JAVA PROGRAMMING

Assignment 2

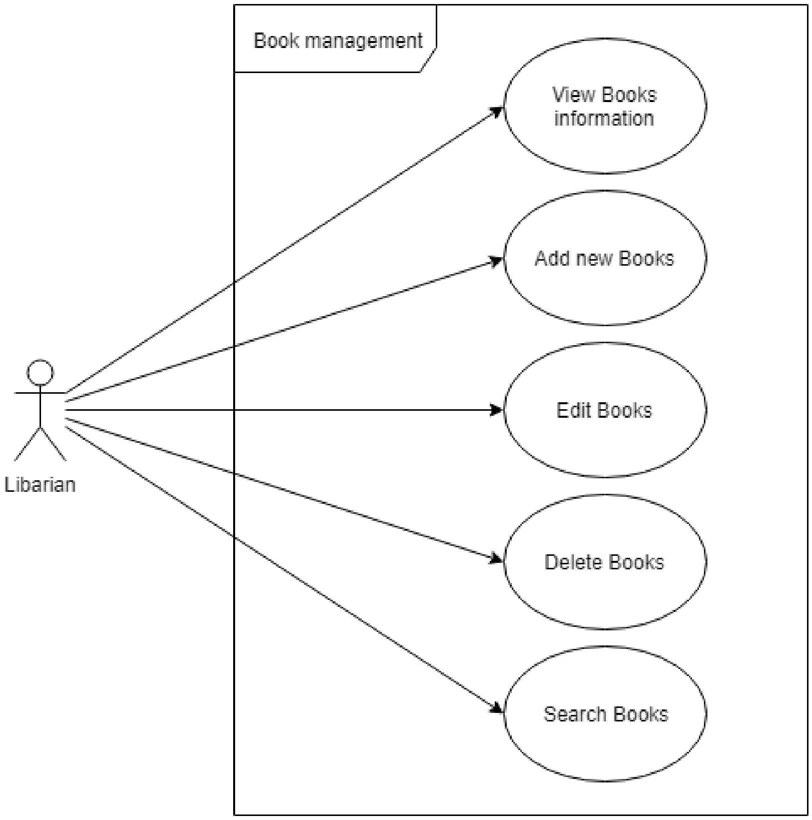
Duong Xuan Trung – GCH15332

Contents

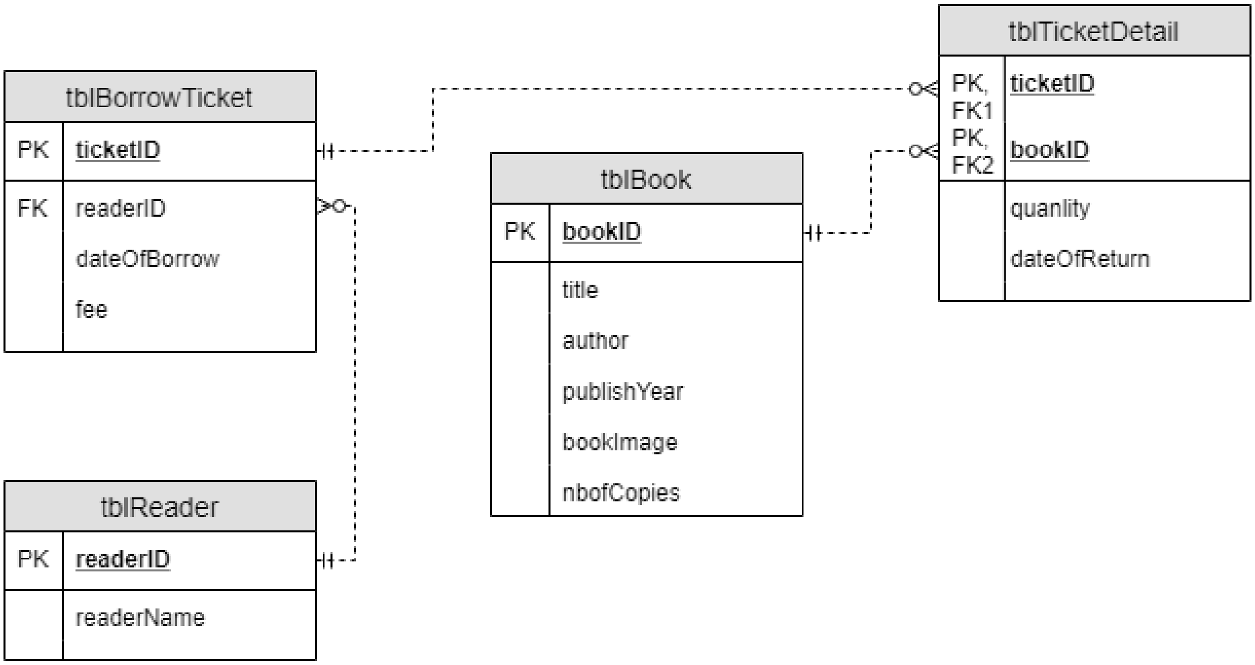
Task 1. Design Java solution

I. Introduction Java programming solution

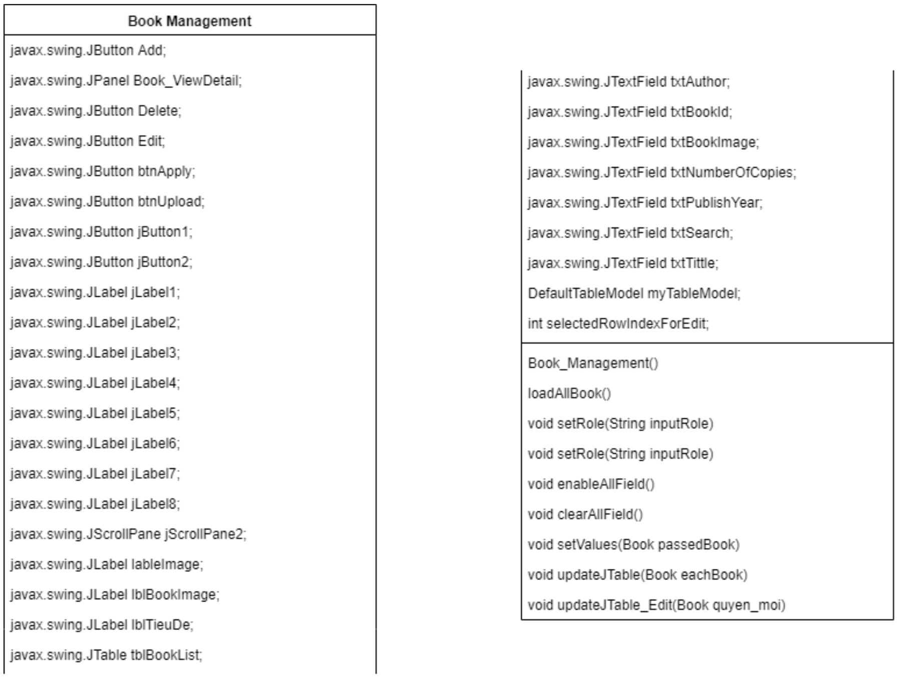
1. Use Case diagram



2. ERD/Database Diagram



3. Class Diagram



Book Management

javax.swing.JButton Add;
javax.swing.JPanel Book_ViewDetail;
javax.swing.JButton Delete;
javax.swing.JButton Edit;
javax.swing.JButton btnApply;
javax.swing.JButton btnUpload;
javax.swing.JButton jButton1;
javax.swing.JButton jButton2;
javax.swing.JLabel jLabel1;
javax.swing.JLabel jLabel2;
javax.swing.JLabel jLabel3;
javax.swing.JLabel jLabel4;
javax.swing.JLabel jLabel5;
javax.swing.JLabel jLabel6;
javax.swing.JLabel jLabel7;
javax.swing.JLabel jLabel8;
javax.swing.JScrollPane jScrollPane2;
javax.swing.JLabel lableImage;
javax.swing.JLabel lblBookImage;
javax.swing.JLabel lblTieuDe;
javax.swing.JTable tblBookList;
javax.swing.JTextField txtAuthor;
javax.swing.JTextField txtBookId;
javax.swing.JTextField txtBookImage;
javax.swing.JTextField txtNumberOfCopies;
javax.swing.JTextField txtPublishYear;
javax.swing.JTextField txtSearch;
javax.swing.JTextField txtTittle;
DefaultTableModel myTableModel;
int selectedRowIndexForEdit;

Book_Management()
loadAllBook()
void setRole(String inputRole)
void setRole(String inputRole)
void enableAllField()
void clearAllField()
void setValues(Book passedBook)
void updateJTable(Book eachBook)
void updateJTable_Edit(Book quyen_moi)

II. The components and data and file structures required to implement a give design

a. MVC

MVC stands for Model – View – Control is a design or software architectural model used in software engineering. For easy to understand, it divide source code into 3 parts, each component has a separate and independent task with other components.

1. Model

This is a component that contains all of the logic profession, processing method, retrieving database, object describing data such as functions, class processing.

2. View

View ensures the display of page information, interacting with the user, which contains all GUI objects such as images, TextBox. It is understandable to understand the set of HTML files and forms.

3. Control

Control has the task of navigating the requirements from the user and invoking the methods of processing them, such as receiving request from the form and the URL for direct operation with the Model component.

- The advantages of MVC: Show the professionalism in programming and analysis of subjects, because of the separate components should be independently active to help develop the application faster, simpler and easy to upgrade , more maintenance.
- The disadvantages of MVC: is that working models are very articles so with small applications, using MVC is very much time consuming and causing much complexity.

b. Database

Database is a widely used term in information technology, data, software programming, website programming... Database is a database, commonly used data such as tables, schemes, queries, reports, and other objects... will be gathered in an organization called database, news database. Database exists as a file in the computer operating system or database management systems, and is stored on a device that remembers hard drive, CD, or USB. The database has a very important role in working with data, without a database, the connection operation will not be able to successfully execute it.

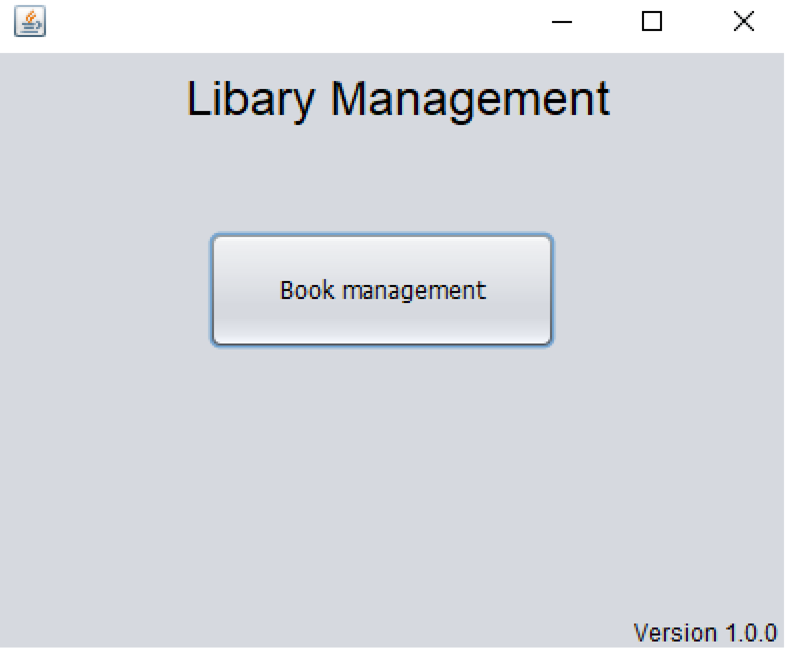
For example: you do the school's letter department, the principals ask you to list a well-studied poor student, but in the office not enough material on poor students, so you may not be able to access the text as required by the principal. The role of the database can also understand the same meaning.

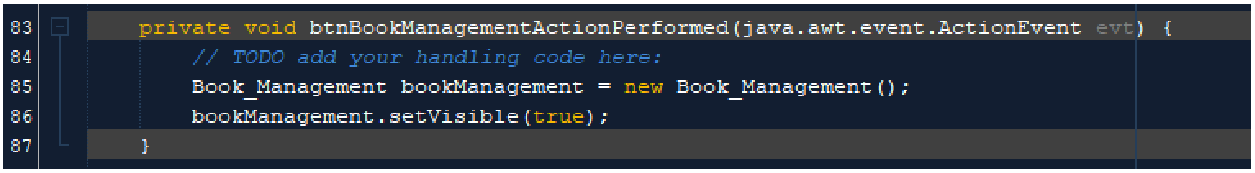
This program are using **SQL Server Management Studio 2008** for store information. **SQL Server Management Studio 2008** is a free integrated environment that supports accessibility, configuration, management, implementation and development of all components of SQL server, as well as combining graphic tools and rich scripting editors, provide developers and administrators access to SQL server. Microsoft has developed the application programming version of Microsoft SQL Server 2008 to support multiple applications, providing an intuitive interface so that users can write the code and run them directly, using Microsoft SQL Server 2008 to perform the authorization the database of each user is easily simple. SQL Server Management Studio 2008 offers many custom options but still supports fast adjustment features. For example, the program has the ability to change query parameters, so that the results and execution parameters will review all requirements of the server administrator.

Task 2

I. Introduce Library manager

a. Main



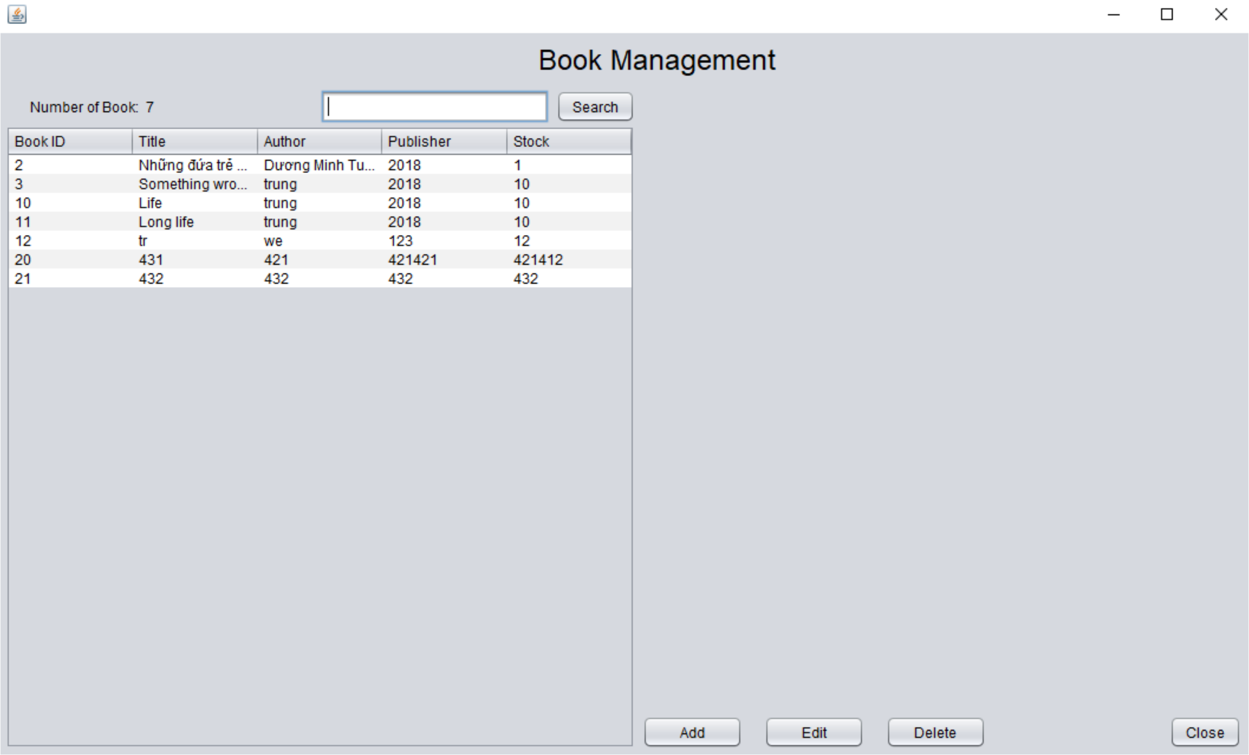


```java
    private void btnBookManagementActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        Book_Management bookManagement = new Book_Management();
        bookManagement.setVisible(true);
    }
```

The main frame will allow user to select function of the program then redirect user to the function frame. At this moment, there are just only one but in future the will be few move such as borrow ticket and manage reader

b. Book management

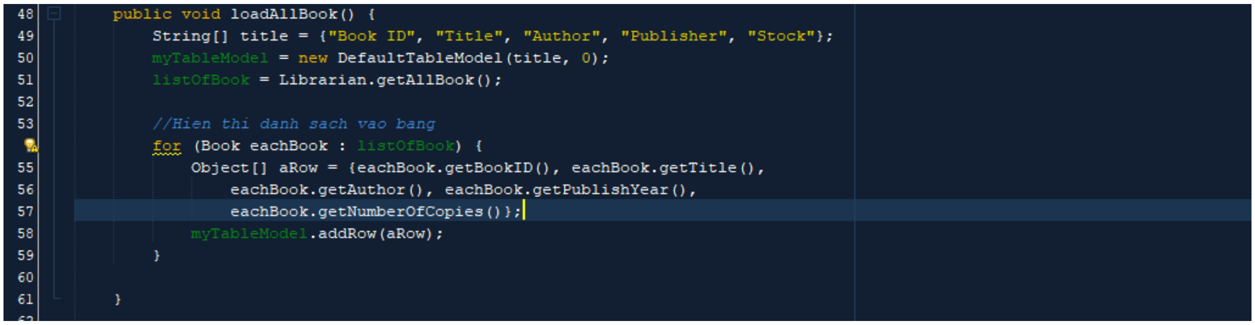
- Main book management frame



Book Management

Number of Book: 7

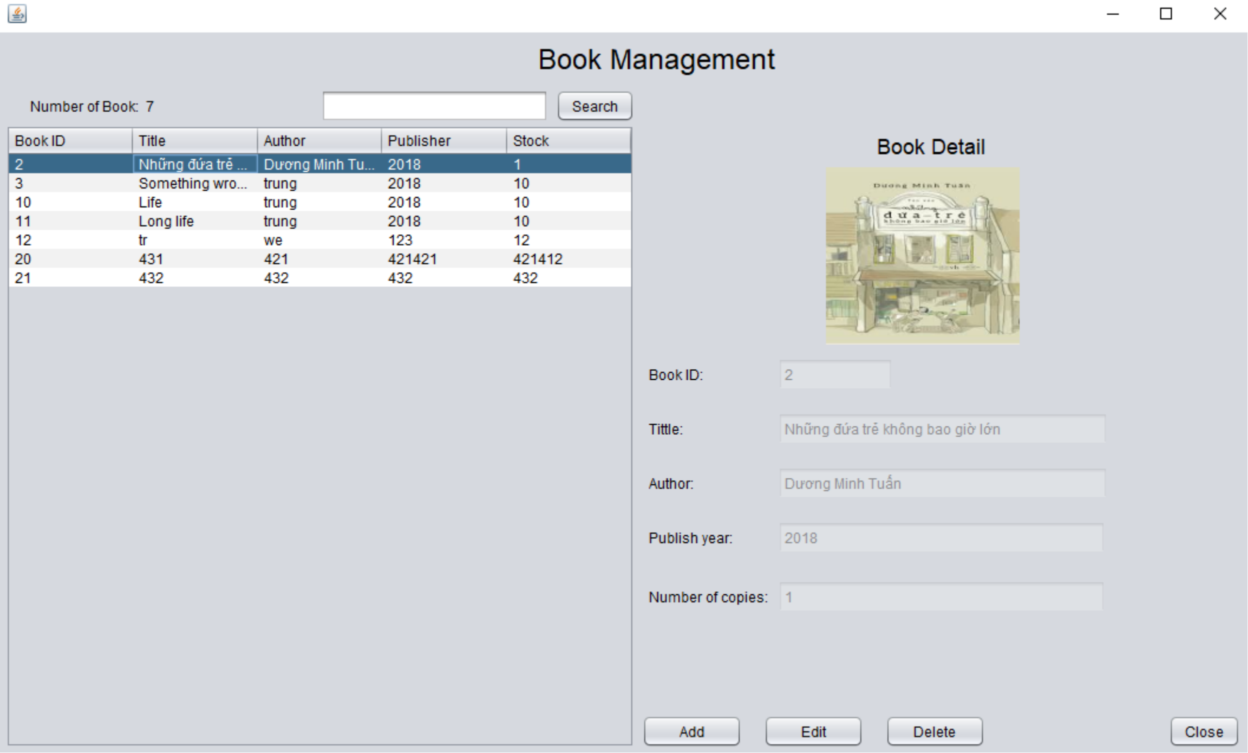
Book ID	Title	Author	Publisher	Stock
2	Những đứa trẻ ...	Dương Minh Tu...	2018	1
3	Something wro...	trung	2018	10
10	Life	trung	2018	10
11	Long life	trung	2018	10
12	tr	we	123	12
20	431	421	421421	421412
21	432	432	432	432



```
    public void loadAllBook() {
        String[] title = {"Book ID", "Title", "Author", "Publisher", "Stock"};
        myTableModel = new DefaultTableModel(title, 0);
        listOfBook = Librarian.getAllBook();

        //Hien thi danh sach vao bang
        for (Book eachBook : listOfBook) {
            Object[] aRow = {eachBook.getBookID(), eachBook.getTitle(),
                eachBook.getAuthor(), eachBook.getPublishYear(),
                eachBook.getNumberOfCopies()};
            myTableModel.addRow(aRow);
        }

    }
```

The main manager book allow user to see the list of the books and also allow them to Add new ones, Edit the current book, remove book from the list and also search for them

- View



Book Management

Number of Book: 7

Book ID	Title	Author	Publisher	Stock
2	Những đứa trẻ ...	Dương Minh Tu...	2018	1
3	Something wro...	trung	2018	10
10	Life	trung	2018	10
11	Long life	trung	2018	10
12	tr	we	123	12
20	431	421	421421	421412
21	432	432	432	432

Book Detail

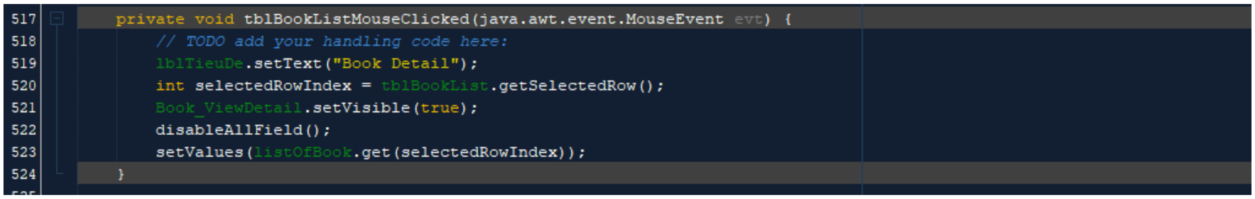
Book ID: 2

Tittle: Những đứa trẻ không bao giờ lớn

Author: Dương Minh Tuấn

Publish year: 2018

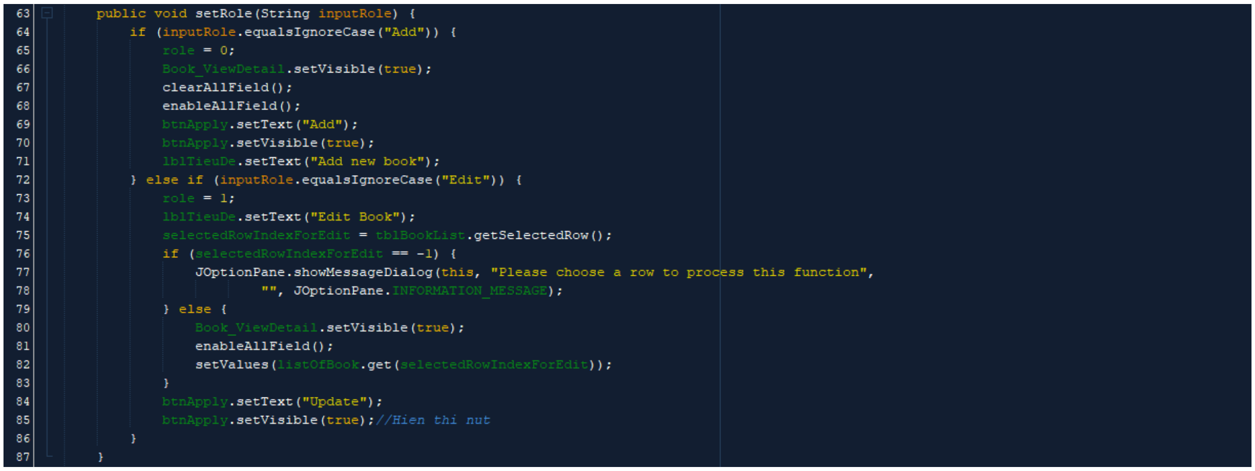
Number of copies: 1



```java
    private void tblBookListMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        lblTieuDe.setText("Book Detail");
        int selectedRowIndex = tblBookList.getSelectedRow();
        Book_ViewDetail.setVisible(true);
        disableAllField();
        setValues(listOfBook.get(selectedRowIndex));
    }
```

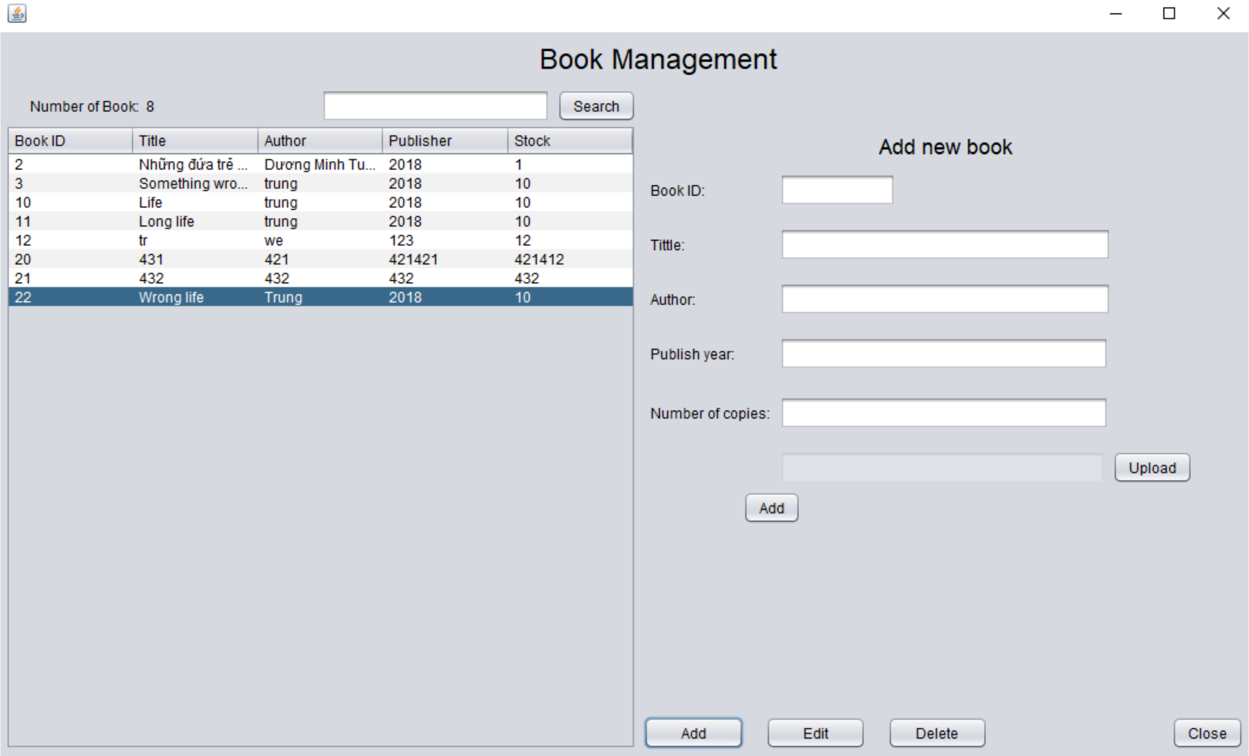
User can view book information by select the book on list.

Before look at functions Add, Edit. Both function are set-up with there own name.



```java
    public void setRole(String inputRole) {
        if (inputRole.equalsIgnoreCase("Add")) {
            role = 0;
            Book_ViewDetail.setVisible(true);
            clearAllField();
            enableAllField();
            btnApply.setText("Add");
            btnApply.setVisible(true);
            lblTieuDe.setText("Add new book");
        } else if (inputRole.equalsIgnoreCase("Edit")) {
            role = 1;
            lblTieuDe.setText("Edit Book");
            selectedRowIndexForEdit = tblBookList.getSelectedRow();
            if (selectedRowIndexForEdit == -1) {
                JOptionPane.showMessageDialog(this, "Please choose a row to process this function",
                        "", JOptionPane.INFORMATION_MESSAGE);
            } else {
                Book_ViewDetail.setVisible(true);
                enableAllField();
                setValues(listOfBook.get(selectedRowIndexForEdit));
            }
            btnApply.setText("Update");
            btnApply.setVisible(true);//Hien thi nut
        }
    }
```

- Add new book

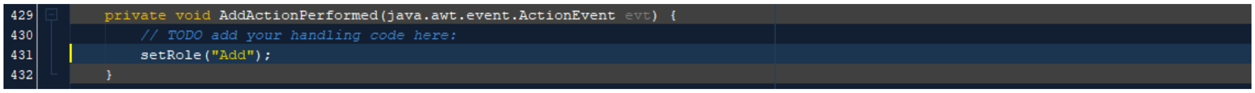


Book Management

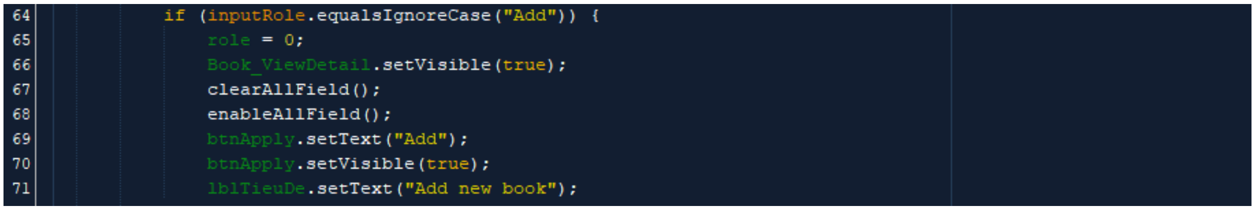
Number of Book: 8

Book ID	Title	Author	Publisher	Stock
2	Những đứa trẻ ...	Dương Minh Tu...	2018	1
3	Something wro...	trung	2018	10
10	Life	trung	2018	10
11	Long life	trung	2018	10
12	tr	we	123	12
20	431	421	421421	421412
21	432	432	432	432
22	Wrong life	Trung	2018	10

Add new book



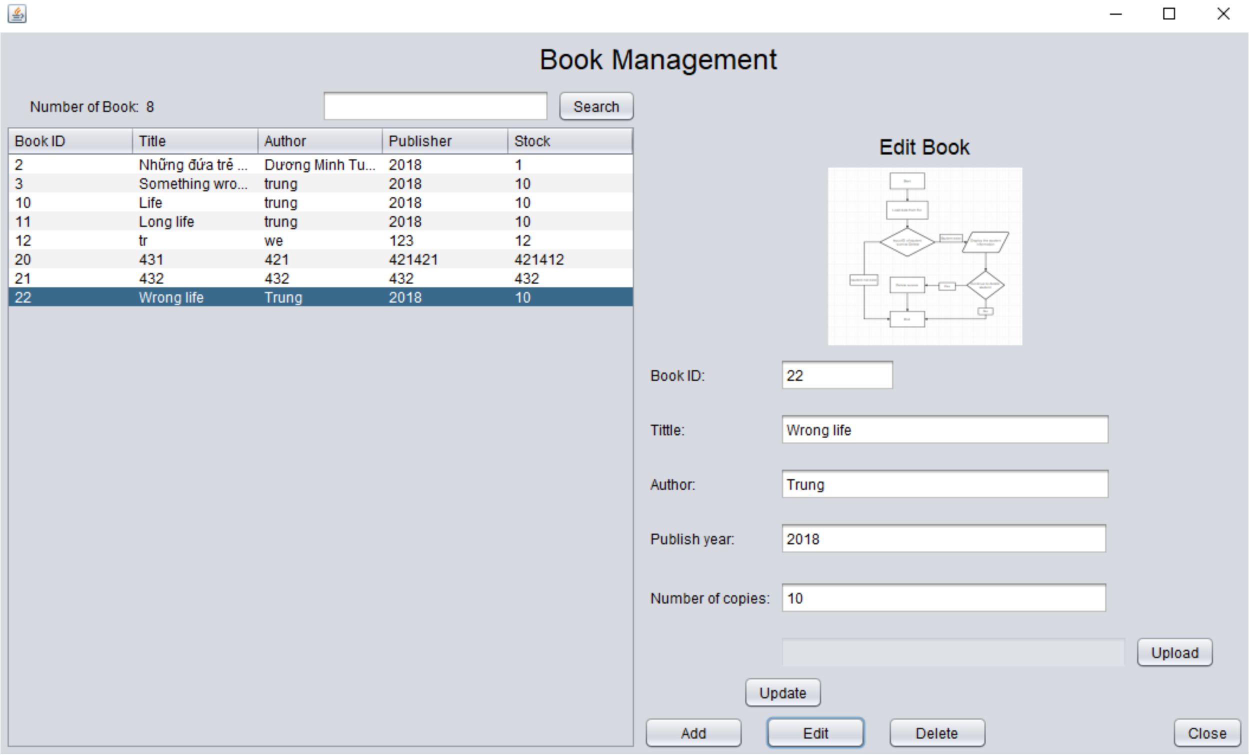
```
    private void AddActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        setRole("Add");
    }
```



```
            if (inputRole.equalsIgnoreCase("Add")) {
                role = 0;
                Book_ViewDetail.setVisible(true);
                clearAllField();
                enableAllField();
                btnApply.setText("Add");
                btnApply.setVisible(true);
                lblTieuDe.setText("Add new book");
```

The function Add will allow user to add new book to list. After click on “Add” button all the field will be clear and available for user to input the new one, also the apply button will appear. Therefore, after input text, user can select apply to input the new one to list.

- Edit book



Number of Book: 8

Book ID	Title	Author	Publisher	Stock
2	Những đứa trẻ ...	Dương Minh Tu...	2018	1
3	Something wro...	trung	2018	10
10	Life	trung	2018	10
11	Long life	trung	2018	10
12	tr	we	123	12
20	431	421	421421	421412
21	432	432	432	432
22	Wrong life	Trung	2018	10

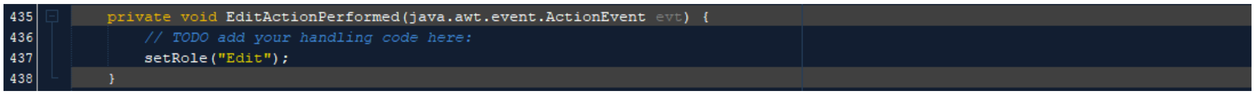
Book ID: 22

Tittle: Wrong life

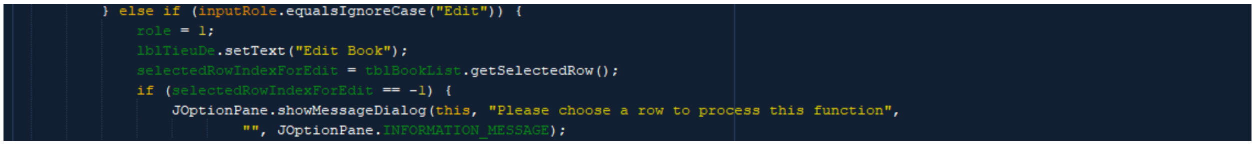
Author: Trung

Publish year: 2018

Number of copies: 10



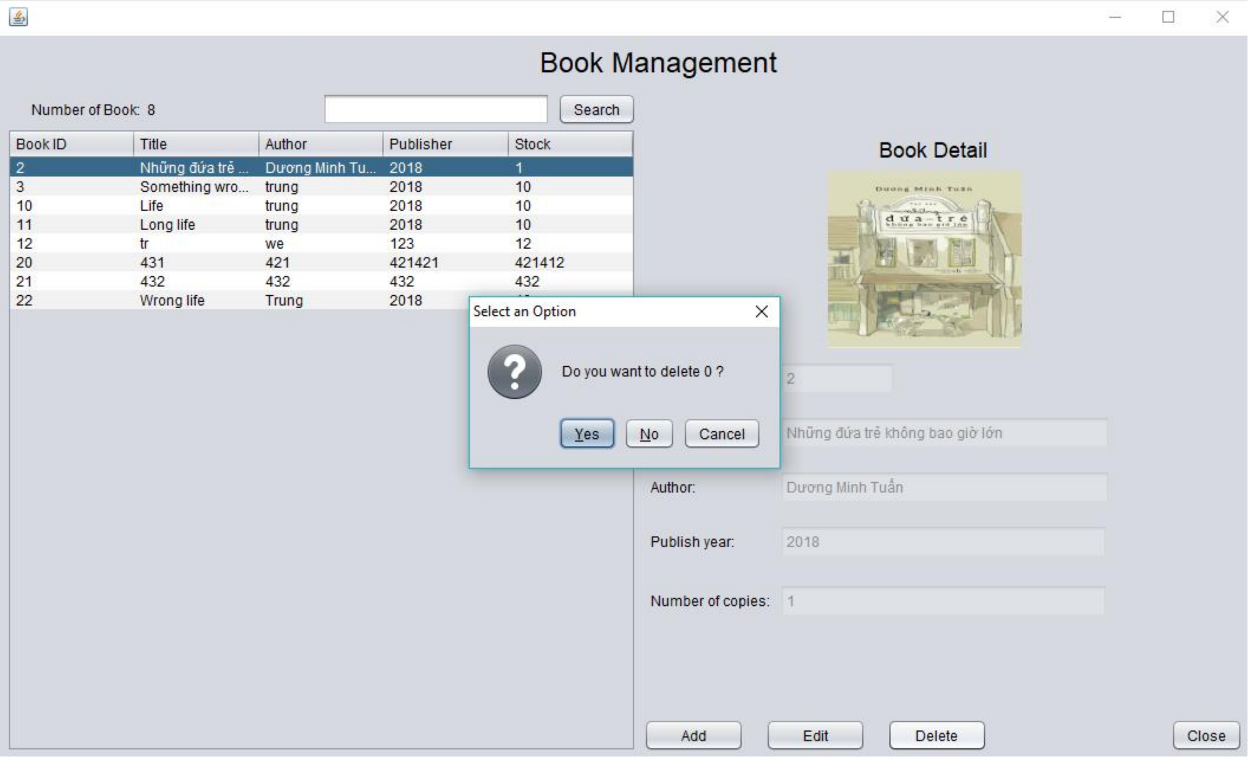
```
private void EditActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    setRole("Edit");
}
```



```
        } else if (inputRole.equalsIgnoreCase("Edit")) {
            role = 1;
            lblTieuDe.setText("Edit Book");
            selectedRowIndexForEdit = tblBookList.getSelectedRow();
            if (selectedRowIndexForEdit == -1) {
                JOptionPane.showMessageDialog(this, "Please choose a row to process this function",
                        "", JOptionPane.INFORMATION_MESSAGE);
```

Edit function will get all the information from the selected row then all the field are also available for user to make change. The Update button will appear right at the Apply button of add function location which after make change, user can select Update button for confirm the update.

- Delete



Book Management

Number of Book: 8

Book ID	Title	Author	Publisher	Stock
2	Những đứa trẻ ...	Dương Minh Tu...	2018	1
3	Something wro...	trung	2018	10
10	Life	trung	2018	10
11	Long life	trung	2018	10
12	tr	we	123	12
20	431	421	421421	421412
21	432	432	432	432
22	Wrong life	Trung	2018	[illegible]

Select an Option

Do you want to delete 0 ?

Book Detail

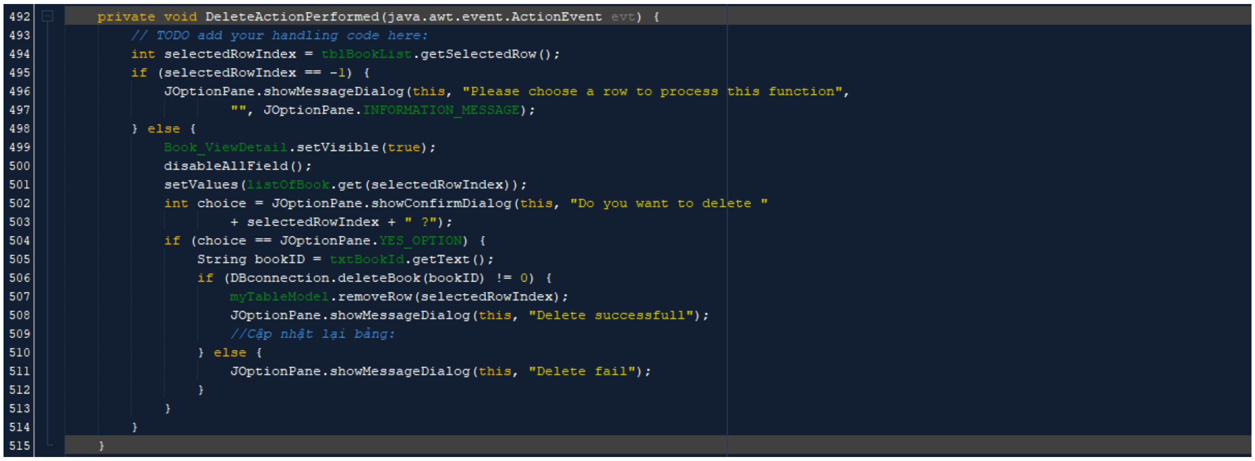
2

Những đứa trẻ không bao giờ lớn

Author: Dương Minh Tuấn

Publish year: 2018

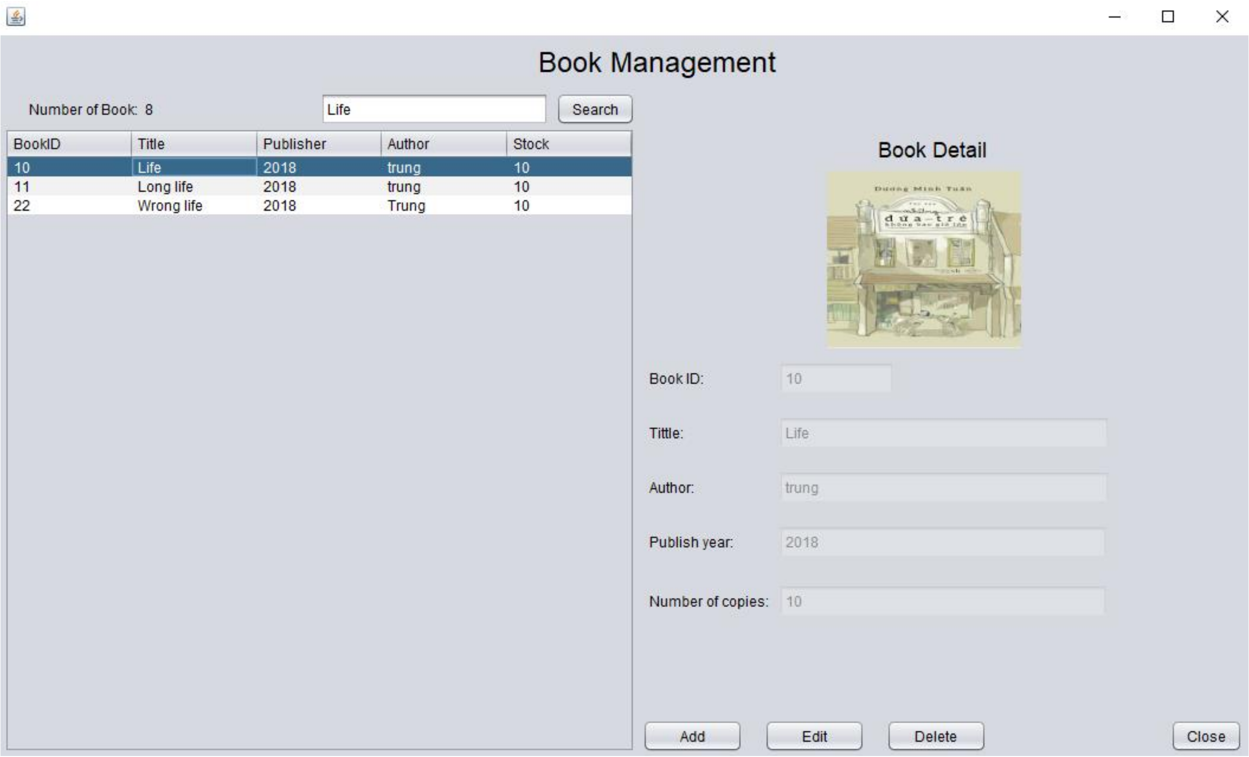
Number of copies: 1



```
private void DeleteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int selectedRowIndex = tblBookList.getSelectedRow();
    if (selectedRowIndex == -1) {
        JOptionPane.showMessageDialog(this, "Please choose a row to process this function",
                "", JOptionPane.INFORMATION_MESSAGE);
    } else {
        Book_ViewDetail.setVisible(true);
        disableAllField();
        setValues(listOfBook.get(selectedRowIndex));
        int choice = JOptionPane.showConfirmDialog(this, "Do you want to delete "
                + selectedRowIndex + " ?");
        if (choice == JOptionPane.YES_OPTION) {
            String bookID = txtBookId.getText();
            if (DBconnection.deleteBook(bookID) != 0) {
                myTableModel.removeRow(selectedRowIndex);
                JOptionPane.showMessageDialog(this, "Delete successfull");
                //Cập nhật lại bảng:
            } else {
                JOptionPane.showMessageDialog(this, "Delete fail");
            }
        }
    }
}
```

Selected this button, there will be a confirmation pop-up that confirm to delete the current selected.

- Search book by name



Book Management

Number of Book: 8

Life

BookID	Title	Publisher	Author	Stock
10	Life	2018	trung	10
11	Long life	2018	trung	10
22	Wrong life	2018	Trung	10

Book Detail

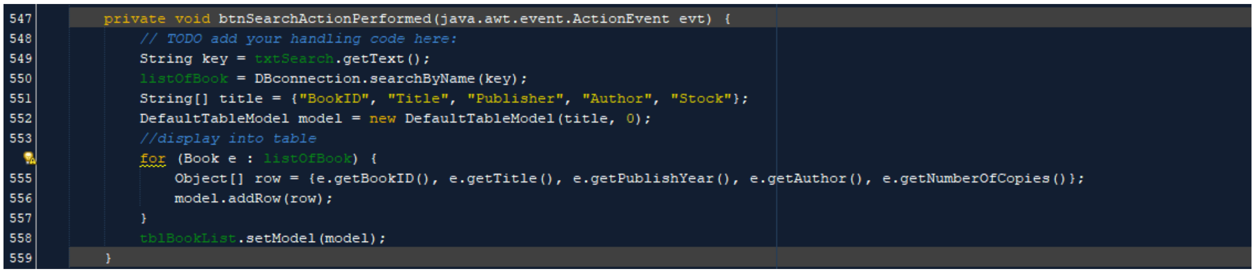
Book ID: 10

Tittle: Life

Author: trung

Publish year: 2018

Number of copies: 10



```
    private void btnSearchActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String key = txtSearch.getText();
        listOfBook = DBconnection.searchByName(key);
        String[] title = {"BookID", "Title", "Publisher", "Author", "Stock"};
        DefaultTableModel model = new DefaultTableModel(title, 0);
        //display into table
        for (Book e : listOfBook) {
            Object[] row = {e.getBookID(), e.getTitle(), e.getPublishYear(), e.getAuthor(), e.getNumberOfCopies()};
            model.addRow(row);
        }
        tblBookList.setModel(model);
    }
```

By input the book name, the search function will look through all book of the list then will show on the table all the books that contain the key word.

II. Code

- Set role for Add and Edit

```
public void setRole(String inputRole) {
    if (inputRole.equalsIgnoreCase("Add")) {
        role = 0;
```

```
      Book_ViewDetail.setVisible(true);
      clearAllField();
      enableAllField();
      btnApply.setText("Add");
      btnApply.setVisible(true);
      lblTieuDe.setText("Add new book");
    } else if (inputRole.equalsIgnoreCase("Edit")) {
      role = 1;
      lblTieuDe.setText("Edit Book");
      selectedRowIndexForEdit = tblBookList.getSelectedRow();
      if (selectedRowIndexForEdit == -1) {
        JOptionPane.showMessageDialog(this, "Please choose a row to process this function",
            "", JOptionPane.INFORMATION_MESSAGE);
      } else {
        Book_ViewDetail.setVisible(true);
        enableAllField();
        setValues(listOfBook.get(selectedRowIndexForEdit));
      }
      btnApply.setText("Update");
      btnApply.setVisible(true);//Hien thi nut
    }
  }
```

By set Role for Add and Edit, we have minimize the code more simple so other can read, understand, and edit more easier.

- Apply and update button

  ```
  private void btnApplyActionPerformed(java.awt.event.ActionEvent evt) {
      // TODO add your handling code here:
      if (role == 1) { //Edit
        String title = txtTittle.getText();
        String author = txtAuthor.getText();
        String publishYear = txtPublishYear.getText();
  ```

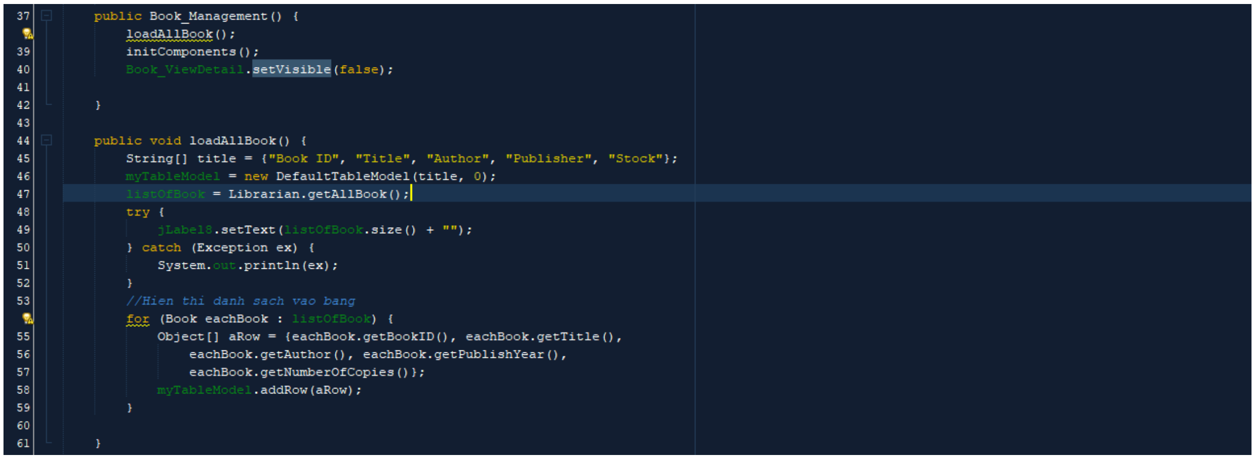
```
            String nbOfCopies = txtNumberOfCopies.getText();
            String bookImage = txtBookImage.getText();
            Book quyen_moi = new Book(quyen_cu.getBookID(), title, author,
Integer.parseInt(publishYear), bookImage, Integer.parseInt(nbOfCopies));
            if (Librarian.updateBook(quyen_cu, quyen_moi)) {
                updateJTable_Edit(quyen_moi);
                JOptionPane.showMessageDialog(this, "Update successfull");
                Book_ViewDetail.setVisible(false);
                disableAllField();
            } else {
                JOptionPane.showMessageDialog(this, "Edit fail");
            }
        } else if (role == 0) { //Add
            String tittle = txtTittle.getText();
            String author = txtAuthor.getText();
            String publishYear = txtPublishYear.getText();
            String nbOfCopies = txtNumberOfCopies.getText();
            String bookImage = txtBookImage.getText();
            Book inputBook = new Book(tittle, author,
                    Integer.parseInt(publishYear),
                    bookImage,
                    Integer.parseInt(nbOfCopies));
            int ID_MOI_SINH = Librarian.addBook(inputBook);

            if (ID_MOI_SINH != 0) {
                JOptionPane.showMessageDialog(this, "Adding successfully!");
                //Cap nhat gia tri moi vao JTable
                inputBook.setBookID(ID_MOI_SINH);
                try {
                    updateJTable(inputBook);
                } catch (Exception ex) {
                    System.out.println(ex);
                }
                Book_ViewDetail.setVisible(false);
                disableAllField();
            } else {
                JOptionPane.showMessageDialog(this, "An error appears. "
                        + "Adding unsuccessfully!");
            }
        }
    }
```

The apply and update button are simple one button which change based on the function that user selected. If user select Add the button Apply will appear, but if select Edit the Update button will appear. By set role for each of the function Role 0 is add and Edit is 1, the button will base on that role to display Apply or Update.

III. Error handling, Data validation, and JOptionPane

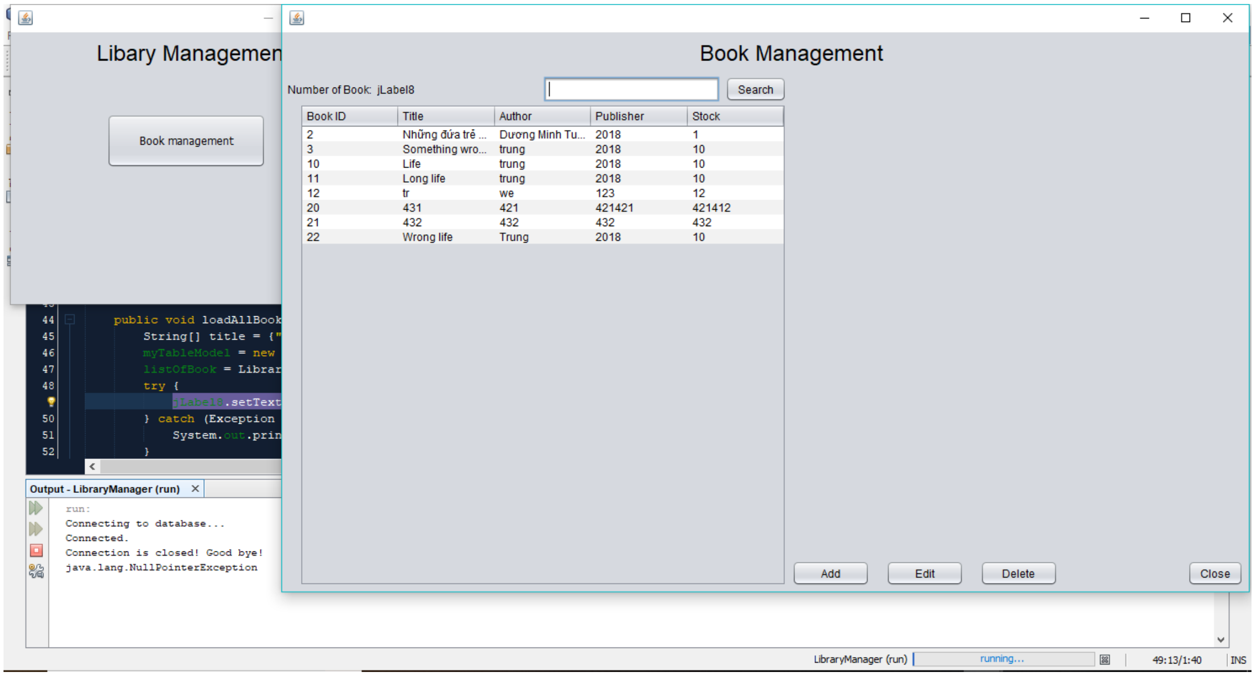
a. Error handling

To make sure the program running smooth, some specific code are putted into try/catch.



```java
    public Book_Management() {
        loadAllBook();
        initComponents();
        Book_ViewDetail.setVisible(false);

    }

    public void loadAllBook() {
        String[] title = {"Book ID", "Title", "Author", "Publisher", "Stock"};
        myTableModel = new DefaultTableModel(title, 0);
        listOfBook = Librarian.getAllBook();
        try {
            jLabel8.setText(listOfBook.size() + "");
        } catch (Exception ex) {
            System.out.println(ex);
        }
        //Hien thi danh sach vao bang
        for (Book eachBook : listOfBook) {
            Object[] aRow = {eachBook.getBookID(), eachBook.getTitle(),
                eachBook.getAuthor(), eachBook.getPublishYear(),
                eachBook.getNumberOfCopies()};
            myTableModel.addRow(aRow);
        }

    }
```

The jLabel8.setText(listOfBook.size() + ""); are putted inside try/catch so when we run the program:



Number of Book: jLabel8

Book ID	Title	Author	Publisher	Stock
2	Những đứa trẻ ...	Dương Minh Tu...	2018	1
3	Something wro...	trung	2018	10
10	Life	trung	2018	10
11	Long life	trung	2018	10
12	tr	we	123	12
20	431	421	421421	421412
21	432	432	432	432
22	Wrong life	Trung	2018	10

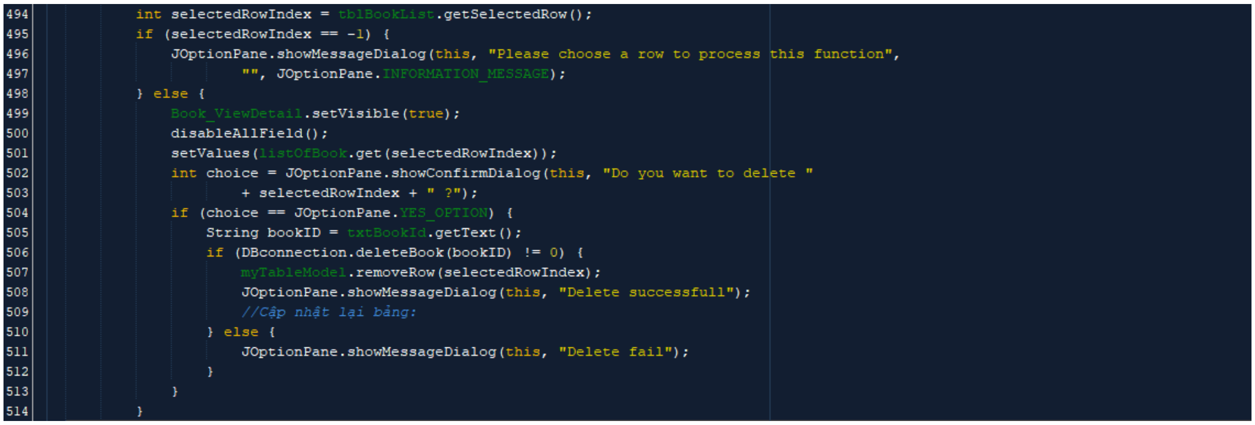
Output - LibraryManager (run)

```
run:
Connecting to database...
Connected.
Connection is closed! Good bye!
java.lang.NullPointerException
```

If there are any error it will appear the error but the program still running.

b. Data validation

If/else is also very important:

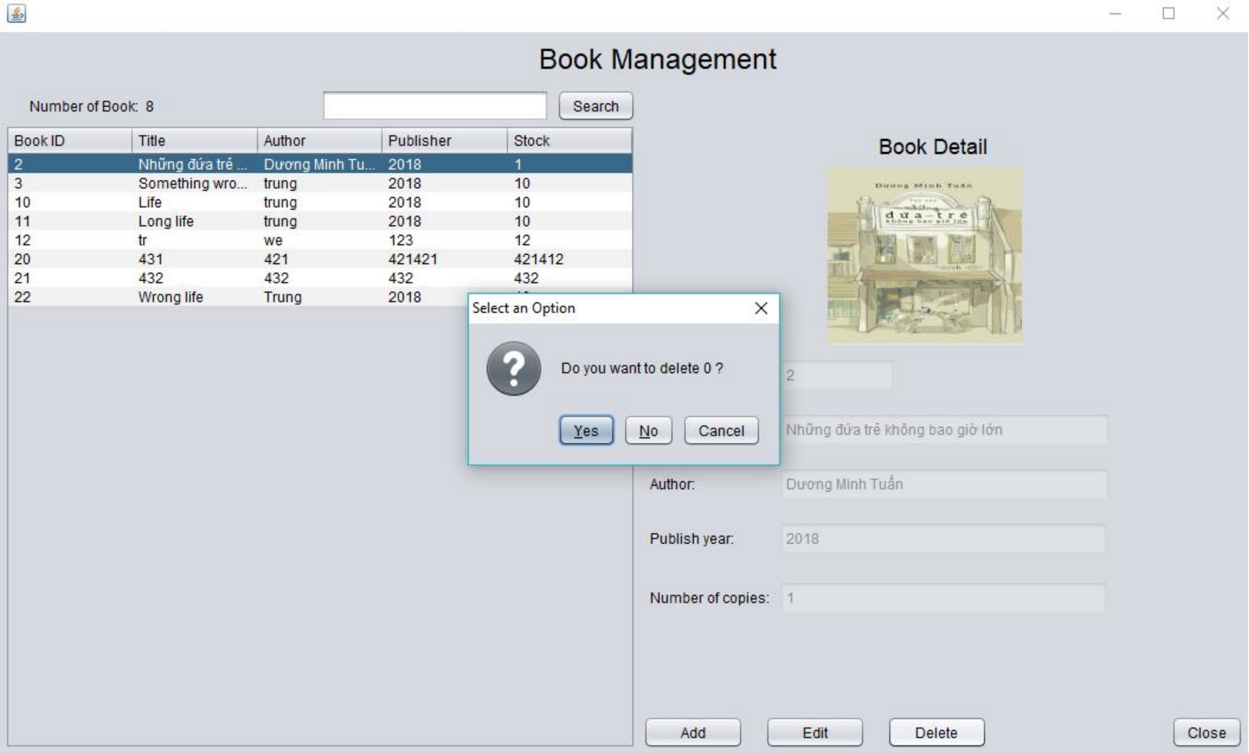


```java
int selectedRowIndex = tblBookList.getSelectedRow();
if (selectedRowIndex == -1) {
    JOptionPane.showMessageDialog(this, "Please choose a row to process this function",
            "", JOptionPane.INFORMATION_MESSAGE);
} else {
    Book_ViewDetail.setVisible(true);
    disableAllField();
    setValues(listOfBook.get(selectedRowIndex));
    int choice = JOptionPane.showConfirmDialog(this, "Do you want to delete "
            + selectedRowIndex + " ?");
    if (choice == JOptionPane.YES_OPTION) {
        String bookID = txtBookId.getText();
        if (DBconnection.deleteBook(bookID) != 0) {
            myTableModel.removeRow(selectedRowIndex);
            JOptionPane.showMessageDialog(this, "Delete successfull");
            //Cập nhật lại bảng:
        } else {
            JOptionPane.showMessageDialog(this, "Delete fail");
        }
    }
}
```

Using if/else for delete function will allow user make their choice. In this case, after the user selected a row then click on Delete the confirmation box will pop-up and ask “Do you want to delete”. If user select “Yes” the selected row will be remove from the table also the database and the “Delete successful” box will pop-up, but in case there are something went wrong the “Delete fail” box will pop-up. However, if user select another option instead of Yes there are nothing happen because there are no specific command for it.

c. JOptionPane

JoptionPane allow the user know what happen after they did something with the program while running it. Delete function are using JOptionPane:



Number of Book: 8

Book ID	Title	Author	Publisher	Stock
2	Những đứa trẻ ...	Dương Minh Tu...	2018	1
3	Something wro...	trung	2018	10
10	Life	trung	2018	10
11	Long life	trung	2018	10
12	tr	we	123	12
20	431	421	421421	421412
21	432	432	432	432
22	Wrong life	Trung	2018	

Select an Option

Do you want to delete 0 ?

2

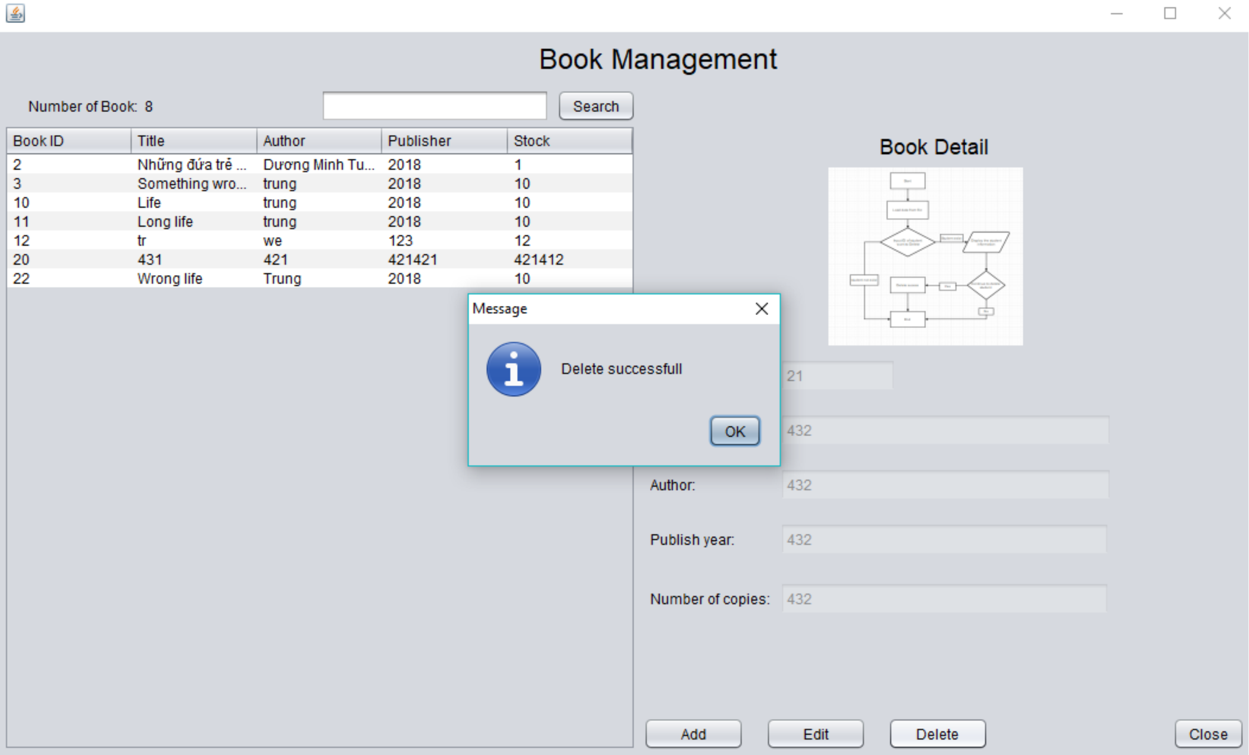
Những đứa trẻ không bao giờ lớn

Author: Dương Minh Tuấn

Publish year: 2018

Number of copies: 1

When user select a row then click on Delete button the JOptionPaneconfirm will appear, after select Yes



Number of Book: 8

Book ID	Title	Author	Publisher	Stock
2	Những đứa trẻ ...	Dương Minh Tu...	2018	1
3	Something wro...	trung	2018	10
10	Life	trung	2018	10
11	Long life	trung	2018	10
12	tr	we	123	12
20	431	421	421421	421412
22	Wrong life	Trung	2018	10

Message

Delete successfull

21

432

Author: 432

Publish year: 432

Number of copies: 432

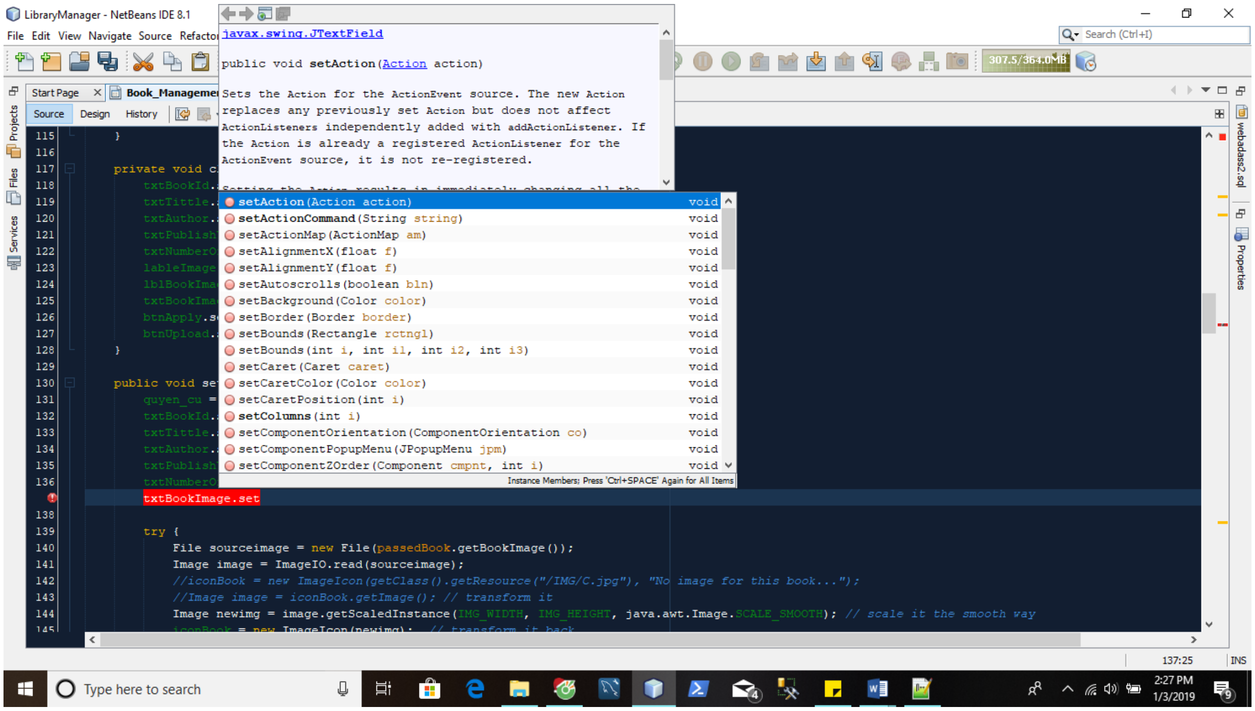
Another JOptionPane will appear and let the user know that they have delete successful

IV. NetBeans

Using NetBeans to code the program make everything more easily, it help auto-finished code, suggestion code, auto-provided getter setter create constructor …

For auto-finished code, coder can just type "Sout" then press Tab, the code will automatically finish and it should look like: System.out.println("");

These picture below will show how NetBeans helping to complete the program:

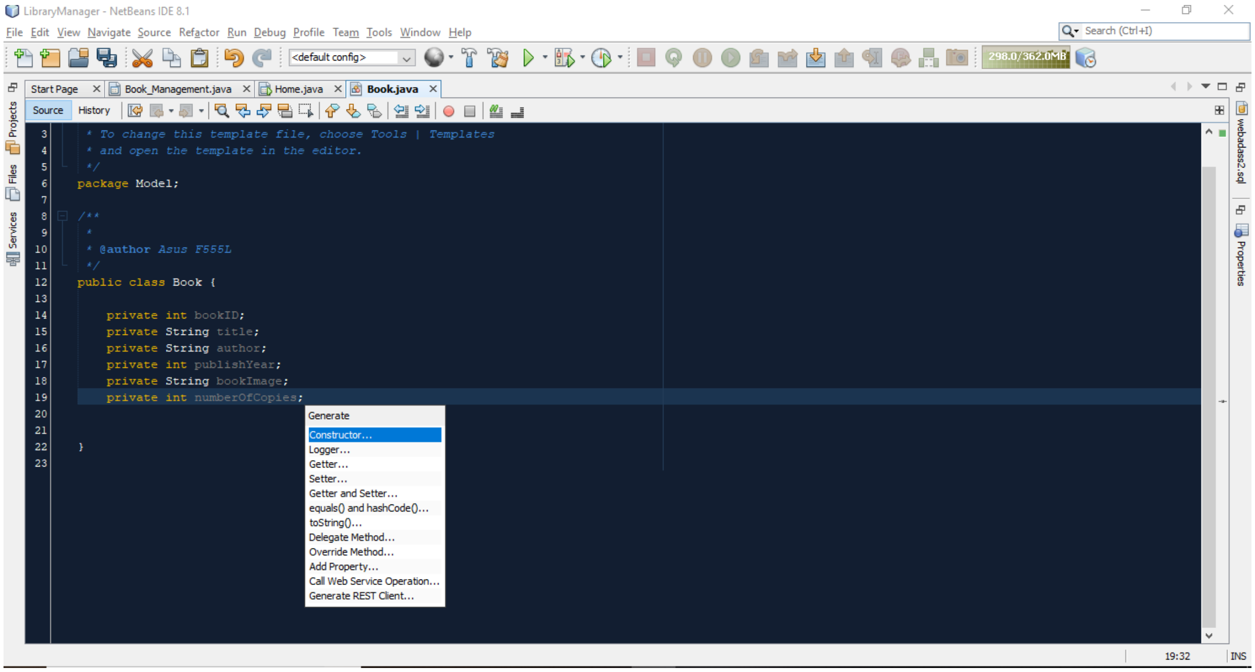


```
javax.swing.JTextField

public void setAction(Action action)

Sets the Action for the ActionEvent source. The new Action
replaces any previously set Action but does not affect
ActionListeners independently added with addActionListener. If
the Action is already a registered ActionListener for the
ActionEvent source, it is not re-registered.
```

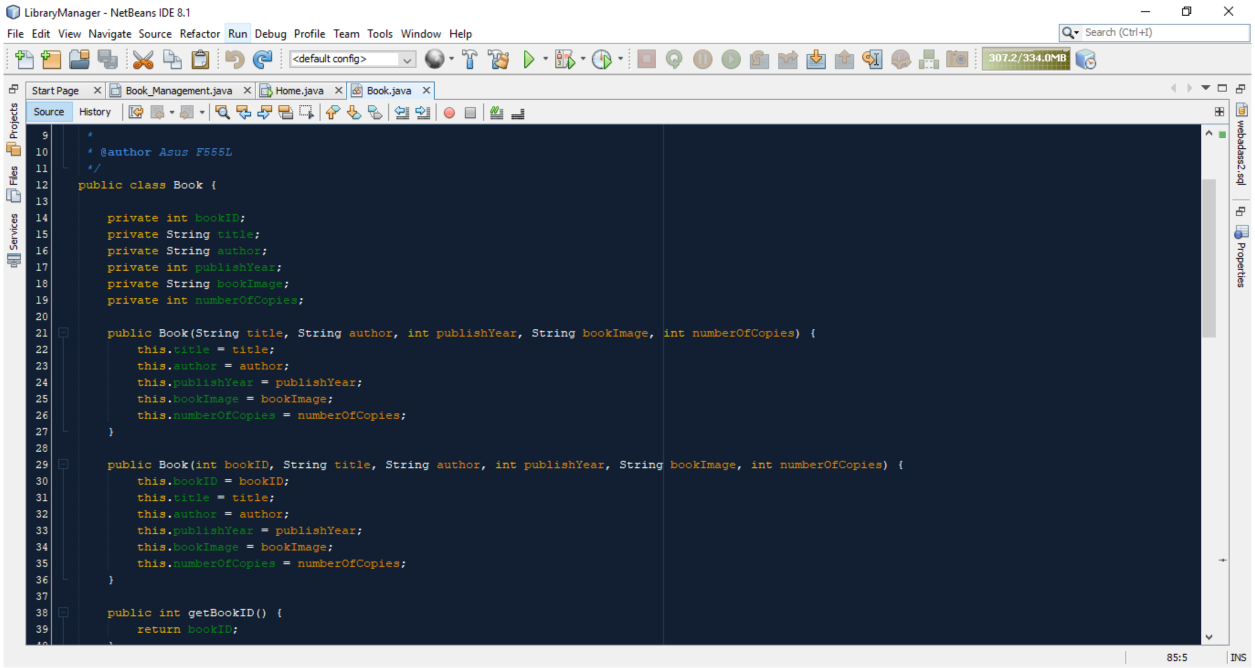
```
setAction(Action action)                                   void
setActionCommand(String string)                            void
setActionMap(ActionMap am)                                 void
setAlignmentX(float f)                                     void
setAlignmentY(float f)                                     void
setAutoscrolls(boolean bln)                                void
setBackground(Color color)                                 void
setBorder(Border border)                                   void
setBounds(Rectangle rctngl)                                void
setBounds(int i, int i1, int i2, int i3)                   void
setCaret(Caret caret)                                      void
setCaretColor(Color color)                                 void
setCaretPosition(int i)                                    void
setColumns(int i)                                          void
setComponentOrientation(ComponentOrientation co)           void
setComponentPopupMenu(JPopupMenu jpm)                      void
setComponentZOrder(Component cmpnt, int i)                 void
                Instance Members; Press 'Ctrl+SPACE' Again for All Items
```

```
    }

    private void c
        txtBookId.
        txtTittle.
        txtAuthor.
        txtPublish
        txtNumberO
        lableImage
        lblBookIma
        txtBookIma
        btnApply.s
        btnUpload.
    }

    public void se
        quyen_cu =
        txtBookId.
        txtTittle.
        txtAuthor.
        txtPublish
        txtNumberO
        txtBookImage.set

        try {
            File sourceimage = new File(passedBook.getBookImage());
            Image image = ImageIO.read(sourceimage);
            //iconBook = new ImageIcon(getClass().getResource("/IMG/C.jpg"), "No image for this book...");
            //Image image = iconBook.getImage(); // transform it
            Image newimg = image.getScaledInstance(IMG_WIDTH, IMG_HEIGHT, java.awt.Image.SCALE_SMOOTH); // scale it the smooth way
            iconBook = new ImageIcon(newimg);  // transform it back
```

NetBeans not just suggest the next code it also show the coder what is it, how to use it, and where the error that program care facing.



```java
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package Model;

/**
 *
 * @author Asus F555L
 */
public class Book {

    private int bookID;
    private String title;
    private String author;
    private int publishYear;
    private String bookImage;
    private int numberOfCopies;


}
```



```java
 *
 * @author Asus F555L
 */
public class Book {

    private int bookID;
    private String title;
    private String author;
    private int publishYear;
    private String bookImage;
    private int numberOfCopies;

    public Book(String title, String author, int publishYear, String bookImage, int numberOfCopies) {
        this.title = title;
        this.author = author;
        this.publishYear = publishYear;
        this.bookImage = bookImage;
        this.numberOfCopies = numberOfCopies;
    }

    public Book(int bookID, String title, String author, int publishYear, String bookImage, int numberOfCopies) {
        this.bookID = bookID;
        this.title = title;
        this.author = author;
        this.publishYear = publishYear;
        this.bookImage = bookImage;
        this.numberOfCopies = numberOfCopies;
    }

    public int getBookID() {
        return bookID;
```

By hold alt + insert (alt + fn + insert) the list will appear and auto-provided getter setter, create constructor for coder.

Task 3

I. Testing and analysis

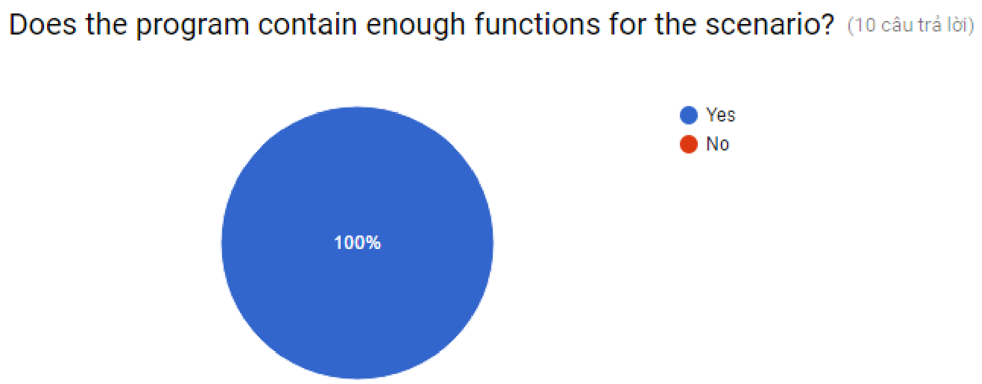
Number	Test case	Input	Expected Output	Actual Output	Status
1	Add Book	BookID: 1 Title: Life Publisher: 2018 Author: Trung Stock:10	New book added to the database, the program show: “Add successfully”.	The program show: “Add fail”. But New book are added to database	Fail
2	Add Book	BookID: 2 Title: Life Publisher: 2018 Author: Trung Stock:10	New book added to the database, the program show: “Add successfully”.	New book added to the database, the program show: “Add successfully”.	Passed
3	Add Book	BookID: 3 Title: Life Publisher: 2018 Author: Trung Stock:10	New book added to the database, the program show: “Add successfully” and table show new book	New book added to the database, the program show: “Add successfully” and table show new book	Passed
4	Edit book	BookID: 1 Title: Life Publisher: 2018 Author: Trung Stock:10 Change Title to Wrong life	Book are updated in the database, the program show: “Update successfully” and table show updated book	Book are updated in the database, the program show: “Update successfully”. But table update at wrong row	Fail
5	Edit book	BookID: 1 Title: Wrong life Publisher: 2018 Author: Trung Stock:10 Change Title to life	Book are updated in the database, the program show: “Update successfully” and table show updated book	Book are updated in the database, the program show: “Update successfully” and table show updated book	Passed
6	Delete book	Select row 0 which is BookID: 1 Title: Wrong life	Current book removed from database and the program show: “Delete successfully”	Current book removed from database and the program show: “Delete successfully”	Passed

		Publisher: 2018 Author: Trung Stock:10 Change Title to life	and current book are removed from table	and current book are removed from table	
7	Search book	Life	Table list show all the book that there name contain “life” key word in there	Table list show all the book that there name contain “life” key word in there	Passed
8	Search book	Life	Table list show all the book that there name contain “life” key word in there. The table change to normal after remove the keyword	Table list show all the book that there name contain “life” key word in there. But keep the current searched list	Fail
9	Search book	Life	Table list show all the book that there name contain “life” key word in there. The table change to normal after remove the keyword and selected search button again	Table list show all the book that there name contain “life” key word in there. The table change to normal after remove the keyword and selected search button again	Passed

II. Feedback and feedback analysis

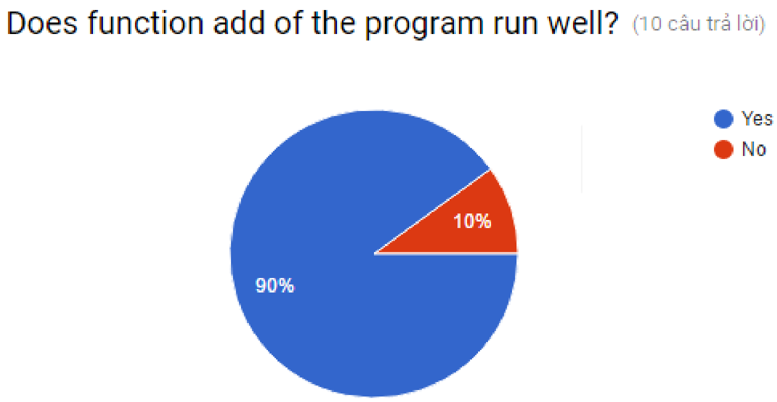
The survey prepared to show the process while running the program and also contains 9 questions about system requirements, usability and performance for peers to take feedbacks.

1. Does the program contain enough functions for the scenario?



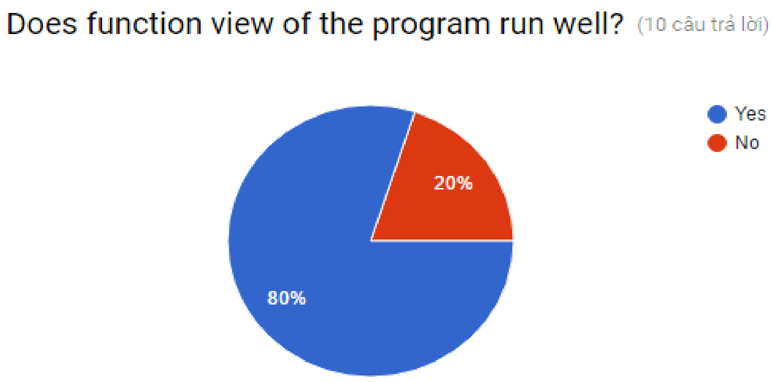
There are 10 answers for this question and all of the answers agreed that the program contains enough functions for the scenario: add, view, delete and update. In the future, the program will update more functions to manage Borrow ticket

2. Does the function add of the program run well?



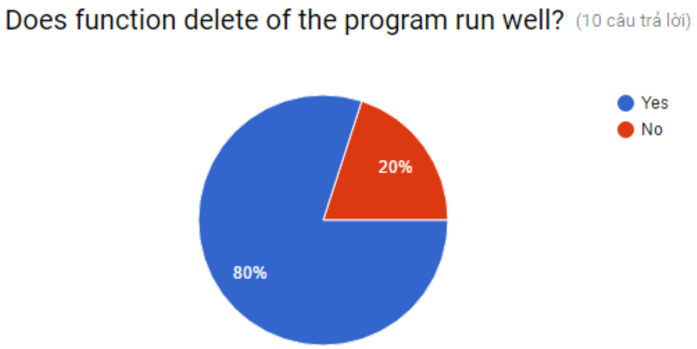
There are 10 answers for this question and about 90% comments of peers agreed that the function add of the program run well and the others 10% disagreed with it.

3. Does the function view of the program run well?



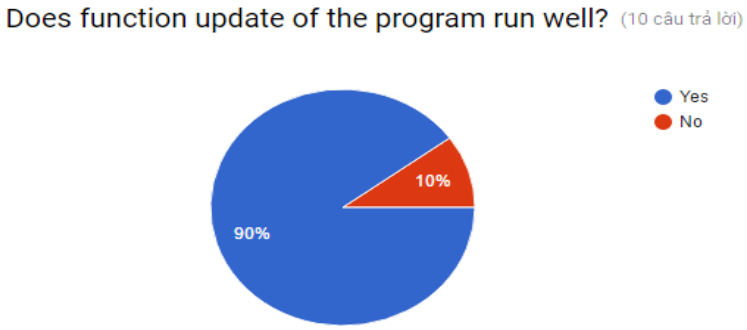
There are 10 answers for this question and about 80% comments of peers agreed that the function view of the program run well and the others 20% disagreed with it. A comment mentioned to this question that in the table of view function, user can view detail information of selected books.

4. Does the function delete of the program run well?



There are 10 answers for this question and about 80% comments of peers agreed that the function view of the program run well and the others 20% disagreed with it. A comment of peer about this function, they suggest that the program need to show confirm dialog to ask user confirmation when user choose to apply new student. Like peer's feedback, the program will be updated in the future.

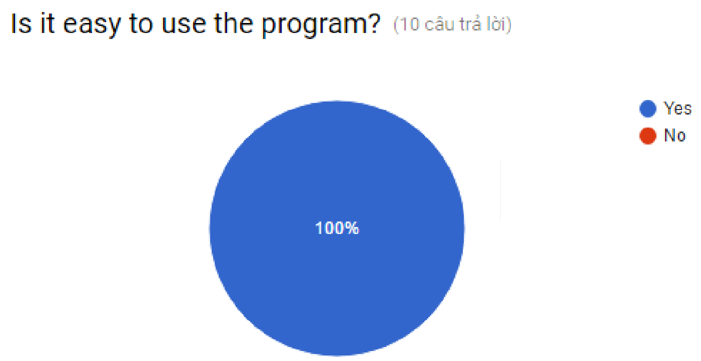
5. Does the function update of the program run well?



There are 10 answers for this question and 10% comments of peers see the problem with update function and take feedbacks, the others 90% agree that function update run well and it doesn't need to update. Just like function delete, a comment suggest that when user want to choose function update, a confirm dialog will be showed to user. It is good and I will fix the code in the function view, if user select a row and click to button update, the program will show a confirm dialog to ask user confirmation.

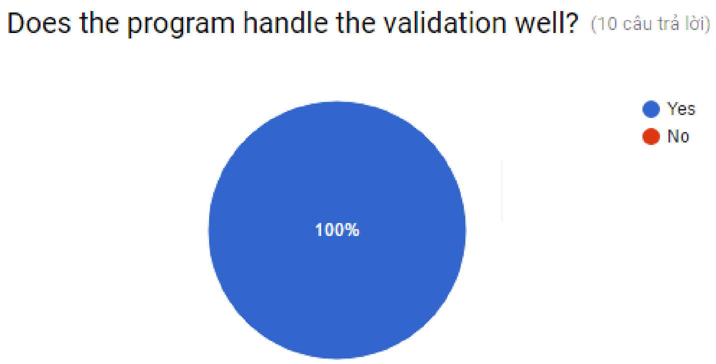
6. Is it easy to use the program?

III.


Is it easy to use the program? (10 câu trả lời)

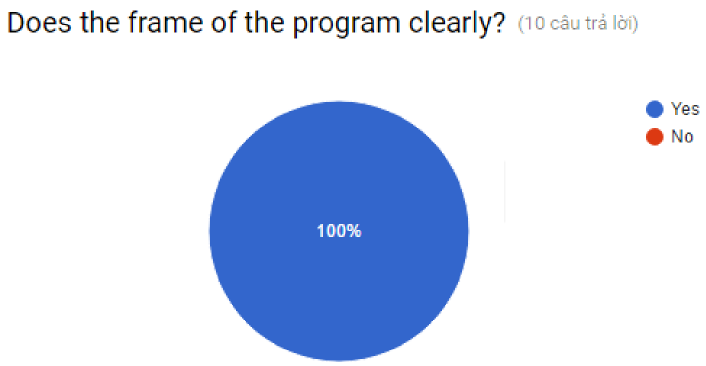
There are 10 answers for this question and all of the answers agreed that the program is easy to use.

7. Does the program handle the validation well?


Does the program handle the validation well? (10 câu trả lời)

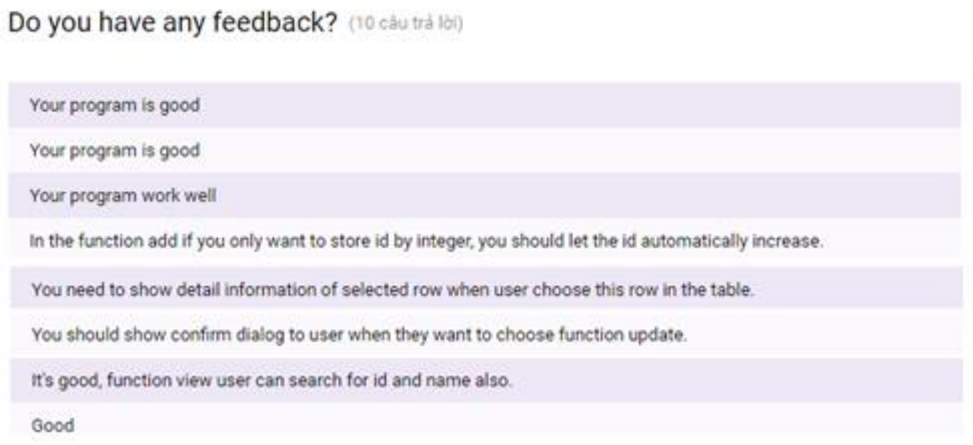
There are 10 answers for this question and there is no comment about the validation of the program. In the program, functions add do not allow user to input character or string into ID and integer into the others text fields. With function view user can search for ID and name also, a good comment mentioned about it. Function update user cannot update ID.

8. Does the frame of the program clearly?


Does the frame of the program clearly? (10 câu trả lời)

There are 10 answers for this question and there is no comment about the frame of the program. Java Swing used in Book Managemer so the frames are clearly, good-looking.

9. Do you have any feedback?



Do you have any feedback? (10 câu trả lời)

Your program is good

Your program is good

Your program work well

In the function add if you only want to store id by integer, you should let the id automatically increase.

You need to show detail information of selected row when user choose this row in the table.

You should show confirm dialog to user when they want to choose function update.

It's good, function view user can search for id and name also.

Good

I’m very appreciated all of peer’s feedbacks, it helps me find out some problems with my program and makes it becoming better in the future.

IV. User documentation

The program design as simple as possible. First user run the program then select Book management (Only function that can be use now). In Book management, user can see detail of the book buy select the row then information will appear at the right side of the program. Moreover, to add more book user can select the Add button on the bottom right of the program screen, then input new book information then select the Apply button which will appear after select Add button.

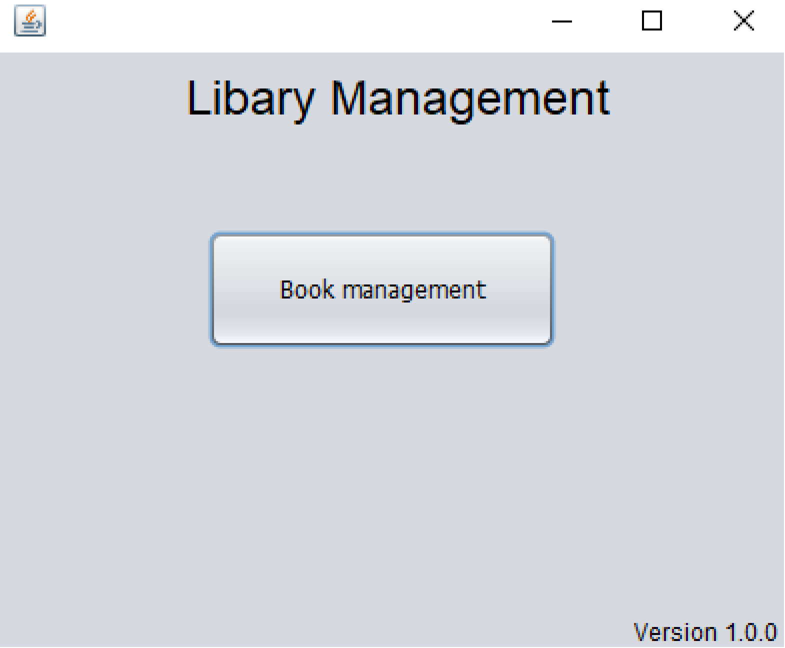
Also user can Edit the book information by select a current book then select Edit button, after edited select Update button to complete the edition.

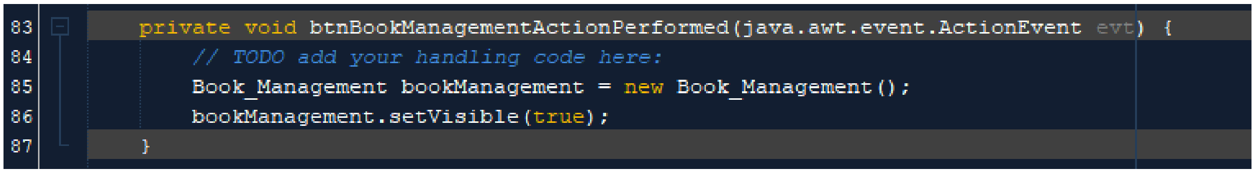
To delete the book, user can select the current book then click on delete button. The confirmation box will appear to confirm delete book user can select Yes, if user change their mind, select other option to cancel the delete.

If there are many book and user just want to find one book, they can use Search function that will be on the top of the book list table. To use search function, input the key word then click on search, the book that contain that key word will be showed on the book list table. To make Book list return to default, remove all the key word on the search box then click search button again.

V. Technical documentation

- Main

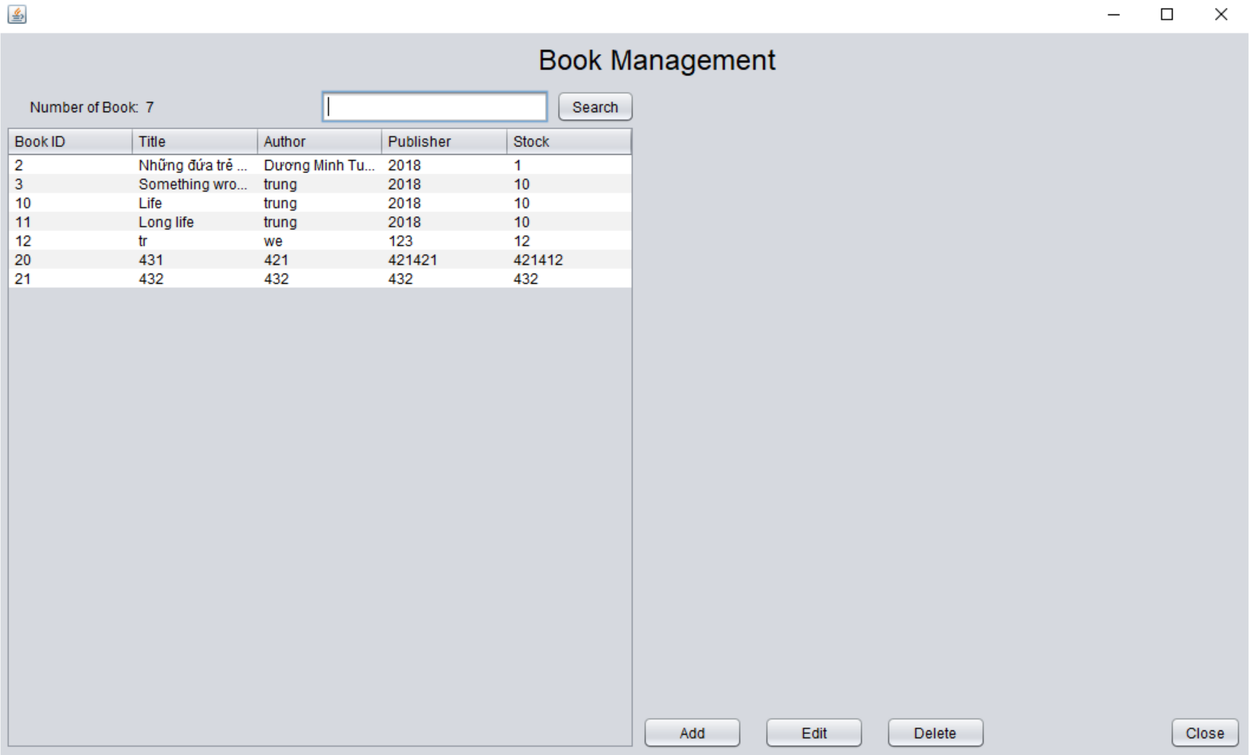




```
    private void btnBookManagementActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        Book_Management bookManagement = new Book_Management();
        bookManagement.setVisible(true);
    }
```

The main frame will allow user to select function of the program then redirect user to the function frame. At this moment, there are just only one but in future the will be few move such as borrow ticket and manage reader

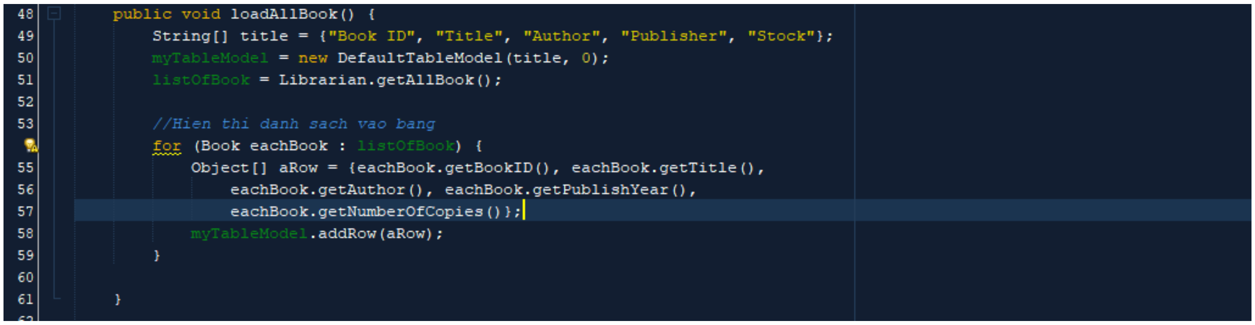
- Book management
- Main book management frame



Book Management

Number of Book: 7

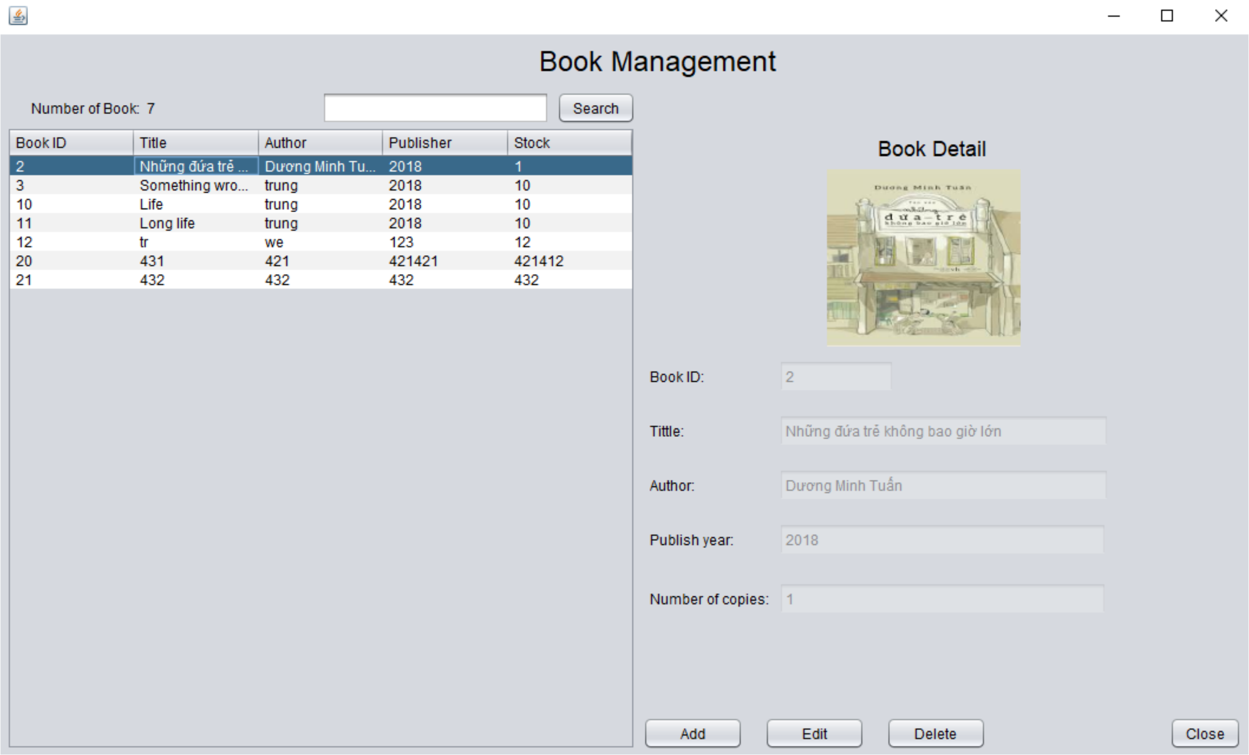
Book ID	Title	Author	Publisher	Stock
2	Những đứa trẻ ...	Dương Minh Tu...	2018	1
3	Something wro...	trung	2018	10
10	Life	trung	2018	10
11	Long life	trung	2018	10
12	tr	we	123	12
20	431	421	421421	421412
21	432	432	432	432



```
    public void loadAllBook() {
        String[] title = {"Book ID", "Title", "Author", "Publisher", "Stock"};
        myTableModel = new DefaultTableModel(title, 0);
        listOfBook = Librarian.getAllBook();

        //Hien thi danh sach vao bang
        for (Book eachBook : listOfBook) {
            Object[] aRow = {eachBook.getBookID(), eachBook.getTitle(),
                eachBook.getAuthor(), eachBook.getPublishYear(),
                eachBook.getNumberOfCopies()};
            myTableModel.addRow(aRow);
        }

    }
```

The main manager book allow user to see the list of the books and also allow them to Add new ones, Edit the current book, remove book from the list and also search for them

- View



Book Management

Number of Book: 7

Book ID	Title	Author	Publisher	Stock
2	Những đứa trẻ ...	Dương Minh Tu...	2018	1
3	Something wro...	trung	2018	10
10	Life	trung	2018	10
11	Long life	trung	2018	10
12	tr	we	123	12
20	431	421	421421	421412
21	432	432	432	432

Book Detail

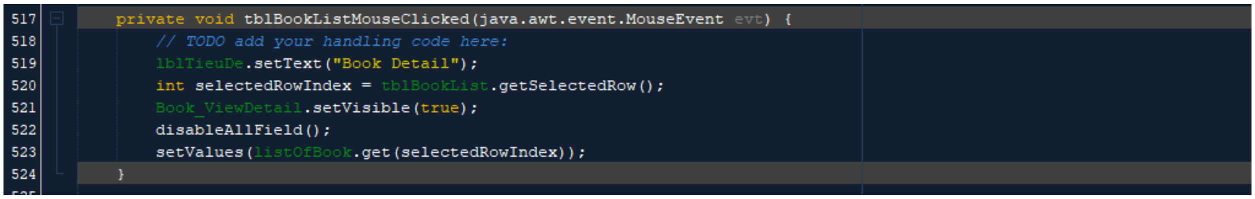
Book ID: 2

Tittle: Những đứa trẻ không bao giờ lớn

Author: Dương Minh Tuấn

Publish year: 2018

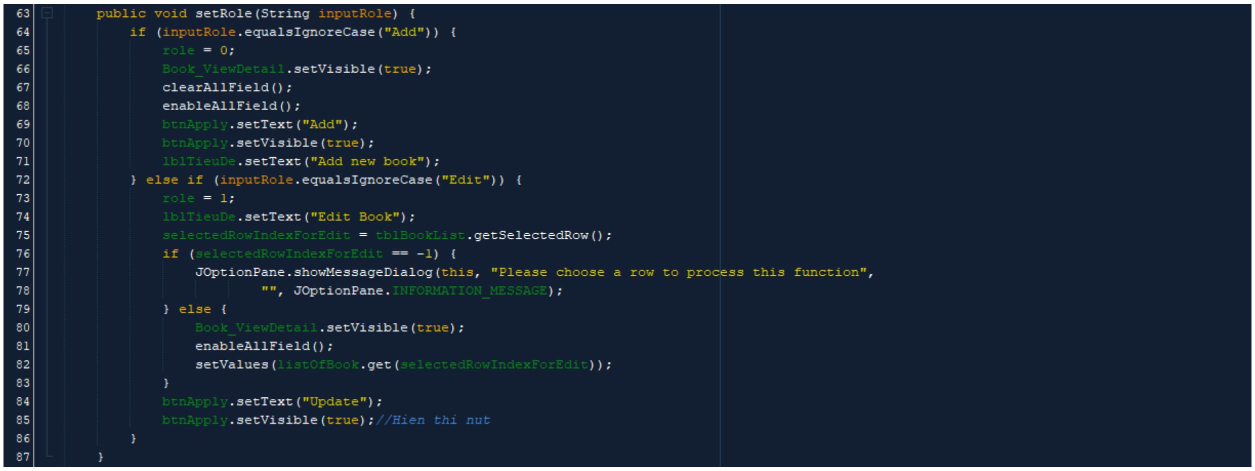
Number of copies: 1



```
    private void tblBookListMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        lblTieuDe.setText("Book Detail");
        int selectedRowIndex = tblBookList.getSelectedRow();
        Book_ViewDetail.setVisible(true);
        disableAllField();
        setValues(listOfBook.get(selectedRowIndex));
    }
```

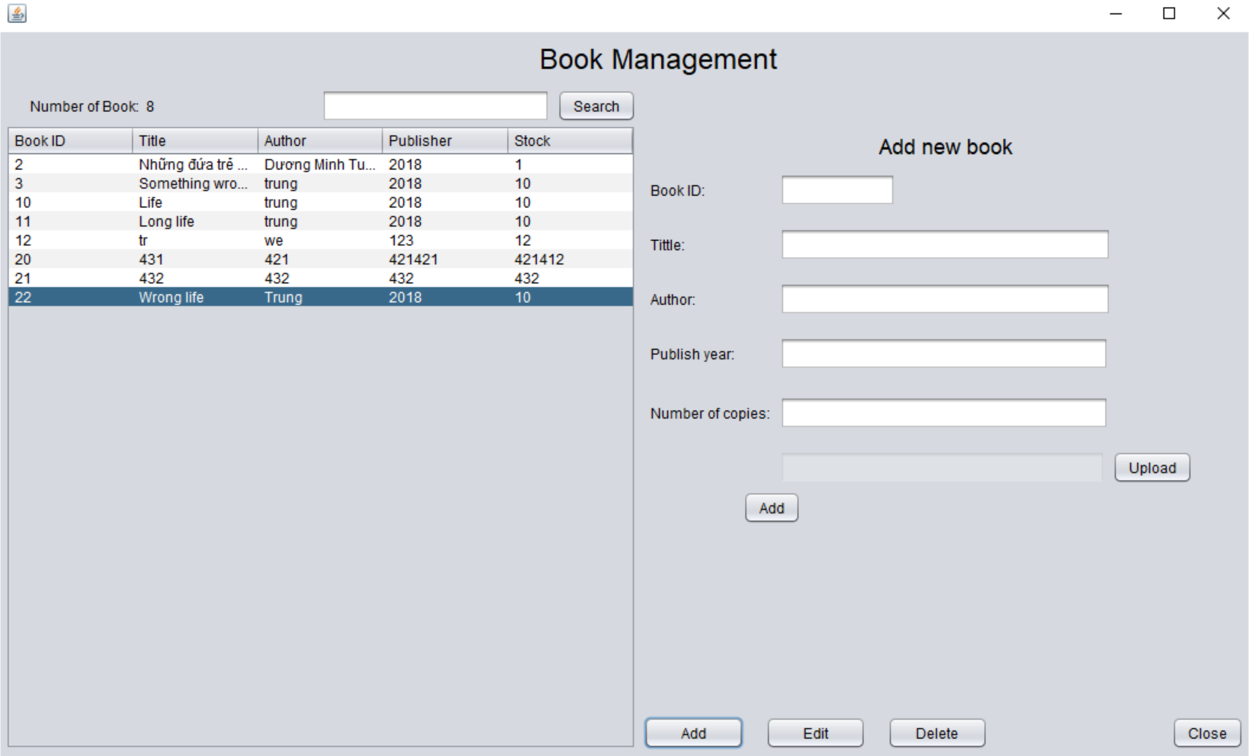
User can view book information by select the book on list.

Before look at functions Add, Edit. Both function are set-up with there own name.



```
    public void setRole(String inputRole) {
        if (inputRole.equalsIgnoreCase("Add")) {
            role = 0;
            Book_ViewDetail.setVisible(true);
            clearAllField();
            enableAllField();
            btnApply.setText("Add");
            btnApply.setVisible(true);
            lblTieuDe.setText("Add new book");
        } else if (inputRole.equalsIgnoreCase("Edit")) {
            role = 1;
            lblTieuDe.setText("Edit Book");
            selectedRowIndexForEdit = tblBookList.getSelectedRow();
            if (selectedRowIndexForEdit == -1) {
                JOptionPane.showMessageDialog(this, "Please choose a row to process this function",
                        "", JOptionPane.INFORMATION_MESSAGE);
            } else {
                Book_ViewDetail.setVisible(true);
                enableAllField();
                setValues(listOfBook.get(selectedRowIndexForEdit));
            }
            btnApply.setText("Update");
            btnApply.setVisible(true);//Hien thi nut
        }
    }
```

- Add new book



Book Management

Number of Book: 8

Book ID	Title	Author	Publisher	Stock
2	Những đứa trẻ ...	Dương Minh Tu...	2018	1
3	Something wro...	trung	2018	10
10	Life	trung	2018	10
11	Long life	trung	2018	10
12	tr	we	123	12
20	431	421	421421	421412
21	432	432	432	432
22	Wrong life	Trung	2018	10

Add new book

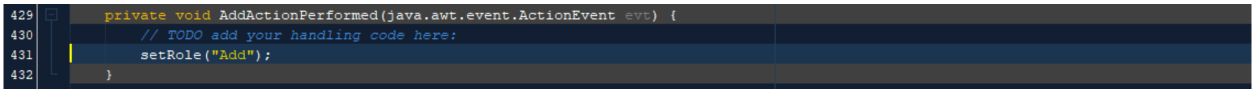
Book ID:

Tittle:

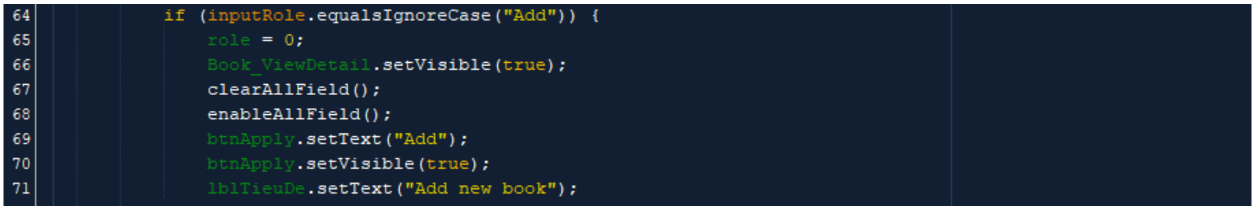
Author:

Publish year:

Number of copies:



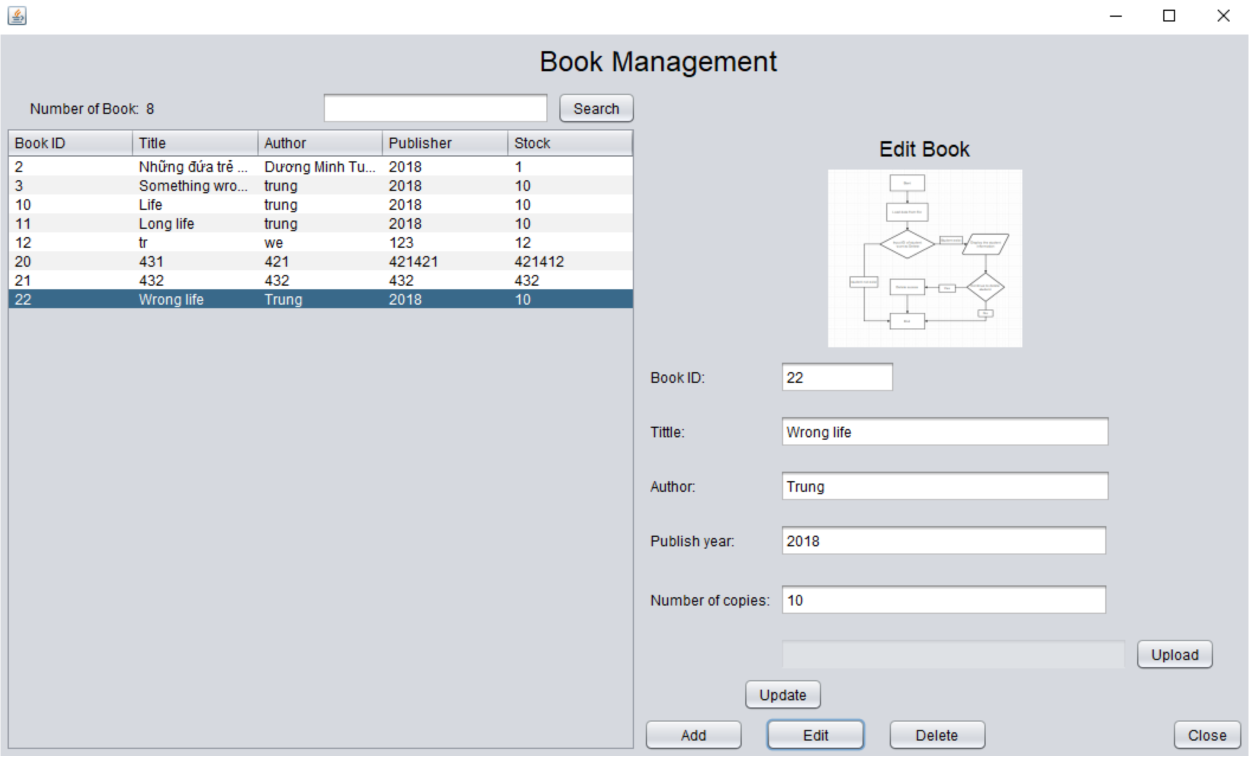
```
    private void AddActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        setRole("Add");
    }
```



```
            if (inputRole.equalsIgnoreCase("Add")) {
                role = 0;
                Book_ViewDetail.setVisible(true);
                clearAllField();
                enableAllField();
                btnApply.setText("Add");
                btnApply.setVisible(true);
                lblTieuDe.setText("Add new book");
```

The function Add will allow user to add new book to list. After click on “Add” button all the field will be clear and available for user to input the new one, also the apply button will appear. Therefore, after input text, user can select apply to input the new one to list.

- Edit book



Number of Book: 8

Book ID	Title	Author	Publisher	Stock
2	Những đứa trẻ ...	Dương Minh Tu...	2018	1
3	Something wro...	trung	2018	10
10	Life	trung	2018	10
11	Long life	trung	2018	10
12	tr	we	123	12
20	431	421	421421	421412
21	432	432	432	432
22	Wrong life	Trung	2018	10

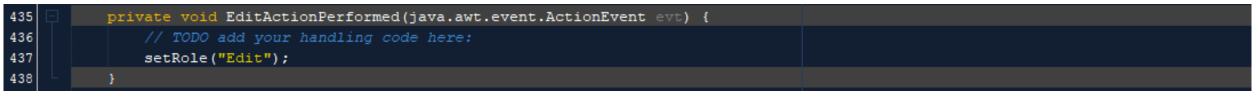
Book ID: 22

Tittle: Wrong life

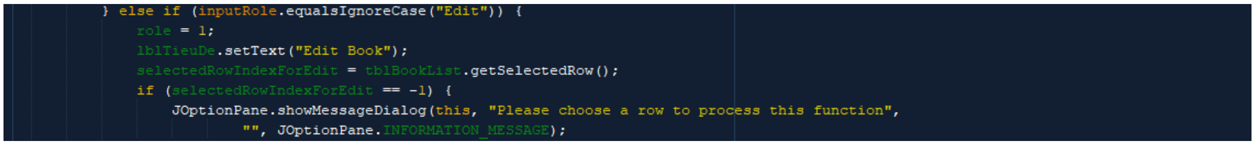
Author: Trung

Publish year: 2018

Number of copies: 10



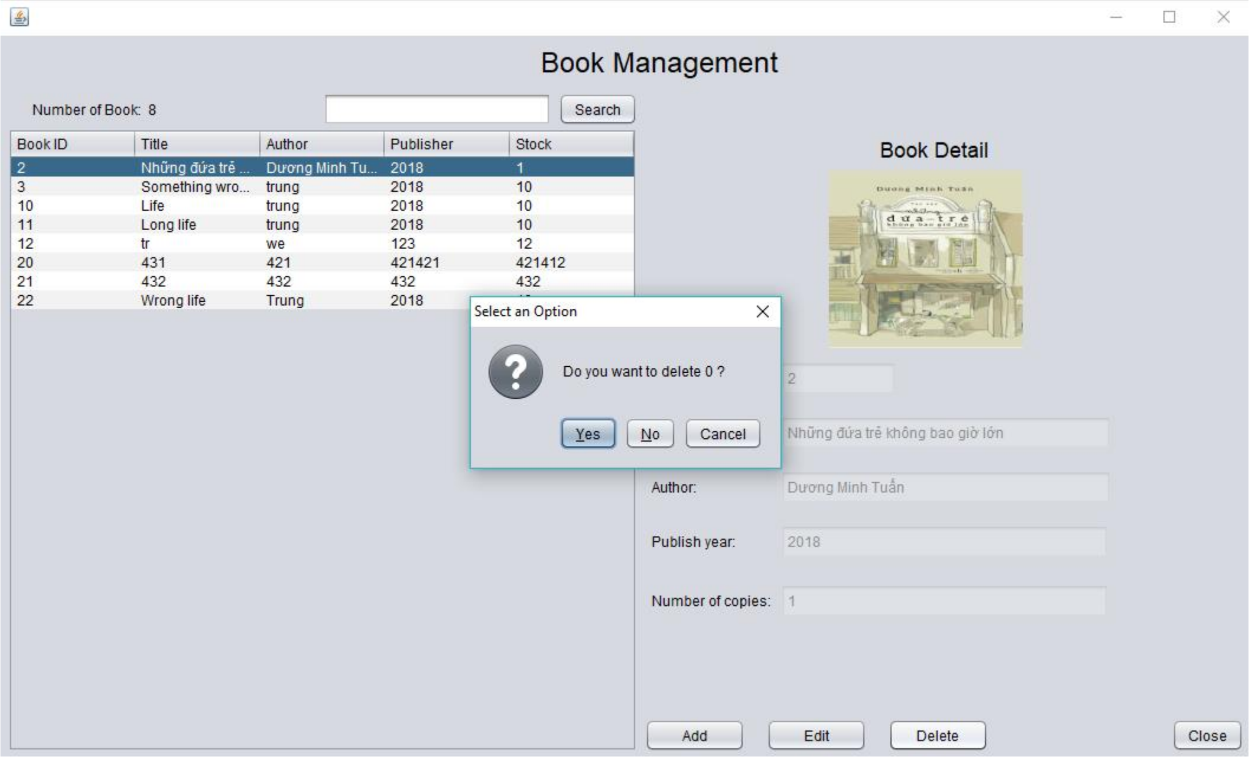
```java
    private void EditActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        setRole("Edit");
    }
```



```java
        } else if (inputRole.equalsIgnoreCase("Edit")) {
            role = 1;
            lblTieuDe.setText("Edit Book");
            selectedRowIndexForEdit = tblBookList.getSelectedRow();
            if (selectedRowIndexForEdit == -1) {
                JOptionPane.showMessageDialog(this, "Please choose a row to process this function",
                        "", JOptionPane.INFORMATION_MESSAGE);
```

Edit function will get all the information from the selected row then all the field are also available for user to make change. The Update button will appear right at the Apply button of add function location which after make change, user can select Update button for confirm the update.

- Delete



Book Management

Number of Book: 8

Book ID	Title	Author	Publisher	Stock
2	Những đứa trẻ ...	Dương Minh Tu...	2018	1
3	Something wro...	trung	2018	10
10	Life	trung	2018	10
11	Long life	trung	2018	10
12	tr	we	123	12
20	431	421	421421	421412
21	432	432	432	432
22	Wrong life	Trung	2018	[illegible]

Select an Option

Do you want to delete 0 ?

Book Detail

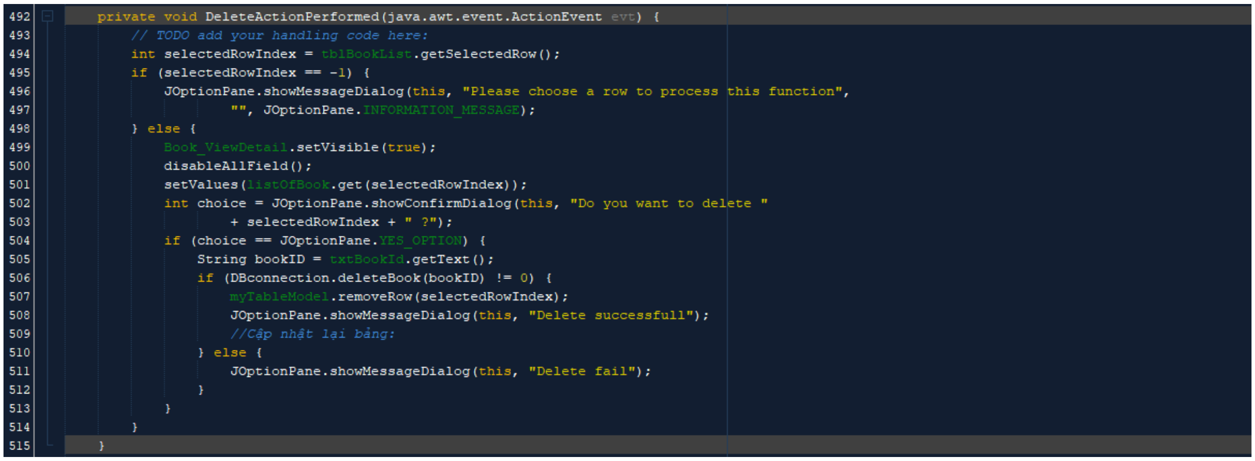
2

Những đứa trẻ không bao giờ lớn

Author: Dương Minh Tuấn

Publish year: 2018

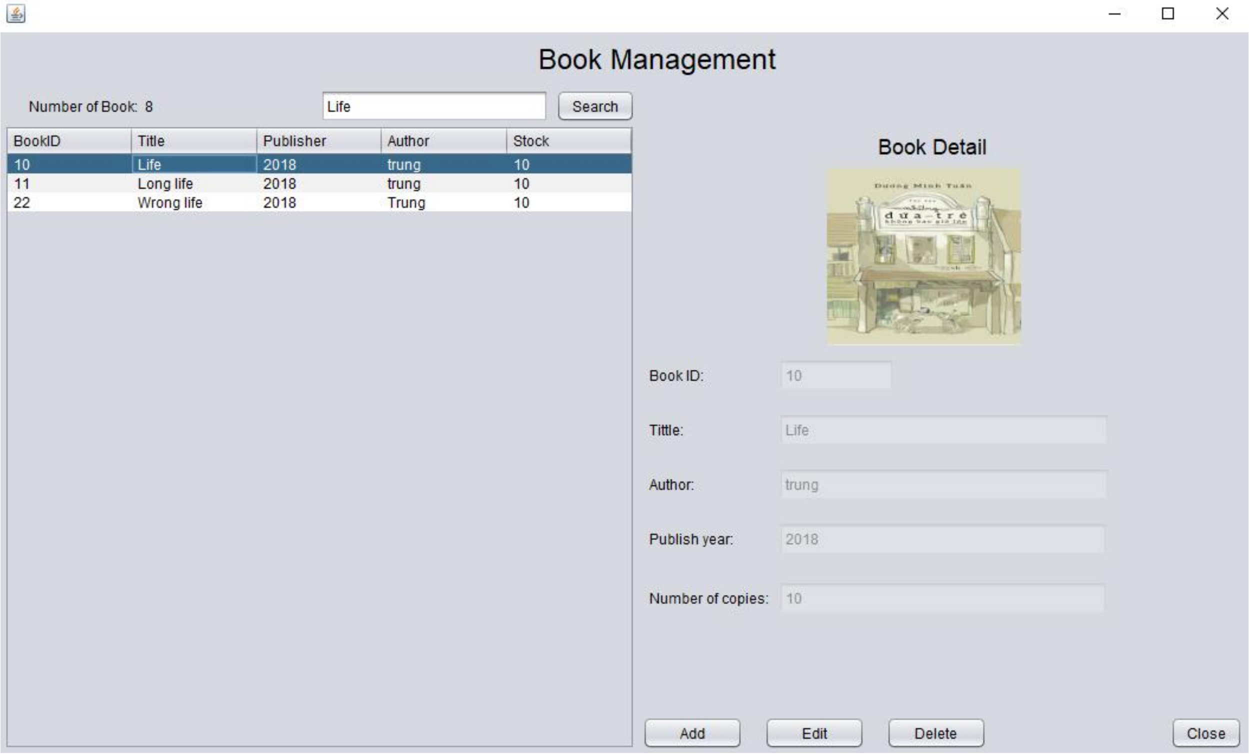
Number of copies: 1



```
    private void DeleteActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        int selectedRowIndex = tblBookList.getSelectedRow();
        if (selectedRowIndex == -1) {
            JOptionPane.showMessageDialog(this, "Please choose a row to process this function",
                    "", JOptionPane.INFORMATION_MESSAGE);
        } else {
            Book_ViewDetail.setVisible(true);
            disableAllField();
            setValues(listOfBook.get(selectedRowIndex));
            int choice = JOptionPane.showConfirmDialog(this, "Do you want to delete "
                    + selectedRowIndex + " ?");
            if (choice == JOptionPane.YES_OPTION) {
                String bookID = txtBookId.getText();
                if (DBconnection.deleteBook(bookID) != 0) {
                    myTableModel.removeRow(selectedRowIndex);
                    JOptionPane.showMessageDialog(this, "Delete successfull");
                    //Cập nhật lại bảng:
                } else {
                    JOptionPane.showMessageDialog(this, "Delete fail");
                }
            }
        }
    }
```

Selected this button, there will be a confirmation pop-up that confirm to delete the current selected.

- Search book by name



Number of Book: 8

Life

BookID	Title	Publisher	Author	Stock
10	Life	2018	trung	10
11	Long life	2018	trung	10
22	Wrong life	2018	Trung	10

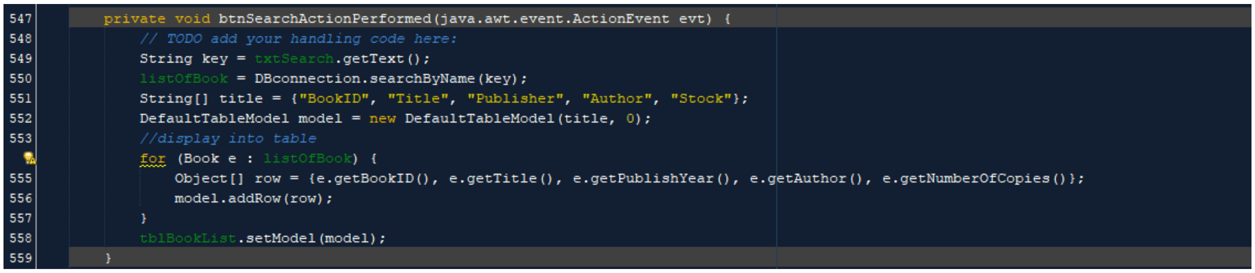
Book ID: 10

Tittle: Life

Author: trung

Publish year: 2018

Number of copies: 10



```
private void btnSearchActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String key = txtSearch.getText();
    listOfBook = DBconnection.searchByName(key);
    String[] title = {"BookID", "Title", "Publisher", "Author", "Stock"};
    DefaultTableModel model = new DefaultTableModel(title, 0);
    //display into table
    for (Book e : listOfBook) {
        Object[] row = {e.getBookID(), e.getTitle(), e.getPublishYear(), e.getAuthor(), e.getNumberOfCopies()};
        model.addRow(row);
    }
    tblBookList.setModel(model);
}
```

By input the book name, the search function will look through all book of the list then will show on the table all the books that contain the key word.

Code

- Set role for Add and Edit

public void setRole(String inputRole) {

if (inputRole.equalsIgnoreCase("Add")) {

role = 0;

```
            Book_ViewDetail.setVisible(true);
            clearAllField();
            enableAllField();
            btnApply.setText("Add");
            btnApply.setVisible(true);
            lblTieuDe.setText("Add new book");
        } else if (inputRole.equalsIgnoreCase("Edit")) {
            role = 1;
            lblTieuDe.setText("Edit Book");
            selectedRowIndexForEdit = tblBookList.getSelectedRow();
            if (selectedRowIndexForEdit == -1) {
                JOptionPane.showMessageDialog(this, "Please choose a row to process this function",
                        "", JOptionPane.INFORMATION_MESSAGE);
            } else {
                Book_ViewDetail.setVisible(true);
                enableAllField();
                setValues(listOfBook.get(selectedRowIndexForEdit));
            }
            btnApply.setText("Update");
            btnApply.setVisible(true);//Hien thi nut
        }
    }
```

By set Role for Add and Edit, we have minimize the code more simple so other can read, understand, and edit more easier.

- Apply and update button

```
private void btnApplyActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        if (role == 1) { //Edit
            String title = txtTittle.getText();
            String author = txtAuthor.getText();
            String publishYear = txtPublishYear.getText();
```

```
      String nbOfCopies = txtNumberOfCopies.getText();
      String bookImage = txtBookImage.getText();
      Book quyen_moi = new Book(quyen_cu.getBookID(), title, author,
Integer.parseInt(publishYear), bookImage, Integer.parseInt(nbOfCopies));
      if (Librarian.updateBook(quyen_cu, quyen_moi)) {
        updateJTable_Edit(quyen_moi);
        JOptionPane.showMessageDialog(this, "Update successfull");
        Book_ViewDetail.setVisible(false);
        disableAllField();
      } else {
        JOptionPane.showMessageDialog(this, "Edit fail");
      }
    } else if (role == 0) { //Add
      String tittle = txtTittle.getText();
      String author = txtAuthor.getText();
      String publishYear = txtPublishYear.getText();
      String nbOfCopies = txtNumberOfCopies.getText();
      String bookImage = txtBookImage.getText();
      Book inputBook = new Book(tittle, author,
          Integer.parseInt(publishYear),
          bookImage,
          Integer.parseInt(nbOfCopies));
      int ID_MOI_SINH = Librarian.addBook(inputBook);

      if (ID_MOI_SINH != 0) {
        JOptionPane.showMessageDialog(this, "Adding successfully!");
        //Cap nhat gia tri moi vao JTable
        inputBook.setBookID(ID_MOI_SINH);
        try {
          updateJTable(inputBook);
        } catch (Exception ex) {
          System.out.println(ex);
        }
        Book_ViewDetail.setVisible(false);
        disableAllField();
      } else {
        JOptionPane.showMessageDialog(this, "An error appears. "
            + "Adding unsuccessfully!");
      }
    }
  }
```

The apply and update button are simple one button which change based on the function that user selected. If user select Add the button Apply will appear, but if select Edit the Update button will appear. By set role for each of the function Role 0 is add and Edit is 1, the button will base on that role to display Apply or Update.

Book manager Version 1.0.0