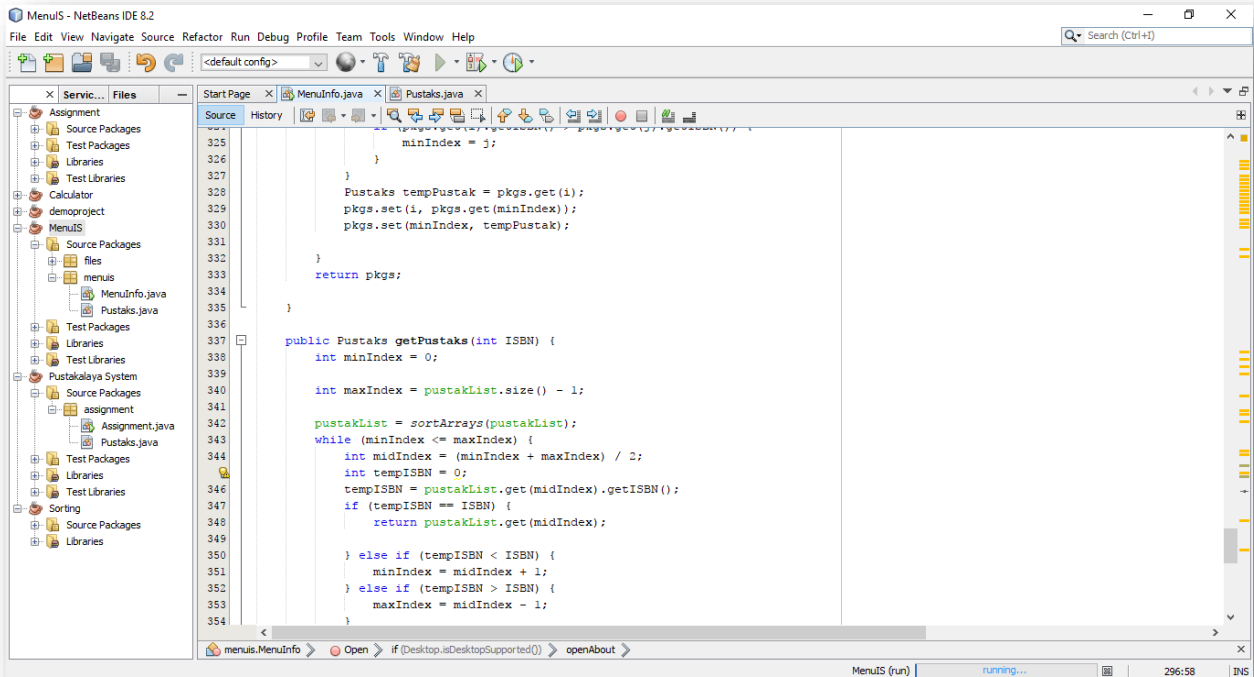
Emerging Programming Platforms and Technologies CS5004NI



```
266 String Edition = cmbEdition.getSelectedItem().toString();
267 String Available = "";
268 if (radYes.isSelected()) {
269     Available = radYes.getText();
270 } else if (radNo.isSelected()) {
271     Available = radNo.getText();
272 }
273
274 Object[] value = {Name, txtISBN.getText(), Genre, txtYear.getText(), Edition, Available};
275 model.addRow(value);
276 Pustaks pustak = new Pustaks(Name, ISBN, Genre, Year, Edition, Available);
277 pustakList.add(pustak);
278
279 } catch (Exception ex) {
280     ex.printStackTrace();
281     JOptionPane.showMessageDialog(null, "Error!!! Enter values");
282 }
283
284 }
285
286 public void Remove() {
287     //
288     model.setRowCount(0);
289     int selectRow = Table.getSelectedRow();
290     model.removeRow(selectRow);
291 }
292
293 public void Open() {
294     if (Desktop.isDesktopSupported()) {
295         //opens file inside src and inside file help.pdf
296         File file = new File("src/files/help.pdf");
297         Desktop.open(file);
298     }
299 }
```



```
295 File file = new File("src/files/help.pdf");
296 Desktop.open(file);
297 if (file.exists()) {
298     try {
299         openAbout.open(file);
300     } catch (IOException ex) {
301         JOptionPane.showMessageDialog(Frame, "", "", JOptionPane.ERROR_MESSAGE);
302     }
303 } else {
304     JOptionPane.showMessageDialog(Frame, "", "", JOptionPane.ERROR);
305 }
306
307 //
308 try {
309     Runtime.getRuntime().exec(new String[]{"usr/bin/open", "Users/Shah/Desktop/Coursework.docx"});
310 } catch (Exception ex) {
311     ex.printStackTrace();
312     JOptionPane.showMessageDialog(null, "Error");
313 }
314
315 public static ArrayList<Pustaks> sortArrays(ArrayList<Pustaks> pkgs) {
316     int minIndex = -1;
317     for (int i = 0; i < pkgs.size() - 1; i++) {
318         minIndex = i;
319         for (int j = i + 1; j < pkgs.size(); j++) {
320             if (pkgs.get(i).getISBN() > pkgs.get(j).getISBN()) {
321                 minIndex = j;
322             }
323         }
324         Pustaks temp = pkgs.get(i);
325         pkgs.set(i, pkgs.get(minIndex));
326         pkgs.set(minIndex, temp);
327     }
328     return pkgs;
329 }
```



Emerging Programming Platforms and Technologies CS5004NI

