Java Programming

Bookshop Application

Higher Diploma in Science in computing

John Purdy

B00056690

**Introduction**

The aim of this report is to provide background information about the Java book application I have created which you can find attached. The programme was designed in order to allow the user to choose from the following options:

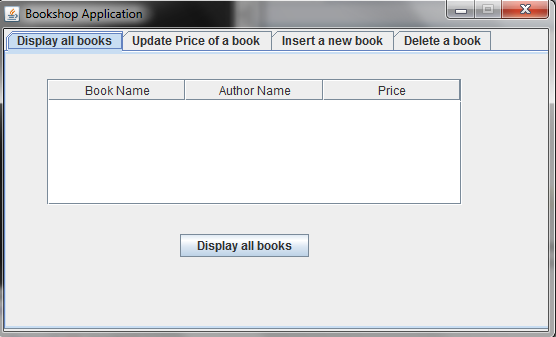
* Option 1: Display all books in the store or database
* Option2: Update the price of a book by selecting its name
* Option3: Insert a new book into the store or database
* Option4: Delete a book from the database or store by selecting its name

This report will now go through this entire process in detail explaining how the application works and describe the classes and methods involved.

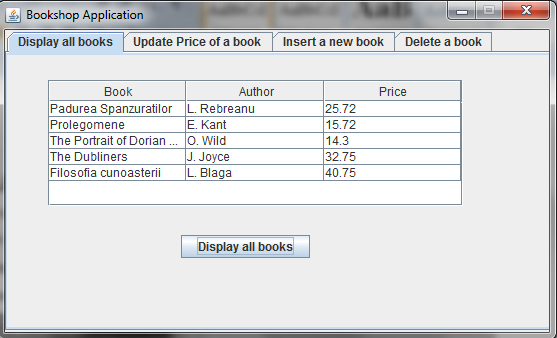
**Display All Books**

When the user first opens the Bookshop Application they are presented with the below screenshot under the Display all books tab. At the moment all fields in the table are blank.

By selecting the Display all books button the user will then be presented with a full list of all available books in the book store.

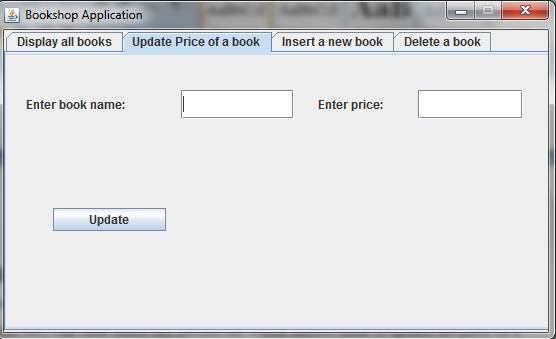
****

As you can see in the screenshot below when the user selects the Display all books button the information is retrieved from the database and all books are displayed in the GUI to the user. The books are displayed using the Book name, the Author name and Price.

****

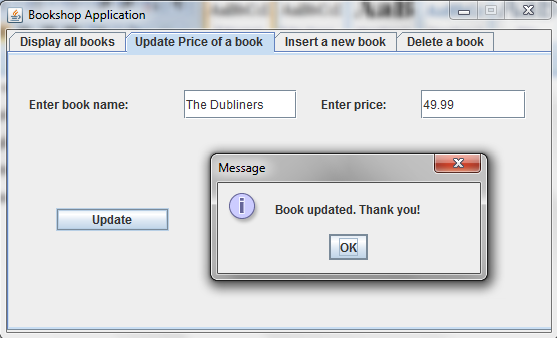
**Update price of a Book**

When the user has seen all the books available in the bookshop from the first tab in the JTabbedPane they can then select the second tab which allows them to update the price of a book. Please see screenshot below.

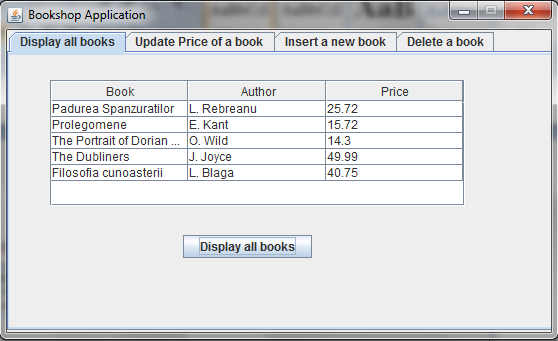


In this tab the user can simply type in the name of a book previously found in the display all books tab and then amend the price in the enter price field. By selecting the update button the price will then be updated to the new price.

For example, if the user would like to change the price of The Dubliners to 49.99 then they simply enter the book name and update the price in the relevant textfields. By clicking the update button this information is then updated in the database and the user will be notified through a showMessageDialog that the book has been updated. Please see screenshot below.

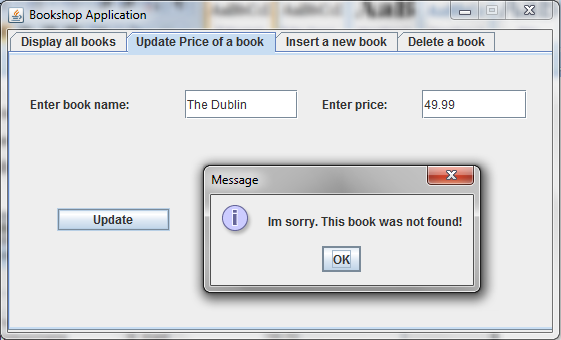


The user can go back into the Display all book tab, click the Display all books button and as you can see from the below screenshot on line four, The Dubliners price has been successfully updated to 49.99 from the original price of 32.75.



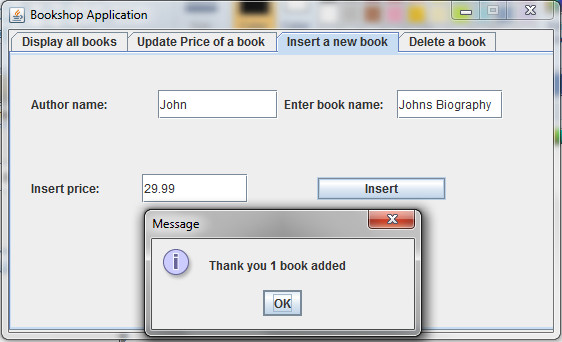
The next point to note is that if an incorrect name is entered or a book name that is not present in the bookshop then the user will be informed with a plain message that the book was not found. As you can see from the below screenshot there is no book called The Dublin. This name is searched for when the update button is selected and is nowhere to be found in the database and the user is informed accordingly.

If a book is not found a price cannot be updated so the user must be sure to enter a book name with the correct spelling and be sure the book is available from the first tab.

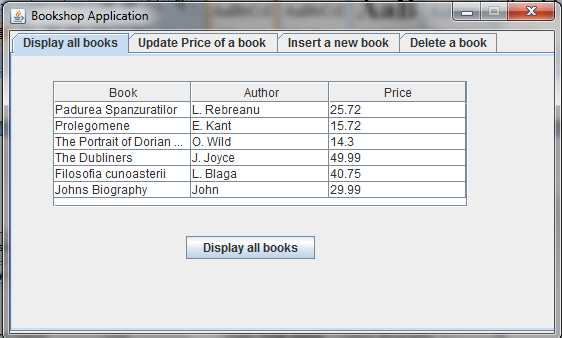


**Insert a new Book**

The third tab on the JTabbedPane is the Insert a new book tab. This allows the user to enter a new book into the database by entering the Author name, book name and price in the relevant textfields. When the user clicks the insert button the book is then added to the database with the relevant name, author and price. The user is also updated with a plain message to let them know that the book has been successfully added to the database. Please see screenshot below.

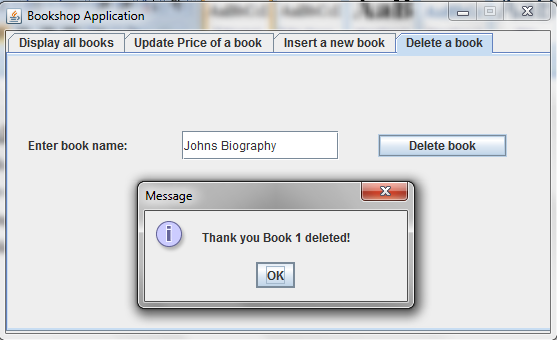


When you go back to the Display all books tab you can now see from the below screenshot that the book Johns Biography has been added to the database and is now available in the book store.

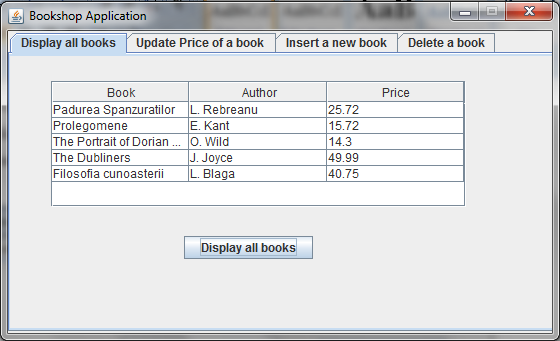


**Delete a Book**

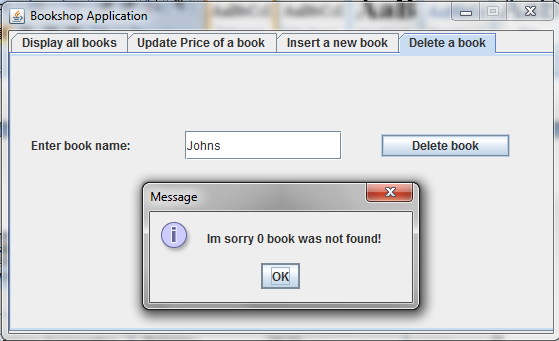
The fourth and final tab in the JTabbedPane is the Delete a book tab. This tab simply allows the user to delete a book from the database by entering the name of the book in the textfield and clicking the delete button. As you can see from the below screenshot we will delete the book Johns Bibliography. The user is then displayed a plain message that the book has been deleted from the database successfully.



We can then pop back into the Display all books tab and see that the book Johns Bibliography has been deleted from the database. Please see below screenshot.



If however, we enter an incorrect name, spelling or book name that is not present in the database then nothing will be found in the database or bookshop. The user will be displayed a pop up message informing them that no book with this name can be found. They must therefore check to ensure that they enter the correct name of a book that exists in the database in order to delete it successfully. Please see example screenshot below.



**Classes & Methods**

The three classes used for this project were class Book, class BookQueries and class BookApplication.

**Book class**

In the book class three variables were set up which were bookName, authorName and price which would interlink with the three fields in our database. In this class there is a constructor named public Book which takes in these three variables.

The next step involved laying out the setters setBookName, setAuthorName and setPrice which sets these three variables. Then the getters getBookName, getAuthorName and getPrice were set up in order to get the values assigned or passed to these variables.

Finally the toString method was used to return all these values from the getters. The toString method is used to get a string object from a number object. Our other classes access these variables through our setters and getters.

**BookQueries class**

The BookQueries class contains our connection parameters and prepared statements which will enable us to search for all books in the database, update the price of a certain book in the database, insert a new book into the database and finally delete a book from the database.

We use the try and catch to establish our connection to the bookshop database and also to initialise our preparedstatements for display, update, insert and delete functions.

If connection fails when we try this will throw and SQL exception.

The selectAllBooks method uses the displayAllBooks preparedStatement containing an SQL query to select all the books in the database and a resultset is returned. These are pulled from the getters already set in the Book class. All books are then displayed to the user as long as there are books available in the database.

The updateBook method uses the updatePrice preparedstatement from BookQueries above containing an SQL query and is used to update the price of a book. The book selected by the user is then updated with the price they input and a new price is then set to the book. If the update is executed successfully a 1 is returned otherwise a 0 is returned.

The insertBook method uses the insertNewBook preparedstatement from BookQueries above containing an SQL query and is used to add a new book to the database. When the user enters a book name, author and price to insert into the book shop database, and if it is updated and executed successfully, it will return a 1 to indicate the update was successful or a 0 if the update failed.

The deleteBook method also uses the deleteBook preparedstatement from BookQueries above containing an SQL query and is used to delete a book from the book shop application. When the user enters a name of a book they wish to delete and if the request is executed successfully then a 1 will be returned to indicate the book was deleted. If not then a 0 will be returned.

Finally the closeConnection method automatically closes the connection to the database when all queries are complete.

**BookApplication class**

The BookApplication class starts off by declaring all the necessary components required for the GUI and implements ActionListener which will allow us to use actionlisteners when clicking buttons and entering text. Here our GUI components are sized and designed to create their look and feel in the GUI using set bounds and sizes to position and size the components.

The first method organiseComponents is basically set up to initialise our GUI components. It starts by adding a panel to the JFrame. Then the main panel p1 is added to the frame.

Our first tab, tab1 is then added to the JTabbedPane which was declared as tp earlier at the top of the programme. This tab1 is the first tab that displays all the books using a table in the panel. The buttonDisplayAll is also added to tab1 at this point. By clicking the button all books in the database are displayed to the user.

Our next tab, tab2 is then added to the JTabbedPane tp. Two JLabels and two textFields are then added to enter a book name and the other to enter a price. Then a JButton is added to tab2 called buttonUpdatePrice which updated the price in the database when clicked.

Tab3 is then added to the JTabbedPane tp. Tab3 includes a JButton called buttonInsertBook which inserts a new book when clicked. Tab3 also contains three JPanels and three textfields to obtain user information for inserting a book.

Tab4 is the delete tab and is added to the JTabbedPane tb.Tab four contains one JPanel and one textfield where a user can enter the book name they wish to delete.

The next step in the code is the actionPerformed method which basically handles all our events such as when buttons are clicked.

This first thing here is the buttonDisplayAll. When this is clicked an event occurs and the table is displayed which retrieves all books in the database and the resultset is displayed in the table on the first tab.

The next event in line with the code is the buttonUpdatePrice. When this is clicked an event occurs which will take in the book name selected and the updated price from the user and change the price in the database accordingly. If the book price updates successfully a message is shown to the user through a showMessageDialog that the update was successful. Otherwise they will be displayed a showMessageDialog confirming the update was not successful.

The next event is the buttonInsertBook. When this event occurs from the user clicking the button the author name, book name and price entered by the user are taken in and inserted into the database and the booklist is updated. If this is successfully inserted the user will be displayed a message to say the update was successful. Otherwise they will be displayed a plain message to say the insert was unsuccessful. This can then be viewed in the display table.

The next event is the buttonDeleteBook which takes in the book name to be deleted from the database when the user has entered the book name and selected the delete button.

If the book name entered is found in the database is found it is deleted from the database and the user will be informed through a plain message that the book has been found.

Otherwise the user will be displayed a message letting them know that the book was not found in the database. This can then be viewed in the display all table to see the book disappear from the database.

The method displayTable is used to display the Book name, Author name and Price in a table when the user searches for all books in the book application. In here all books are selected form the selectAllBooks method in the BookQueries class. A for loop is then created to get the book name author, name and price which are then returned and displayed to the user.

**Conclustion & Recommendation**

I feel this is a very useful application to develop and I have learned a lot from completing this task. In terms of the application I would recommend a search option where books could be sorted by name, price or author in ascending or descending order. I think this could be a useful addition to the application. This could be possibly done by inserting a new panel or adding extra GUI features to the first tabs where it displays all books.

**Bibliography**

**Videos**

Videos include topics on objects, setters, getters, toString and GUI.

* <https://www.youtube.com/watch?v=eqP5X6APc5w&index=40&list=PLFE2CE09D83EE3E28>
* <https://www.youtube.com/watch?v=MK2SMJZbUmU&list=PLFE2CE09D83EE3E28&index=41>
* <https://www.youtube.com/watch?v=l0N6WvIVoUI&index=42&list=PLFE2CE09D83EE3E28>
* <https://www.youtube.com/watch?v=jJjg4JweJZU&index=50&list=PLFE2CE09D83EE3E28>
* <https://www.youtube.com/watch?v=jUdIAgJ7JKo&list=PLFE2CE09D83EE3E28&index=51>
* <https://www.youtube.com/watch?v=3EE7E3bvfe8&index=52&list=PLFE2CE09D83EE3E28>
* <https://www.youtube.com/watch?v=qhYook53olE&list=PLFE2CE09D83EE3E28&index=53>
* <https://www.youtube.com/watch?v=M1_-sigEPtE&list=PLFE2CE09D83EE3E28&index=54>
* <https://www.youtube.com/watch?v=6iV-v_m0z0w&index=62&list=PLFE2CE09D83EE3E28>
* <https://www.youtube.com/watch?v=3RQOikbGGUM&index=63&list=PLFE2CE09D83EE3E28>
* <https://www.youtube.com/watch?v=hsHqhX0s7Rs&index=74&list=PLFE2CE09D83EE3E28>
* <https://www.youtube.com/watch?v=MpIHF4V3zMc&index=75&list=PLFE2CE09D83EE3E28>

**Websites**

* <http://www.javatpoint.com/java-jbutton>
* <http://www.javatpoint.com/java-jlabel>
* <http://www.javatpoint.com/java-jtextfield>
* <http://www.javatpoint.com/java-jtable>
* <http://www.javatpoint.com/java-jscrollbar>
* <http://www.javatpoint.com/java-jtabbedpane>
* <http://www.javatpoint.com/java-jdialog>
* <http://www.javatpoint.com/java-jpanel>
* <http://www.javatpoint.com/understanding-toString()-method>
* <http://wideskills.com/java-tutorial/java-tostring-method>
* <http://wideskills.com/java-tutorial/java-jtextarea-class-example>
* <http://wideskills.com/java-tutorial/java-jtabbedpane-class-example>
* <http://wideskills.com/java-tutorial/java-button-class-example>
* <http://wideskills.com/java-tutorial/java-jlabel-class-example>
* <http://wideskills.com/java-tutorial/java-frame-class-example>

**Books**

* Java How to Program: Late Objects Version (8th Edition) 8th Edition

by [Paul Deitel](https://www.amazon.com/s/ref=dp_byline_sr_book_1?ie=UTF8&text=Paul+Deitel&search-alias=books&field-author=Paul+Deitel&sort=relevancerank) (Author)

**Moodle**

* Irene Murtage Lecture notes obtained from moodle ITB software engineering 1
* Aurelia Power Lecture notes and examples obtained from moodle ITB sorftware engineering 1 and 2
* Lab work from software engineering 1 and 2

**Other References**

* As well as the references mentioned above I also worked with a lot of my other class mates and a friend. Together we designed pseudo code to establish what way the code should be structured and how it could be displayed to suit the requirements of the task. We also helped each other out when we got compile errors. Class mate Melvin helped me with the code to design the table for the display all books.
* Java API

DROP DATABASE IF EXISTS bookshop;

CREATE DATABASE bookshop;

USE bookshop;

CREATE TABLE book

(

bookID int NOT NULL AUTO\_INCREMENT PRIMARY KEY,

bookName varchar(30) NOT NULL,

authorName varchar(30) NOT NULL,

price double(5, 2)

);

INSERT INTO book (bookName, authorName, price) VALUES ('Padurea Spanzuratilor', 'L. Rebreanu', 25.72);

INSERT INTO book (bookName, authorName, price) VALUES ('Prolegomene', 'E. Kant', 15.72);

INSERT INTO book (bookName, authorName, price) VALUES ('The Portrait of Dorian Gray', 'O. Wild', 14.30);

INSERT INTO book (bookName, authorName, price) VALUES ('The Dubliners', 'J. Joyce', 32.75);

INSERT INTO book (bookName, authorName, price) VALUES ('Filosofia cunoasterii', 'L. Blaga', 40.75);