Even Semester (2015)

****

**BINUS UNIVERSITY**

# BINUS INTERNATIONAL

### Assignment Cover Letter

**(Group Work)**

**Student Information**: ***Surname Given Names Student ID Number***

1. *Darmawan Albert 1801433554*
2. *Kharli Matthew Brian Teguh 1801436575*
3. *Larosa Elbert Christopher 1801442464*

**Course Code : COMP6213 Course Name : Object Oriented Programming**

**Class : L2AC Name of Lecturer(s) :** Tri Asih Budiono

**Major : Computer Science**

**Title of Assignment** : Picadilly Bookstore Management System

**Type of Assignment : Final Project**

**Submission Pattern : Soft copy of the external documentation and Java source code files.**

**Due Date : 15 June 2015 Submission Date : 15 June 2015**

The assignment should meet the below requirements.

1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer’s instructions.
2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3. The above information is complete and legible.
4. Compiled pages are firmly stapled.
5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

## Plagiarism/Cheating

BiNus International seriously regards all forms of plagiarism, cheating and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

**Declaration of Originality**

By signing this assignment, I understand, accept and consent to BiNus International terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

*Signature of Student: (Name of Student)*

*1. Albert Darmawan*

*2. Matthew Brian Teguh Kharli*

*3. Elbert Christopher Larosa*

**Table of Contents**

Title ...........................................................................................................................................1

Requirements / Specifications ...................................................................................................2

Design ........................................................................................................................................3

Limitations ................................................................................................................................4

Brief User Guide .......................................................................................................................4

Listings ......................................................................................................................................6

Testing ...................................................................................................................................115

**Title**

The title of this project is Picadilly Bookstore Management System. This project is made by Albert Darmawan, Elbert Christopher Larosa, and Matthew Brian Teguh Kharli. The project is finished on June 15th, 2015. This project consists of 5 packages, which are:

1. picadillybookstore

This package has a main file called PicadillyBookStore.java. The fourth step (according to “Guide to Completing COMP6213 The Final Project”) can be seen by running this main file. This main file already connects with DAO (Data Access Object).

1. picadillybookstore.dao

The main purpose of the files inside this package is to connect Persistent Data Storage (XML file) with the data from the Model. This package has 5 class files, which are:

* Admins.java
* Books.java
* Customers.java
* DAO.java
* Transactions.java

1. picadillybookstore.domain

All files inside this package act as Model in MVC (Model-View-Controller) software architectural pattern. This package has 4 class files and 4 enumeration files, which are:

* Admin.java
* Binding.java
* Book.java
* Customer.java
* Gender.java
* PurchaseStatus.java
* Signed.java
* Transaction.java

1. picadillybookstore.gui

This package has a main file called PicadillyBookstoreGUI.java. The fifth and the sixth step of this project (according to “Guide to Completing COMP6213 The Final Project”) can be seen by running this main file. This file acts as View in MVC software architectural pattern. Java Swing library is used for this project.

1. picadillybookstore.step3

This package contains a main file called PicadillyBookStoreStepThree.java. The third step of this project (according to “Guide to Completing COMP6213 The Final Project”) can be seen by running this main file.

JAXB (Java Architecture for XML Binding) is used for this project. The XML (Extensible Markup Language) files for this project are:

* admin.xml
* book.xml
* customer.xml
* transaction.xml

The purpose of this project is to create a bookstore management system that gives an easy access for admins to manage the bookstore and for customers to buy books from the bookstore.

**Requirements / Specification**

The program’s requirements and specification will be explained based on the two users, which are:

1. Administrator (Admin)

For admin, the program should enable the Admin to add books (by inputting the book’s info), modify books, and delete books. Secondly, the program should enable the Admin to search and view books that are already added. Lastly, the program should also be able to keep records of books, customers, and transactions into the XML file.

1. Customer

For customer, the program should enable customer to sign up (by inputting the customer’s own personal info, account info, and credit card info). Secondly, the program should enable customer to modify own’s profile. Thirdly, the program should enable customer to search and view books that already have been added by the Admin. Lastly, the program should also enable customer to purchase books, which will create transaction records that should be saved into the XML file.

**Design**

This is the UML design for the domain classes. Setter and getter methods are not included in this design.

|  |
| --- |
| **Customer** |
| - userId: long  - name: String  - gender: Gender  - username: String  - password: String  - email: String  - phoneNumber: String  - numberOfPruchase: int  - streetAddress: String  - city: String  - zipCode: String  - country: String  - creditCardName: String  - creditCardNo: String  - cvc: String  - creditCardMonth: String  - creditCardYear: String  - purchasedBook: ArrayList<Book> |
| **+** isCreditCardNumberVaild(): boolean  + isCreditCardNotExpired(): boolean  + purchaseBook(Book bookBeingPurchased): PurchaseStatus  **+** display(): void |

|  |
| --- |
| **Transaction** |
| - transactionId: long  - customerId: long  - isbn: String  - quantity: long  - date: Date  - paymentVerification: boolean |
| **+** display(): void |

|  |
| --- |
| **Admin** |
| - userId: long  - name: String  - gender: Gender  - username: String  - password: String  - email: String  - phoneNumber: String |
| **+** display(): void |

|  |
| --- |
| **Book** |
| - author: String  - title: String  - isbn: String  - price: long  - quantity: long  - publisherName: String  - yearPublished: int  - description: String  - binding: Binding  - signed: Signed  - category: List  - keywords: List |
| **+** display(): void |

*establishes*

|  |
| --- |
| **<<enumeration>>**  **Binding** |
| HARDCOVER  PAPERBACK |

|  |
| --- |
| **<<enumeration>>**  **Signed** |
| SIGNED  NOTSIGNED |

|  |
| --- |
| **<<enumeration>>**  **Gender** |
| MALE  FEMALE |

|  |
| --- |
| **<<enumeration>>**  **PurchaseStatus** |
| success  noStock  creditCardInvalid |

**Limitations**

The features that are asked for but not implemented in this project are:

* Modify book (as admin)
* Modify customer’s own profile (as customer)
* Search book and customer (as admin)
* Search book (as customer)

Other than those features, all other features are already implemented in this project. Another shortfalls of this project (outside of the project’s requirements) are that there is no input verification in this program, so bad inputs are still taken. The passwords in this program are also not encrypted, which means that the program is likely to be unsecure. The last shortfall would be that the GUI for this program is not in the consistent layout, such as that some of the parts may change its position as the size of the window is being changed manually.

**Brief User Guide**

The user guide is going to be explained based on the steps according to “Guide to Completing COMP6213 The Final Project”:

1. Step 3: Hard-coded data

To run the program at this step, right click at PicadillyBookStoreStepThree.java file (inside picadillybookstore.step3 package) and then choose Run File. The program will then run and show the book list, customer list, and admin list, also cases of transaction which are all hard-coded data.

1. Step 4: Data Access Object

To run the program at this step, right click at PicadillyBookStore.java file (inside picadillybookstore package) and then choose Run File. The program will display the same contents as in the step 3. The difference here is that this time, the data is not hard-coded. Instead, the data is stored inside the XML files and are retrieved from the XML files when it is going to be used. An exception has already been made to handle files that may not be found.

1. Step 5: Interface or GUI (with Step 6: The Administrator Class included)

To run the program at this step, right click at PicadillyBookstoreGUI.java (inside picadillybookstore.gui package) and then choose Run File. The User Choice panel will then appear, where the user can choose if the user is an admin or a customer.

* Admin

After choosing Admin, the Admin Login panel will appear, where the Admin has to login using the username and password that are already stored in the XML files. After logging in, the Admin’s main menu will appear where the Admin can choose to look at the books list, the customers list, or the transactions list. In the Books List panel, the Admin can add new books, view books, and delete books. In the Customers List panel, the Admin can view customers and their info. Lastly, in the Transactions List panel, the Admin can view the transactions that have already been done.

* Customer

After choosing Customer, the Customer Login panel will appear, where the customer can login if the customer already signed up. The customer can also sign up by filling the personal info, account info, and credit card info form. The customer’s credit card is verified by using Luhn’s algorithm. After logging in, the customer’s main menu will appear where the customer can view books and purchase books.

**Listings**

Code in PicadillyBookStore.java (inside picadillybookstore package):

1 package picadillybookstore;

2

3 import java.util.ArrayList;

4 import java.util.List;

5 import java.util.Arrays;

6 import java.util.Date;

7

8 import picadillybookstore.domain.Admin;

9 import picadillybookstore.domain.Binding;

10 import picadillybookstore.domain.Book;

11 import picadillybookstore.domain.Customer;

12 import picadillybookstore.domain.Gender;

13 import picadillybookstore.domain.Signed;

14 import picadillybookstore.domain.Transaction;

15 import picadillybookstore.domain.PurchaseStatus;

16 import picadillybookstore.dao.DAO;

17 import picadillybookstore.dao.Books;

18 import picadillybookstore.dao.Customers;

19 import picadillybookstore.dao.Admins;

20 import picadillybookstore.dao.Transactions;

21

22 /\*\*

23 \* **PicadillyBookStore.java**

24 \* **This** **class** **is** **the** **container** **of** **main** **function**

25 \* **@author** Albert Darmawan

26 \* **@version** 1.0

27 \* **@since** 2015-05-26

28 \*/

29 public class **PicadillyBookStore** {

30 //------------

31 // Array List Initialization

32 //------------

33 public static ArrayList<Book> *bookCatalog*;

34 public static ArrayList<Customer> *customerList*;

35 public static ArrayList<Admin> *adminList*;

36 public static ArrayList<Transaction> *transactionHistory*;

37 public static int *numberOfTransaction* = 0;

38

39 /\*\*

40 \*

41 \* **@param** args

42 \*/

43

44 //------------

45 // Main Functions

46 //------------

47 public static void ***main***(String[] args) {

48 // Object variable declaration and initialization

49 Book b1 = null, b2 = null, b3 = null;

50 Customer c1 = null, c2 = null;

51 Admin a1 = null, a2 = null;

52

53 // Retrieve data from book.xml and put it in Book variables

54 Books bookCollection = (Books) DAO.*getAllBooks*();

55 int booksLoopCounter = 1;

56 for (Book bookChecked : bookCollection.getBooks())

57 {

58 switch(booksLoopCounter)

59 {

60 case 1:

61 b1 = bookChecked;

62 break;

63 case 2:

64 b2 = bookChecked;

65 break;

66 case 3:

67 b3 = bookChecked;

68 break;

69 }

70 booksLoopCounter++;

71 }

72

73 // Adding 3 books to Book array list

74 *bookCatalog* = new ArrayList<>();

75 *bookCatalog*.add(b1);

76 *bookCatalog*.add(b2);

77 *bookCatalog*.add(b3);

78

79 // Retrieve data from customer.xml and put it in Customer variables

80 Customers customerCollection = (Customers) DAO.*getAllCustomers*();

81 int customersLoopCounter = 1;

82 for (Customer customerChecked : customerCollection.getCustomers())

83 {

84 switch(customersLoopCounter)

85 {

86 case 1:

87 c1 = customerChecked;

88 break;

89 case 2:

90 c2 = customerChecked;

91 break;

92 }

93 customersLoopCounter++;

94 }

95

96

97 // Adding 2 customers to Customer array list

98 *customerList* = new ArrayList<>();

99 *customerList*.add(c1);

100 *customerList*.add(c2);

101

102

103 // Retrieve data from admin.xml and put it in Admin variables

104 Admins adminCollection = (Admins) DAO.*getAllAdmins*();

105 int adminsLoopCounter = 1;

106 for (Admin adminChecked : adminCollection.getAdmins())

107 {

108 switch(adminsLoopCounter)

109 {

110 case 1:

111 a1 = adminChecked;

112 break;

113 case 2:

114 a2 = adminChecked;

115 break;

116 }

117 adminsLoopCounter++;

118 }

119

120 // Adding 2 admins to Admin array list

121 *adminList* = new ArrayList<>();

122 *adminList*.add(a1);

123 *adminList*.add(a2);

124

125 // Creating Transaction array List

126 *transactionHistory* = new ArrayList<>();

127

128 // Outputing Title

129 System.*out*.println("================================================================");

130 System.*out*.println(" Welcome to Picadilly Book Store! **\n\t**Where customer's satisfaction is our main priority");

131 System.*out*.println("================================================================");

132

133 // Displaying Books List

134 System.*out*.println("================================================================");

135 System.*out*.println(" Book List");

136 System.*out*.println("================================================================");

137

138 for (Book eachBook : *bookCatalog*) {

139 eachBook.display();

140 }

141

142 // Displaying Customer List

143 System.*out*.println("================================================================");

144 System.*out*.println(" Customer List");

145 System.*out*.println("================================================================");

146

147 for (Customer eachCustomer : *customerList*) {

148 eachCustomer.display();

149 }

150

151 // Displaying Admin List

152 System.*out*.println("================================================================");

153 System.*out*.println(" Admin List");

154 System.*out*.println("================================================================");

155

156 for (Admin eachAdmin : *adminList*) {

157 eachAdmin.display();

158 }

159

160 // Customers start to purchase books

161 System.*out*.println("================================================================");

162 System.*out*.println(" Book Purchasing");

163 System.*out*.println("================================================================");

164 *attemptToPurchaseBook*(c1,b1);

165 *attemptToPurchaseBook*(c2,b2);

166

167 // Transaction History

168 System.*out*.println("================================================================");

169 System.*out*.println(" Transaction History");

170 System.*out*.println("================================================================");

171

172 for (Transaction eachTransaction : *transactionHistory*) {

173 eachTransaction.display();

174 }

175 }

176

177 //------------

178 // Function to manage book purchasing

179 //------------

180 private static void ***attemptToPurchaseBook***(Customer customerInvolved, Book bookInvolved)

181 {

182 Transaction currentTransaction;

183 Date currentDate;

184 System.*out*.println(customerInvolved.getName()+" is trying to buy **\"**" + bookInvolved.getTitle() + "**\"**");

185 PurchaseStatus thisPurchaseStatus = customerInvolved.purchaseBook(bookInvolved);

186 System.*out*.println(thisPurchaseStatus.value());

187 *numberOfTransaction*++;

188 // Create a new Date object

189 currentDate = new Date();

190 // Create a new Transaction object

191 currentTransaction = new Transaction();

192 currentTransaction.setTransactionId(*numberOfTransaction*);

193 currentTransaction.setCustomerId(customerInvolved.getUserId());

194 currentTransaction.setIsbn(bookInvolved.getIsbn());

195 currentTransaction.setQuantity(1);

196 currentTransaction.setDate(currentDate);

197 if (thisPurchaseStatus== PurchaseStatus.*success*)

198 {

199 // Updating book and customer information when the purchasing is successful

200 DAO.*replaceBookData*(bookInvolved);

201 DAO.*replaceCustomerData*(customerInvolved);

202 currentTransaction.setPaymentVerification(true);

203 System.*out*.println("Remaining books titled **\"**"+bookInvolved.getTitle()+"**\"**" + " : " + bookInvolved.getQuantity());

204 }

205 else {

206 currentTransaction.setPaymentVerification(false);

207 }

208

209 // Add currentTransaction to transactionHistory ArrayList

210 *transactionHistory*.add(currentTransaction);

211

212 // Add currentTransaction to transaction.xml

213 DAO.*addTransaction*(currentTransaction);

214 System.*out*.println("");

215 }

216 }

Code in Admins.java (inside picadillybookstore.dao package):

1 /\*

2 \* To change this license header, choose License Headers in Project Properties.

3 \* To change this template file, choose Tools | Templates

4 \* and open the template in the editor.

5 \*/

6 package picadillybookstore.dao;

7 import java.util.List;

8 import javax.xml.bind.annotation.XmlAccessType;

9 import javax.xml.bind.annotation.XmlAccessorType;

10 import javax.xml.bind.annotation.XmlElement;

11 import javax.xml.bind.annotation.XmlRootElement;

12 import picadillybookstore.domain.Admin;

13 /\*\*

14 \* **Admins.java**

15 \* **This** **class** **is** **the** **container** **of** **Admin** **List<>**

16 \* **@author** Albert Darmawan

17 \* **@version** 1.0

18 \* **@since** 2015-06-13

19 \*/

20 @XmlRootElement(name="admins")

21 @XmlAccessorType(XmlAccessType.*FIELD*)

22 public class **Admins** {

23 //------------

24 // Attribute

25 //------------

26 @XmlElement(name="admin")

27 private List<Admin> admins = null;

28 /\*\*

29 \*

30 \* **@return**

31 \*/

32

33 //------------

34 // Accessor

35 //------------

36 public List<Admin> **getAdmins**()

37 {

38 return admins;

39 }

40

41 //------------

42 // Mutator

43 //------------

44 public void **setAdmins**(List<Admin> admins)

45 {

46 this.admins =admins;

47 }

48 }

49

Code in Books.java (inside picadillybookstore.dao package):

1 /\*

2 \* To change this license header, choose License Headers in Project Properties.

3 \* To change this template file, choose Tools | Templates

4 \* and open the template in the editor.

5 \*/

6 package picadillybookstore.dao;

7 import java.util.List;

8 import javax.xml.bind.annotation.XmlAccessType;

9 import javax.xml.bind.annotation.XmlAccessorType;

10 import javax.xml.bind.annotation.XmlElement;

11 import javax.xml.bind.annotation.XmlRootElement;

12 import picadillybookstore.domain.Book;

13 /\*\*

14 \* **Books.java**

15 \* **This** **class** **is** **the** **container** **of** **Book** **List<>**

16 \* **@author** Albert Darmawan

17 \* **@version** 1.0

18 \* **@since** 2015-06-13

19 \*/

20 @XmlRootElement(name="books")

21 @XmlAccessorType(XmlAccessType.*FIELD*)

22 public class **Books** {

23 //------------

24 // Attribute

25 //------------

26 @XmlElement(name="book")

27 private List<Book> books = null;

28

29 //------------

30 // Accessor

31 //------------

32 public List<Book> **getBooks**()

33 {

34 return books;

35 }

36

37 //------------

38 // Mutator

39 //------------

40 public void **setBooks**(List<Book> books)

41 {

42 this.books =books;

43 }

44 }

45

Code in Customers.java (inside picadillybookstore.dao package):

1 /\*

2 \* To change this license header, choose License Headers in Project Properties.

3 \* To change this template file, choose Tools | Templates

4 \* and open the template in the editor.

5 \*/

6 package picadillybookstore.dao;

7

8 import java.util.List;

9 import javax.xml.bind.annotation.XmlAccessType;

10 import javax.xml.bind.annotation.XmlAccessorType;

11 import javax.xml.bind.annotation.XmlElement;

12 import javax.xml.bind.annotation.XmlRootElement;

13 import picadillybookstore.domain.Customer;

14 /\*\*

15 \* **Customers.java**

16 \* **This** **class** **is** **the** **container** **of** **Customer** **List<>**

17 \* **@author** Albert Darmawan

18 \* **@version** 1.0

19 \* **@since** 2015-06-06

20 \*/

21 @XmlRootElement(name="customers")

22 @XmlAccessorType(XmlAccessType.*FIELD*)

23 public class **Customers** {

24 //------------

25 // Attribute

26 //------------

27 @XmlElement(name="customer")

28 private List<Customer> customers;

29 /\*\*

30 \*

31 \* **@return**

32 \*/

33

34 //------------

35 // Accessor

36 //------------

37 public List<Customer> **getCustomers**()

38 {

39 return customers;

40 }

41

42 //------------

43 // Mutator

44 //------------

45 public void **setCustomers**(List<Customer> customers)

46 {

47 this.customers =customers;

48 }

49 }

50

Code in DAO.java (inside picadillybookstore.dao package):

1 /\*

2 \* To change this license header, choose License Headers in Project Properties.

3 \* To change this template file, choose Tools | Templates

4 \* and open the template in the editor.

5 \*/

6 package picadillybookstore.dao;

7

8 import java.io.File;

9 import java.util.ArrayList;

10 import java.util.List;

11 import picadillybookstore.domain.Book;

12 import picadillybookstore.domain.Customer;

13 import picadillybookstore.domain.Admin;

14 import picadillybookstore.domain.Transaction;

15 import javax.xml.bind.JAXBContext;

16 import javax.xml.bind.JAXBException;

17 import javax.xml.bind.Marshaller;

18 import javax.xml.bind.Unmarshaller;

19

20 /\*\*

21 \* **DAO.java**

22 \* **This** **class** **contains** **every** **function** **related** **to** **accessing** **data** **from** **XML** **and** **storing** **data** **to** **XML**

23 \*

24 \* **@author** Albert Darmawan

25 \* **@version** 1.0

26 \* **@since** 2015-06-13

27 \*/

28 public class **DAO** {

29 //----------------

30 // Book-related functions

31 //----------------

32

33 public static Books ***getAllBooks***()

34 {

35 Books bookListReturned;

36 bookListReturned = null;

37 try {

38 File bookFile = new File("book.xml");

39

40 JAXBContext jaxbContext = JAXBContext.*newInstance*(Books.class);

41 // Unmarshaling (convert data from XML into Java Object)

42 Unmarshaller jaxbUnmarshaller;

43 jaxbUnmarshaller = jaxbContext.createUnmarshaller();

44 bookListReturned = (Books) jaxbUnmarshaller.unmarshal(bookFile);

45

46 }

47 catch (JAXBException e)

48 {

49 System.*out*.println("JAXB error, " + e.getMessage() );

50

51 }

52 return bookListReturned;

53 }

54

55 public static Book ***searchBook***(String isbn)

56 {

57 Books bookCollection = (Books) DAO.*getAllBooks*();

58 Book bookReturned = null;

59 // sequential search. Complexity : O(n)

60 for (Book bookChecked : bookCollection.getBooks())

61 {

62 if (bookChecked.getIsbn().equals(isbn))

63 {

64 bookReturned = bookChecked;

65 }

66 }

67 return bookReturned;

68 }

69

70 public static void ***addBook***(Book bookAdded)

71 {

72 Books bookCollection = (Books) DAO.*getAllBooks*();

73 bookCollection.getBooks().add(bookAdded);

74 try {

75 File bookFile = new File("book.xml");

76 // Marshaling (Converting Java Object into XML file)

77 JAXBContext jaxbContext = JAXBContext.*newInstance*(Books.class);

78 Marshaller jaxbMarshaller = jaxbContext.createMarshaller();

79 // output pretty printed

80 jaxbMarshaller.setProperty(Marshaller.*JAXB\_FORMATTED\_OUTPUT*, true);

81 jaxbMarshaller.marshal(bookCollection, bookFile);

82 }

83 catch (JAXBException e)

84 {

85 System.*out*.println("JAXB error, " + e.getMessage() );

86 }

87 }

88

89 public static void ***deleteBook***(String isbn)

90 {

91 Books bookCollection = (Books) DAO.*getAllBooks*();

92 Books newBookCollection = new Books();

93 newBookCollection.setBooks(new ArrayList<Book>());

94 try {

95 File bookFile = new File("book.xml");

96 for (Book bookChecked : bookCollection.getBooks())

97 {

98 if (!(bookChecked.getIsbn().equals(isbn)))

99 {

100 newBookCollection.getBooks().add(bookChecked);

101 }

102 }

103 // Marshalling (Converting Java Object into XML file)

104 JAXBContext jaxbContext = JAXBContext.*newInstance*(Books.class);

105 Marshaller jaxbMarshaller = jaxbContext.createMarshaller();

106 // output pretty printed

107 jaxbMarshaller.setProperty(Marshaller.*JAXB\_FORMATTED\_OUTPUT*, true);

108 jaxbMarshaller.marshal(newBookCollection, bookFile);

109 }

110 catch (JAXBException e)

111 {

112 System.*out*.println("JAXB error, " + e.getMessage() );

113 }

114 }

115

116 public static void ***replaceBookData***(Book bookEdited)

117 {

118 Books bookCollection = (Books) DAO.*getAllBooks*();

119 String bookCheckedIsbn;

120 String bookEditedIsbn = bookEdited.getIsbn();

121 Books newBookCollection = new Books();

122 newBookCollection.setBooks(new ArrayList<Book>());

123 try {

124 File bookFile = new File("book.xml");

125 for (Book bookChecked : bookCollection.getBooks())

126 {

127 bookCheckedIsbn = bookChecked.getIsbn();

128 if (bookCheckedIsbn.equals(bookEditedIsbn))

129 {

130 newBookCollection.getBooks().add(bookEdited);

131 }

132 else {

133 newBookCollection.getBooks().add(bookChecked);

134 }

135 }

136 // Marshalling (Converting Java Object into XML file)

137 JAXBContext jaxbContext = JAXBContext.*newInstance*(Books.class);

138 Marshaller jaxbMarshaller = jaxbContext.createMarshaller();

139 // output pretty printed

140 jaxbMarshaller.setProperty(Marshaller.*JAXB\_FORMATTED\_OUTPUT*, true);

141 jaxbMarshaller.marshal(newBookCollection, bookFile);

142 }

143 catch (JAXBException e)

144 {

145 System.*out*.println("JAXB error, " + e.getMessage() );

146 }

147 }

148

149 //----------------

150 // Customer-related functions

151 //----------------

152

153 public static Customers ***getAllCustomers***()

154 {

155 Customers customerListReturned;

156 customerListReturned = null;

157 try {

158 File customerFile = new File("customer.xml");

159

160 JAXBContext jaxbContext = JAXBContext.*newInstance*(Customers.class);

161 // Unmarshaling (Converting XML file into Java Object)

162 Unmarshaller jaxbUnmarshaller;

163 jaxbUnmarshaller = jaxbContext.createUnmarshaller();

164 customerListReturned = (Customers) jaxbUnmarshaller.unmarshal(customerFile);

165 }

166 catch (JAXBException e)

167 {

168 System.*out*.println("JAXB error, " + e.getMessage() );

169 }

170 return customerListReturned;

171 }

172

173 public static Customer ***searchCustomer***(String username)

174 {

175 Customers customerList = (Customers) DAO.*getAllCustomers*();

176 Customer customerReturned = null;

177 // sequential search. Complexity : O(n)

178 for (Customer customerChecked : customerList.getCustomers())

179 {

180 if (customerChecked.getUsername().equals(username))

181 {

182 customerReturned = customerChecked;

183 }

184 }

185 return customerReturned;

186 }

187

188 public static void ***addCustomer***(Customer customerAdded)

189 {

190 Customers customerList = (Customers) DAO.*getAllCustomers*();

191 customerList.getCustomers().add(customerAdded);

192 try {

193 File customerFile = new File("customer.xml");

194 // Marshalling (Converting Java Object into XML file)

195 JAXBContext jaxbContext = JAXBContext.*newInstance*(Customers.class);

196 Marshaller jaxbMarshaller = jaxbContext.createMarshaller();

197 // output pretty printed

198 jaxbMarshaller.setProperty(Marshaller.*JAXB\_FORMATTED\_OUTPUT*, true);

199 jaxbMarshaller.marshal(customerList, customerFile);

200 }

201 catch (JAXBException e)

202 {

203 System.*out*.println("JAXB error, " + e.getMessage() );

204 }

205 }

206

207 public static void ***replaceCustomerData***(Customer customerEdited)

208 {

209 Customers customerCollection = (Customers) DAO.*getAllCustomers*();

210 long customerCheckedUserId;

211 long customerEditedUserId = customerEdited.getUserId();

212 Customers newCustomerCollection = new Customers();

213 newCustomerCollection.setCustomers(new ArrayList<Customer>());

214 try {

215 File customerFile = new File("customer.xml");

216 for (Customer customerChecked : customerCollection.getCustomers())

217 {

218 customerCheckedUserId = customerChecked.getUserId();

219 if (customerCheckedUserId == customerEditedUserId)

220 {

221 newCustomerCollection.getCustomers().add(customerEdited);

222 }

223 else {

224 newCustomerCollection.getCustomers().add(customerChecked);

225 }

226 }

227 // Marshalling (Converting Java Object into XML file)

228 JAXBContext jaxbContext = JAXBContext.*newInstance*(Customers.class);

229 Marshaller jaxbMarshaller = jaxbContext.createMarshaller();

230 // output pretty printed

231 jaxbMarshaller.setProperty(Marshaller.*JAXB\_FORMATTED\_OUTPUT*, true);

232 jaxbMarshaller.marshal(newCustomerCollection, customerFile);

233 }

234 catch (JAXBException e)

235 {

236 System.*out*.println("JAXB error, " + e.getMessage() );

237 }

238 }

239

240 public static long ***numberOfCustomers***()

241 {

242 Customers customerCollection = (Customers) DAO.*getAllCustomers*();

243 long customerCount = 0;

244 for (Customer customerChecked : customerCollection.getCustomers())

245 {

246 customerCount++;

247 }

248 return customerCount;

249 }

250

251 //----------------

252 // Admin-related functions

253 //----------------

254 public static Admins ***getAllAdmins***()

255 {

256 Admins adminListReturned;

257 adminListReturned = null;

258 try {

259 File adminFile = new File("admin.xml");

260

261 JAXBContext jaxbContext = JAXBContext.*newInstance*(Admins.class);

262 // Unmarshaling (Converting XML file into Java Object)

263 Unmarshaller jaxbUnmarshaller;

264 jaxbUnmarshaller = jaxbContext.createUnmarshaller();

265 adminListReturned = (Admins) jaxbUnmarshaller.unmarshal(adminFile);

266 }

267 catch (JAXBException e)

268 {

269 System.*out*.println("JAXB error, " + e.getMessage() );

270 }

271 return adminListReturned;

272 }

273

274 public static void ***addAdmin***(Admin adminAdded)

275 {

276 Admins adminList = (Admins) DAO.*getAllAdmins*();

277 adminList.getAdmins().add(adminAdded);

278 try {

279 File adminFile = new File("admin.xml");

280 // Marshalling (Converting Java Object into XML file)

281 JAXBContext jaxbContext = JAXBContext.*newInstance*(Admins.class);

282 Marshaller jaxbMarshaller = jaxbContext.createMarshaller();

283 // output pretty printed

284 jaxbMarshaller.setProperty(Marshaller.*JAXB\_FORMATTED\_OUTPUT*, true);

285 jaxbMarshaller.marshal(adminList, adminFile);

286 }

287 catch (JAXBException e)

288 {

289 System.*out*.println("JAXB error, " + e.getMessage() );

290 }

291 }

292

293 //----------------

294 // Transaction-related functions

295 //----------------

296 public static Transactions ***getAllTransactions***()

297 {

298 Transactions transactionListReturned;

299 transactionListReturned = null;

300 try {

301 File transactionFile = new File("transaction.xml");

302

303 JAXBContext jaxbContext = JAXBContext.*newInstance*(Transactions.class);

304 // Unmarshaling (Converting XML file into Java Object)

305 Unmarshaller jaxbUnmarshaller;

306 jaxbUnmarshaller = jaxbContext.createUnmarshaller();

307 transactionListReturned = (Transactions) jaxbUnmarshaller.unmarshal(transactionFile);

308 }

309 catch (JAXBException e)

310 {

311 System.*out*.println("JAXB error, " + e.getMessage() );

312 }

313 return transactionListReturned;

314 }

315

316 public static void ***addTransaction***(Transaction transactionAdded)

317 {

318 Transactions transactionList = (Transactions) DAO.*getAllTransactions*();

319 transactionList.getTransactions().add(transactionAdded);

320 try {

321 File transactionFile = new File("transaction.xml");

322 // Marshalling (Converting Java Object into XML file)

323 JAXBContext jaxbContext = JAXBContext.*newInstance*(Transactions.class);

324 Marshaller jaxbMarshaller = jaxbContext.createMarshaller();

325 // output pretty printed

326 jaxbMarshaller.setProperty(Marshaller.*JAXB\_FORMATTED\_OUTPUT*, true);

327 jaxbMarshaller.marshal(transactionList, transactionFile);

328 }

329 catch (JAXBException e)

330 {

331 System.*out*.println("JAXB error, " + e.getMessage() );

332 }

333 }

334

335 public static long ***numberOfTransaction***() {

336 Transactions transactionCollection = (Transactions) DAO.*getAllTransactions*();

337 long transactionCount = 0;

338 for (Transaction transactionChecked : transactionCollection.getTransactions())

339 {

340 transactionCount++;

341 }

342 return transactionCount;

343 }

344 }

345

Code in Transactions,java (inside picadillybookstore.dao package):

1 /\*

2 \* To change this license header, choose License Headers in Project Properties.

3 \* To change this template file, choose Tools | Templates

4 \* and open the template in the editor.

5 \*/

6 package picadillybookstore.dao;

7 import java.util.List;

8 import javax.xml.bind.annotation.XmlAccessType;

9 import javax.xml.bind.annotation.XmlAccessorType;

10 import javax.xml.bind.annotation.XmlElement;

11 import javax.xml.bind.annotation.XmlRootElement;

12 import picadillybookstore.domain.Transaction;

13 /\*\*

14 \* **Transactions.java**

15 \* **This** **class** **is** **the** **container** **of** **Transaction** **List<>**

16 \* **@author** Albert Darmawan

17 \* **@version** 1.0

18 \* **@since** 2015-06-14

19 \*/

20

21 @XmlRootElement(name="transactions")

22 @XmlAccessorType(XmlAccessType.FIELD)

23 public class Transactions {

24 //------------

25 // Attribute

26 //------------

27 @XmlElement(name="transaction")

28 private List<Transaction> transactions = null;

29

30 //------------

31 // Accessor

32 //------------

33 public List<Transaction> getTransactions()

34 {

35 return transactions;

36 }

37

38 //------------

39 // Mutator

40 //------------

41 public void setTransactions(List<Transaction> transactions)

42 {

43 this.transactions =transactions;

44 }

45 }

46

Code in Admin.java (inside picadillybookstore.domain package):

1 package picadillybookstore.domain;

2

3 import javax.xml.bind.annotation.XmlRootElement;

4 import javax.xml.bind.annotation.XmlAttribute;

5 import javax.xml.bind.annotation.XmlElement;

6 /\*\*

7 \* **Admin.java**

8 \* **This** **class** **encapsulates** **all** **attributes** **that** **an** **administrator** **has**

9 \* **@author** Albert Darmawan

10 \* **@version** 1.0

11 \* **@since** 2015-05-26

12 \*/

13

14 @XmlRootElement(name="admin")

15 public class Admin {

16 //------------

17 // Attributes.

18 //------------

19 private long userId;

20 private String name;

21 private Gender gender;

22 private String username;

23 private String password;

24 private String email;

25 private String phoneNumber;

26 //----------------

27 // Constructor(s).

28 //----------------

29

30 public Admin ()

31 {

32

33 }

34 public Admin(long userId, String name, Gender gender, String username, String password, String email, String phoneNumber) {

35 this.userId = userId;

36 this.name = name;

37 this.gender = gender;

38 this.username = username;

39 this.password = password;

40 this.email = email;

41 this.phoneNumber = phoneNumber;

42 }

43

44 //----------------------------------

45 // Accessor methods.

46 //----------------------------------

47 public long getUserId() {

48 return userId;

49 }

50

51 public String getName() {

52 return name;

53 }

54

55 public Gender getGender() {

56 return gender;

57 }

58

59 public String getUsername() {

60 return username;

61 }

62

63 public String getPassword() {

64 return password;

65 }

66

67 public String getEmail() {

68 return email;

69 }

70

71 public String getPhoneNumber() {

72 return phoneNumber;

73 }

74 //----------------------------------

75 // Mutator methods.

76 //----------------------------------

77 @XmlAttribute

78 public void setUserId(long userId) {

79 this.userId = userId;

80 }

81

82 @XmlElement

83 public void setName(String name) {

84 this.name = name;

85 }

86

87 @XmlElement

88 public void setGender(Gender gender) {

89 this.gender = gender;

90 }

91

92 @XmlElement

93 public void setUsername(String username) {

94 this.username = username;

95 }

96

97 @XmlElement

98 public void setPassword(String password) {

99 this.password = password;

100 }

101

102 @XmlElement

103 public void setEmail(String email) {

104 this.email = email;

105 }

106

107 @XmlElement

108 public void setPhoneNumber(String phoneNumber) {

109 this.phoneNumber = phoneNumber;

110 }

111

112 //------------------------------------

113 // Miscellaneous other methods.

114 //------------------------------------

115 public void display()

116 {

117 System.out.println("User ID : " + this.getUserId());

118 System.out.println("Name : " + this.getName());

119 System.out.println("Gender : " + this.getGender());

120 System.out.println("Username : " + this.getUsername());

121 System.out.println("Email : " + this.getEmail());

122 System.out.println("Phone Number : " + this.getPhoneNumber());

123 System.out.println("");

124 System.out.println("");

125 }

126 }

127

Code in Binding.java (inside picadillybookstore.domain package):

1 package picadillybookstore.domain;

2

3 /\*\*

4 \* **Binding.java**

5 \* **This** **file** **contains** **Binding** **Enumeration**

6 \* **@author** Albert Darmawan

7 \* **@version** 1.0

8 \* **@since** 2015-05-28

9 \*/

10 public enum **Binding** {

11 *HARDCOVER*,*PAPERBACK*;

12 }

13

Code in Book.java (inside picadillybookstore.domain package):

1 package picadillybookstore.domain;

2

3 /\*\*

4 \* **Book.java**

5 \* **This** **class** **encapsulates** **all** **attribute** **that** **a** **book** **has**

6 \* **@author** Elbert Christopher Larosa

7 \* **@version** 1.0

8 \* **@since** 2015-5-28

9 \*/

10

11 import java.util.ArrayList;

12 import java.util.List;

13 import javax.xml.bind.annotation.XmlAttribute;

14 import javax.xml.bind.annotation.XmlElement;

15 import javax.xml.bind.annotation.XmlRootElement;

16

17

18 @XmlRootElement(name="book")

19 public class **Book** {

20 //------------

21 // Attributes.

22 //------------

23 String author;

24 String title;

25 String isbn;

26 long price;

27 long quantity;

28 String publisherName;

29 int yearPublished;

30 String description;

31 Binding binding;

32 Signed signed;

33 List keywords = new ArrayList();

34 List category = new ArrayList();

35

36 //----------------

37 // Constructor(s).

38 //----------------

39 public **Book**(String author, String title, String isbn, long price, long quantity, String publisherName, int yearPublished, String description, Binding binding, Signed signed, List keywords, List category) {

40 this.author = author;

41 this.title = title;

42 this.isbn = isbn;

43 this.price = price;

44 this.quantity = quantity;

45 this.publisherName = publisherName;

46 this.yearPublished = yearPublished;

47 this.description = description;

48 this.binding = binding;

49 this.signed = signed;

50 this.keywords = keywords;

51 this.category = category;

52 }

53

54 public **Book**()

55 {

56 }

57

58 //----------------------------------

59 // Accessor methods.

60 //----------------------------------

61 public String **getAuthor**() {

62 return author;

63 }

64

65 public String **getTitle**() {

66 return title;

67 }

68

69 public String **getIsbn**() {

70 return isbn;

71 }

72

73 public long **getPrice**() {

74 return price;

75 }

76

77 public long **getQuantity**() {

78 return quantity;

79 }

80

81 public String **getPublisherName**() {

82 return publisherName;

83 }

84

85 public int **getYearPublished**() {

86 return yearPublished;

87 }

88

89 public String **getDescription**() {

90 return description;

91 }

92

93 public Binding **getBinding**() {

94 return binding;

95 }

96

97 public Signed **getSigned**() {

98 return signed;

99 }

100

101 public List **getKeywords**() {

102 return keywords;

103 }

104

105 public List **getCategory**() {

106 return category;

107 }

108

109 //----------------------------------

110 // Mutator methods.

111 //----------------------------------

112 @XmlElement

113 public void **setAuthor**(String author) {

114 this.author = author;

115 }

116

117 @XmlElement

118 public void **setTitle**(String title) {

119 this.title = title;

120 }

121

122 @XmlElement

123 public void **setPrice**(long price) {

124 this.price = price;

125 }

126

127 @XmlElement

128 public void **setQuantity**(long quantity) {

129 this.quantity = quantity;

130 }

131

132 @XmlElement

133 public void **setPublisherName**(String publisherName) {

134 this.publisherName = publisherName;

135 }

136

137 @XmlElement

138 public void **setYearPublished**(int yearPublished) {

139 this.yearPublished = yearPublished;

140 }

141

142 @XmlElement

143 public void **setDescription**(String description) {

144 this.description = description;

145 }

146

147 @XmlElement

148 public void **setBinding**(Binding binding) {

149 this.binding = binding;

150 }

151

152 @XmlElement

153 public void **setSigned**(Signed signed) {

154 this.signed = signed;

155 }

156

157 @XmlElement

158 public void **setKeywords**(List keywords) {

159 this.keywords = keywords;

160 }

161

162 @XmlElement

163 public void **setCategory**(List category) {

164 this.category = category;

165 }

166

167 @XmlAttribute

168 public void **setIsbn**(String isbn) {

169 this.isbn = isbn;

170 }

171

172 //------------------------------------

173 // Miscellaneous other methods.

174 //------------------------------------

175

176 public void **display**()

177 {

178 String element;

179 int keywordsCounter;

180 int categoryCounter;

181 System.*out*.println("Title : " + this.getTitle());

182 System.*out*.println("Author : " + this.getAuthor());

183 System.*out*.println("ISBN : " + this.getIsbn());

184 System.*out*.println("Price : Rp. " + this.getPrice());

185 System.*out*.println("Quantity : " + this.getQuantity());

186 System.*out*.println("Publisher Name : " + this.getPublisherName());

187 System.*out*.println("Year Published : " + this.getYearPublished());

188 System.*out*.println("Description : " + this.getDescription());

189 System.*out*.println("Binding : " + this.getBinding());

190 System.*out*.println("Signed : " + this.getSigned());

191 System.*out*.print("Keywords : ");

192 keywordsCounter =0;

193 for(Object object : keywords) {

194 keywordsCounter+=1;

195 element = (String) object;

196 System.*out*.print(element);

197 if (keywordsCounter!=this.keywords.size())

198 {

199 System.*out*.print(" , ");

200 }

201 }

202 System.*out*.println("");

203

204 System.*out*.print("Category : ");

205 categoryCounter=0;

206 for(Object object : category) {

207 categoryCounter+=1;

208 element = (String) object;

209 System.*out*.print(element);

210 if (categoryCounter!=this.category.size())

211 {

212 System.*out*.print(" , ");

213 }

214 }

215 System.*out*.println("");

216 System.*out*.println("");

217 }

218 }

219

Code in Customer.java (inside picadillybookstore.domain package):

1 package picadillybookstore.domain;

2

3 /\*\*

4 \* **Customer.java**

5 \* **This** **class** **encapsulates** **all** **attributes** **and** **methods** **that** **a** **customer**

6 \* **@author** Albert Darmawan

7 \* **@version** 1.0

8 \* **@since** 2015-05-26

9 \*/

10 import java.util.ArrayList;

11 import java.util.Calendar;

12 import javax.xml.bind.annotation.XmlRootElement;

13 import javax.xml.bind.annotation.XmlAttribute;

14 import javax.xml.bind.annotation.XmlElement;

15

16 @XmlRootElement(name="customer")

17 public class **Customer** {

18 //------------

19 // Constant(s).

20 //------------

21 public final int NUMBER\_OF\_CREDIT\_CARD\_SUBSTRING = 4;

22

23 //------------

24 // Attributes.

25 //------------

26 private long userId;

27 private String name;

28 private Gender gender;

29 private String username;

30 private String password;

31 private String email;

32 private String phoneNumber;

33 private int numberOfPurchase;

34 private String streetAddress;

35 private String city;

36 private String zipCode;

37 private String country;

38 private String creditCardName;

39 private String creditCardNo;

40 private String cvc;

41 private String creditCardMonth;

42 private String creditCardYear;

43 ArrayList<Book> purchasedBook = new ArrayList<>();

44 //----------------

45 // Constructor(s).

46 //----------------

47 public **Customer**() {

48

49 }

50 /\*\*

51 \*

52 \* **@param** userId

53 \* **@param** name

54 \* **@param** gender

55 \* **@param** username

56 \* **@param** password

57 \* **@param** email

58 \* **@param** phoneNumber

59 \* **@param** numberOfPurchase

60 \* **@param** streetAddress

61 \* **@param** city

62 \* **@param** zipCode

63 \* **@param** country

64 \* **@param** creditCardName

65 \* **@param** creditCardNo

66 \* **@param** cvc

67 \* **@param** creditCardMonth

68 \* **@param** creditCardYear

69 \* **@param** getZipCode

70 \*/

71 public **Customer**(long userId, String name, Gender gender, String username, String password, String email, String phoneNumber, int numberOfPurchase, String streetAddress, String city, String zipCode, String country, String creditCardName, String creditCardNo, String cvc, String creditCardMonth, String creditCardYear) {

72 this.userId = userId;

73 this.name = name;

74 this.gender = gender;

75 this.username = username;

76 this.password = password;

77 this.email = email;

78 this.phoneNumber = phoneNumber;

79 this.numberOfPurchase = numberOfPurchase;

80 this.streetAddress = streetAddress;

81 this.city = city;

82 this.zipCode = zipCode;

83 this.country = country;

84 this.creditCardName = creditCardName;

85 this.creditCardNo = creditCardNo;

86 this.cvc = cvc;

87 this.creditCardMonth = creditCardMonth;

88 this.creditCardYear = creditCardYear;

89 }

90

91 //----------------------------------

92 // Accessor methods.

93 //----------------------------------

94 public long **getUserId**() {

95 return userId;

96 }

97

98 public String **getName**() {

99 return name;

100 }

101

102 public Gender **getGender**() {

103 return gender;

104 }

105

106 public String **getUsername**() {

107 return username;

108 }

109

110 public String **getPassword**() {

111 return password;

112 }

113

114 public String **getEmail**() {

115 return email;

116 }

117

118 public String **getPhoneNumber**() {

119 return phoneNumber;

120 }

121

122 public int **getNumberOfPurchase**() {

123 return numberOfPurchase;

124 }

125

126 public String **getStreetAddress**() {

127 return streetAddress;

128 }

129

130 public String **getCity**() {

131 return city;

132 }

133

134 public String **getZipCode**() {

135 return zipCode;

136 }

137

138 public String **getCountry**() {

139 return country;

140 }

141

142 public String **getCreditCardName**() {

143 return creditCardName;

144 }

145

146 public String **getCreditCardNo**() {

147 return creditCardNo;

148 }

149

150 public String **getCvc**() {

151 return cvc;

152 }

153

154 public String **getCreditCardMonth**() {

155 return creditCardMonth;

156 }

157

158 public String **getCreditCardYear**() {

159 return creditCardYear;

160 }

161

162 public ArrayList<Book> **getPurchasedBook**() {

163 return purchasedBook;

164 }

165

166 //----------------------------------

167 // Mutator methods.

168 //----------------------------------

169 @XmlAttribute

170 public void **setUserId**(long userId) {

171 this.userId = userId;

172 }

173

174 @XmlElement

175 public void **setName**(String name) {

176 this.name = name;

177 }

178

179 @XmlElement

180 public void **setGender**(Gender gender) {

181 this.gender = gender;

182 }

183

184 @XmlElement

185 public void **setUsername**(String username) {

186 this.username = username;

187 }

188

189 @XmlElement

190 public void **setPassword**(String password) {

191 this.password = password;

192 }

193

194 @XmlElement

195 public void **setEmail**(String email) {

196 this.email = email;

197 }

198

199 @XmlElement

200 public void **setPhoneNumber**(String phoneNumber) {

201 this.phoneNumber = phoneNumber;

202 }

203

204 @XmlElement

205 public void **setNumberOfPurchase**(int numberOfPurchase) {

206 this.numberOfPurchase = numberOfPurchase;

207 }

208

209 @XmlElement

210 public void **setStreetAddress**(String streetAddress) {

211 this.streetAddress = streetAddress;

212 }

213

214 @XmlElement

215 public void **setCity**(String city) {

216 this.city = city;

217 }

218

219 @XmlElement

220 public void **setZipCode**(String zipCode) {

221 this.zipCode = zipCode;

222 }

223

224 @XmlElement

225 public void **setCountry**(String country) {

226 this.country = country;

227 }

228

229 @XmlElement

230 public void **setCreditCardName**(String creditCardName) {

231 this.creditCardName = creditCardName;

232 }

233

234 @XmlElement

235 public void **setCreditCardNo**(String creditCardNo) {

236 this.creditCardNo = creditCardNo;

237 }

238

239 @XmlElement

240 public void **setCvc**(String cvc) {

241 this.cvc = cvc;

242 }

243

244 @XmlElement

245 public void **setCreditCardMonth**(String creditCardMonth) {

246 this.creditCardMonth = creditCardMonth;

247 }

248

249 @XmlElement

250 public void **setCreditCardYear**(String creditCardYear) {

251 this.creditCardYear = creditCardYear;

252 }

253

254 @XmlElement

255 public void **setPurchasedBook**(ArrayList<Book> purchasedBook) {

256 this.purchasedBook = purchasedBook;

257 }

258

259 //-----------------------------

260 // Business logic methods.

261 //-----------------------------

262 public boolean **isCreditCardNumberValid**()

263 {

264 // Using Luhn Algorithm

265 int sum = 0;

266 boolean alternate = false;

267 for (int i = this.getCreditCardNo().length() - 1; i >= 0; i--)

268 {

269 int n = Integer.*parseInt*(this.getCreditCardNo().substring(i, i + 1));

270 if (alternate)

271 {

272 n \*= 2;

273 if (n > 9)

274 {

275 n = (n % 10) + 1;

276 }

277 }

278 sum += n;

279 alternate = !alternate;

280 }

281 return (sum % 10 == 0);

282 }

283

284 public boolean **isCreditCardNotExpired**()

285 {

286 Calendar currentTime = Calendar.*getInstance*();

287 int currentYear = currentTime.get(Calendar.*YEAR*);

288 int currentMonth = currentTime.get(Calendar.*MONTH*)+1;

289 String creditCardYearFourDigit = "20"+this.getCreditCardYear();

290 int creditCardYearInteger = Integer.*parseInt*(creditCardYearFourDigit);

291 int creditCardMonthInteger = Integer.*parseInt*(this.getCreditCardMonth());

292 if (currentMonth<=creditCardMonthInteger)

293 {

294 if (currentYear<=creditCardYearInteger) {

295 return true;

296 }

297 else {

298 return false;

299 }

300 }

301 else {

302 if (currentYear<=creditCardYearInteger) {

303 return true;

304 }

305 else {

306 return false;

307 }

308 }

309 }

310

311 public PurchaseStatus **purchaseBook**(Book bookBeingPurchased)

312 {

313

314 if (bookBeingPurchased.getQuantity()!=0)

315 {

316 if (isCreditCardNotExpired() && isCreditCardNumberValid())

317 {

318 bookBeingPurchased.quantity-=1;

319 this.setNumberOfPurchase(this.getNumberOfPurchase()+1);

320 purchasedBook.add(bookBeingPurchased);

321 return PurchaseStatus.*success*;

322 }

323 else

324 {

325 return PurchaseStatus.*creditCardInvalid*;

326 }

327 }

328 else {

329 return PurchaseStatus.*noStock*;

330 }

331 }

332

333 //------------------------------------

334 // Miscellaneous other methods.

335 //------------------------------------

336

337 public void **display**()

338 {

339 System.*out*.println("Personal Information");

340 System.*out*.println("---------------------------------------------");

341 System.*out*.println("Name : " + this.getName());

342 System.*out*.println("User ID : " + this.getUserId());

343 System.*out*.println("Gender : " + this.getGender());

344 System.*out*.println("Username : " + this.getUsername());

345 System.*out*.println("Email : " + this.getEmail());

346 System.*out*.println("Phone Number : " + this.getPhoneNumber());

347 System.*out*.println("Number of Purchase : " + this.getNumberOfPurchase());

348 System.*out*.println("Address : " + this.getStreetAddress() + "**\n\t** " + this.getCity() + " , "+ this.getZipCode() + "**\n\t** " + this.getCountry());

349 System.*out*.println("");

350 System.*out*.println("Credit Card Information");

351 System.*out*.println("---------------------------------------------");

352 System.*out*.println("Name : " + this.getCreditCardName());

353 System.*out*.print("No : ");

354 for (int loopCounter=0;loopCounter<NUMBER\_OF\_CREDIT\_CARD\_SUBSTRING;loopCounter++)

355 {

356 System.*out*.print(this.getCreditCardNo().substring(loopCounter\*NUMBER\_OF\_CREDIT\_CARD\_SUBSTRING,(loopCounter\*NUMBER\_OF\_CREDIT\_CARD\_SUBSTRING)+NUMBER\_OF\_CREDIT\_CARD\_SUBSTRING));

357 if (loopCounter!=NUMBER\_OF\_CREDIT\_CARD\_SUBSTRING-1)

358 {

359 System.*out*.print("-");

360 }

361 }

362 System.*out*.println("");

363 System.*out*.println("Expired Date : " + this.getCreditCardMonth() + "/" + this.getCreditCardYear());

364 System.*out*.println("");

365 System.*out*.println("");

366 }

367 }

368

Code in Gender.java (inside picadillybookstore.domain package):

1 package picadillybookstore.domain;

2

3 /\*\*

4 \* **Gender.java**

5 \* **This** **file** **contain** **Gender** **Enumeration**

6 \* **@author** Albert Darmawan

7 \* **@version** 1.0

8 \* **@since** 2015-05-26

9 \*/

10 public enum **Gender** {

11 *MALE*,*FEMALE*;

12 }

13

Code in PurchaseStatus.java (inside picadillybookstore.domain package):

1 package picadillybookstore.domain;

2

3 /\*\*

4 \* **PurchaseStatus.java**

5 \* **This** **file** **contains** **PurchaseStatus** **Enumeration**

6 \* **@author** Albert Darmawan

7 \* **@version** 1.0

8 \* **@since** 2015-05-28

9 \*/

10

11 public enum **PurchaseStatus** {

12 // Enumerate the values that the enum can assume.

13 *success*("Purchasing successful!"),

14 *noStock*("Purchasing failed! This book is not available anymore"),

15 *creditCardInvalid*("Purchasing failed! Your credit card is invalid!");

16

17 // This represents the value of an enum instance.

18 private final String value;

19

20 // A "constructor" of sorts (used above).

21 **PurchaseStatus**(String value) {

22 this.value = value;

23 }

24

25 // Accessor for the value of an enum instance.

26 public String **value**() {

27 return value;

28 }

29 }

30

Code in Signed.java (inside picadillybookstore.domain package):

1 package picadillybookstore.domain;

2

3 /\*\*

4 \* **Signed.java**

5 \* **This** **file** **contains** **Signed** **Enumeration**

6 \* **@author** Albert Darmawan

7 \* **@version** 1.0

8 \* **@since** 2015-05-28

9 \*/

10 public enum **Signed** {

11 *SIGNED*,*NOTSIGNED*;

12 }

13

Code in Transaction.java (inside picadillybookstore.domain package):

1 package picadillybookstore.domain;

2

3 import java.util.Date;

4 import javax.xml.bind.annotation.XmlAccessType;

5 import javax.xml.bind.annotation.XmlAccessorType;

6 import javax.xml.bind.annotation.XmlRootElement;

7 /\*\*

8 \* **Transaction.java**

9 \* **This** **class** **encapsulates** **all** **attributes** **that** **a** **transaction** **has**

10 \* **@author** Matthew Kharli

11 \* **@version** 1.0

12 \* **@since** 2015-5-28

13 \*/

14

15 @XmlRootElement(name="transaction")

16 @XmlAccessorType(XmlAccessType.*FIELD*)

17 public class **Transaction** {

18

19 //------------

20 // Attributes.

21 //------------

22 private long transactionId;

23 private long customerId;

24 private String isbn;

25 private long quantity;

26 private Date date;

27 public boolean paymentVerification;

28

29 //----------------

30 // Constructor(s).

31 //----------------

32 public **Transaction**()

33 {

34

35 }

36

37 public **Transaction**(long transactionId, long customerId, String isbn, long quantity, Date date, boolean paymentVerification) {

38 this.transactionId = transactionId;

39 this.customerId = customerId;

40 this.isbn = isbn;

41 this.quantity = quantity;

42 this.date = date;

43 this.paymentVerification = paymentVerification;

44 }

45

46 //----------------------------------

47 // Accessor methods.

48 //----------------------------------

49 public long **getTransactionId**(){

50 return transactionId;

51 }

52

53 public long **getCustomerId**(){

54 return customerId;

55 }

56

57 public String **getIsbn**(){

58 return isbn;

59 }

60

61 public long **getQuantity**(){

62 return quantity;

63 }

64

65 public Date **getDate**() {

66 return (Date) this.date.clone();

67 }

68

69 public boolean **isPaymentVerification**() {

70 return paymentVerification;

71 }

72

73 //----------------------------------

74 // Mutator methods.

75 //----------------------------------

76 public void **setCustomerId**(long customerId) {

77 this.customerId = customerId;

78 }

79

80 public void **setIsbn**(String isbn){

81 this.isbn = isbn;

82 }

83

84 public void **setQuantity**(long quantity) {

85 this.quantity = quantity;

86 }

87

88 public void **setDate**(Date date) {

89 this.date = (Date) date.clone();

90 }

91

92 public void **setPaymentVerification**(boolean paymentVerification) {

93 this.paymentVerification = paymentVerification;

94 }

95

96 public void **setTransactionId**(long transactionId) {

97 this.transactionId = transactionId;

98 }

99

100 //------------------------------------

101 // Miscellaneous other methods.

102 //------------------------------------

103 public void **display**()

104 {

105 System.*out*.println("Transaction ID : " + this.getTransactionId());

106 System.*out*.println("Customer ID : " + this.getCustomerId());

107 System.*out*.println("ISBN : " + this.getIsbn());

108 System.*out*.println("Quantity : " + this.getQuantity());

109 System.*out*.println("Date : " + this.getDate());

110 System.*out*.print("Payment Verification : ");

111 if(this.isPaymentVerification()==true)

112 {

113 System.*out*.println("Verified");

114 }

115 else

116 {

117 System.*out*.println("Not verified");

118 }

119 System.*out*.println("");

120 }

121 }

122

123

Code in PicadillyBookstoreGUI.java (inside picadillybookstore.gui package):

1 /\*

2 \* To change this license header, choose License Headers in Project Properties.

3 \* To change this template file, choose Tools | Templates

4 \* and open the template in the editor.

5 \*/

6

7 package picadillybookstore.gui;

8 import java.util.Arrays;

9 import java.util.Date;

10 import javax.swing.table.DefaultTableModel;

11

12 import picadillybookstore.domain.Gender;

13 import picadillybookstore.domain.Signed;

14 import picadillybookstore.domain.Binding;

15 import picadillybookstore.dao.DAO;

16 import picadillybookstore.dao.Admins;

17 import picadillybookstore.dao.Customers;

18 import picadillybookstore.dao.Books;

19 import picadillybookstore.dao.Transactions;

20 import picadillybookstore.domain.Admin;

21 import picadillybookstore.domain.Customer;

22 import picadillybookstore.domain.Book;

23 import picadillybookstore.domain.Transaction;

24 import picadillybookstore.domain.PurchaseStatus;

25

26 /\*\*

27 \* **PicadillyBookstoreGUI.java**

28 \* **This** **class** **contains** **GUI** **version** **of** **our** **program**

29 \*

30 \* **@author** Albert Darmawan

31 \* **@version** 1.0

32 \* **@since** 2015-06-14

33 \*/

34 public class **PicadillyBookstoreGUI** extends javax.swing.JFrame {

35 Customer currentCustomer;

36 DefaultTableModel adminBookTableModel;

37 DefaultTableModel adminTransactionTableModel;

38 DefaultTableModel adminCustomerTableModel;

39 DefaultTableModel customerBookTableModel;

40 /\*\*

41 \* Creates new form PicadillyBookstoreGUI

42 \*/

43

44 public **PicadillyBookstoreGUI**() {

45 initComponents();

46 // Display UserChoicePanel Only

47 adminBookTableModel = (DefaultTableModel) adminBookTable.getModel();

48 adminTransactionTableModel = (DefaultTableModel) adminTransactionTable.getModel();

49 adminCustomerTableModel = (DefaultTableModel) adminCustomerTable.getModel();

50 customerBookTableModel = (DefaultTableModel) customerBookTable.getModel();

51 this.hideAllPanel();

52 UserChoicePanel.setVisible(true);

53 }

54

55 private void **displayAdminBookList**() {

56 adminBookTableModel.setRowCount(0);

57 Books bookList = DAO.*getAllBooks*();

58 for (Book bookChecked : bookList.getBooks())

59 {

60 adminBookTableModel.insertRow(adminBookTableModel.getRowCount(), new Object[]{

61 bookChecked.getIsbn(),

62 bookChecked.getTitle(),

63 bookChecked.getAuthor(),

64 bookChecked.getQuantity(),

65 bookChecked.getPrice()

66 });

67 }

68 }

69

70 private void **displayCustomerBookList**() {

71 customerBookTableModel.setRowCount(0);

72 Books bookList = DAO.*getAllBooks*();

73 for (Book bookChecked : bookList.getBooks())

74 {

75 customerBookTableModel.insertRow(customerBookTableModel.getRowCount(), new Object[]{

76 bookChecked.getIsbn(),

77 bookChecked.getTitle(),

78 bookChecked.getAuthor(),

79 bookChecked.getQuantity(),

80 bookChecked.getPrice()

81 });

82 }

83 }

84

85 private void **displayAdminTransactionList**() {

86 adminTransactionTableModel.setRowCount(0);

87 Transactions transactionList = DAO.*getAllTransactions*();

88 String verificationResult;

89 for (Transaction transactionChecked : transactionList.getTransactions())

90 {

91 if (transactionChecked.paymentVerification)

92 {

93 verificationResult = "Verified";

94

95 }

96 else {

97 verificationResult = "Not verified";

98 }

99 adminTransactionTableModel.insertRow(adminTransactionTableModel.getRowCount(), new Object[]{

100 transactionChecked.getTransactionId(),

101 transactionChecked.getCustomerId(),

102 transactionChecked.getIsbn(),

103 transactionChecked.getQuantity(),

104 transactionChecked.getDate(),

105 verificationResult

106 });

107 }

108 }

109

110 private void **displayAdminCustomerList**() {

111 adminCustomerTableModel.setRowCount(0);

112 Customers customerList = DAO.*getAllCustomers*();

113 for (Customer customerChecked : customerList.getCustomers())

114 {

115 adminCustomerTableModel.insertRow(adminCustomerTableModel.getRowCount(), new Object[]{

116 customerChecked.getUserId(),

117 customerChecked.getName(),

118 customerChecked.getEmail(),

119 customerChecked.getNumberOfPurchase(),

120 customerChecked.getCountry()

121 });

122 }

123 }

124

125 /\*\*

126 \* **This** **method** **is** **called** **from** **within** **the** **constructor** **to** **initialize** **the** **form.**

127 \* WARNING: Do NOT modify this code. The content of this method is always

128 \* regenerated by the Form Editor.

129 \*/

130 @SuppressWarnings("unchecked")

131 // <editor-fold defaultstate="collapsed" desc="Generated Code">

132 private void **initComponents**() {

133

134 filler1 = new javax.swing.Box.Filler(new java.awt.Dimension(0, 0), new java.awt.Dimension(0, 0), new java.awt.Dimension(32767, 0));

135 buttonGroup1 = new javax.swing.ButtonGroup();

136 UserChoicePanel = new javax.swing.JPanel();

137 userChoicePanelTitle = new javax.swing.JLabel();

138 adminButton = new javax.swing.JButton();

139 customerButton = new javax.swing.JButton();

140 AdminLoginPanel = new javax.swing.JPanel();

141 adminLoginPanelSubtitle = new javax.swing.JLabel();

142 adminLoginPanelTitle = new javax.swing.JLabel();

143 adminUsernameLabel = new javax.swing.JLabel();

144 adminPasswordLabel = new javax.swing.JLabel();

145 adminUsernameField = new javax.swing.JTextField();

146 adminLoginButton = new javax.swing.JButton();

147 adminPasswordField = new javax.swing.JPasswordField();

148 AdminMenuPanel = new javax.swing.JPanel();

149 adminMenuPanelTitle = new javax.swing.JLabel();

150 adminMenuPanelSubtitle = new javax.swing.JLabel();

151 adminMenuBooksButton = new javax.swing.JButton();

152 adminMenuCustomersButton = new javax.swing.JButton();

153 adminMenuTransactionsButton = new javax.swing.JButton();

154 AdminTransactionsListPanel = new javax.swing.JPanel();

155 AdminTransactionsListPanelTitle = new javax.swing.JLabel();

156 AdminTransactionsListPanelSubtitle = new javax.swing.JLabel();

157 adminTransactionsListLogoutButton = new javax.swing.JButton();

158 adminTransactionsListMenuButton = new javax.swing.JButton();

159 jScrollPane2 = new javax.swing.JScrollPane();

160 adminTransactionTable = new javax.swing.JTable();

161 AdminBooksListPanel = new javax.swing.JPanel();

162 AdminBooksListPanelTitle = new javax.swing.JLabel();

163 AdminBooksListPanelSubtitle = new javax.swing.JLabel();

164 adminBooksListLogoutButton = new javax.swing.JButton();

165 addBookButton = new javax.swing.JButton();

166 adminBooksListMenuButton = new javax.swing.JButton();

167 jScrollPane1 = new javax.swing.JScrollPane();

168 adminBookTable = new javax.swing.JTable();

169 isbnToDeleteField = new javax.swing.JTextField();

170 AdminBooksListISBNLabel = new javax.swing.JLabel();

171 AdminDeleteBookButton = new javax.swing.JButton();

172 AdminCustomersListPanel = new javax.swing.JPanel();

173 AdminCustomersListPanelTitle = new javax.swing.JLabel();

174 AdminCustomersListPanelSubtitle = new javax.swing.JLabel();

175 adminCustomersListLogoutButton = new javax.swing.JButton();

176 adminCustomersListMenuButton = new javax.swing.JButton();

177 jScrollPane3 = new javax.swing.JScrollPane();

178 adminCustomerTable = new javax.swing.JTable();

179 CustomerLoginPanel = new javax.swing.JPanel();

180 customerLoginPanelTitle = new javax.swing.JLabel();

181 customerLoginPanelSubtitle = new javax.swing.JLabel();

182 customerSignUpOptionButton = new javax.swing.JButton();

183 customerUsernameLabel = new javax.swing.JLabel();

184 customerPasswordLabel = new javax.swing.JLabel();

185 customerUsernameField = new javax.swing.JTextField();

186 customerLoginButton = new javax.swing.JButton();

187 customerPasswordField = new javax.swing.JPasswordField();

188 CustomerMainMenuPanel = new javax.swing.JPanel();

189 CustomerMainMenuPanelTitle = new javax.swing.JLabel();

190 CustomerMainMenuPanelSubtitle = new javax.swing.JLabel();

191 customerMainMenuLogoutButton = new javax.swing.JButton();

192 jScrollPane4 = new javax.swing.JScrollPane();

193 customerBookTable = new javax.swing.JTable();

194 CustomerISBNChoiceLabel = new javax.swing.JLabel();

195 purchaseButton = new javax.swing.JButton();

196 customerIsbnChoiceField = new javax.swing.JTextField();

197 purchasingStatusLabel = new javax.swing.JLabel();

198 CustomerEditProfilePanel = new javax.swing.JPanel();

199 jLabel51 = new javax.swing.JLabel();

200 jTextField26 = new javax.swing.JTextField();

201 jLabel52 = new javax.swing.JLabel();

202 jLabel53 = new javax.swing.JLabel();

203 jLabel54 = new javax.swing.JLabel();

204 jTextField27 = new javax.swing.JTextField();

205 jButton32 = new javax.swing.JButton();

206 jTextField28 = new javax.swing.JTextField();

207 jLabel55 = new javax.swing.JLabel();

208 jLabel56 = new javax.swing.JLabel();

209 jLabel57 = new javax.swing.JLabel();

210 jLabel58 = new javax.swing.JLabel();

211 jTextField29 = new javax.swing.JTextField();

212 jTextField30 = new javax.swing.JTextField();

213 jTextField31 = new javax.swing.JTextField();

214 jLabel59 = new javax.swing.JLabel();

215 jLabel60 = new javax.swing.JLabel();

216 jLabel61 = new javax.swing.JLabel();

217 jTextField32 = new javax.swing.JTextField();

218 jTextField33 = new javax.swing.JTextField();

219 jLabel62 = new javax.swing.JLabel();

220 jLabel63 = new javax.swing.JLabel();

221 jTextField34 = new javax.swing.JTextField();

222 jTextField35 = new javax.swing.JTextField();

223 jSeparator3 = new javax.swing.JSeparator();

224 jSeparator4 = new javax.swing.JSeparator();

225 jLabel64 = new javax.swing.JLabel();

226 jLabel65 = new javax.swing.JLabel();

227 jLabel66 = new javax.swing.JLabel();

228 jTextField36 = new javax.swing.JTextField();

229 jTextField37 = new javax.swing.JTextField();

230 jLabel67 = new javax.swing.JLabel();

231 jLabel68 = new javax.swing.JLabel();

232 jTextField38 = new javax.swing.JTextField();

233 jLabel69 = new javax.swing.JLabel();

234 jTextField39 = new javax.swing.JTextField();

235 jLabel70 = new javax.swing.JLabel();

236 jTextField40 = new javax.swing.JTextField();

237 jLabel71 = new javax.swing.JLabel();

238 jCheckBox3 = new javax.swing.JCheckBox();

239 jCheckBox4 = new javax.swing.JCheckBox();

240 jLabel93 = new javax.swing.JLabel();

241 jTextField43 = new javax.swing.JTextField();

242 AdminAddBookPanel = new javax.swing.JPanel();

243 AdminAddBookPanelTitle = new javax.swing.JLabel();

244 AdminSubmitBook = new javax.swing.JButton();

245 AdminAddBookPanelSubtitle = new javax.swing.JLabel();

246 bookQuantityLabel = new javax.swing.JLabel();

247 quantityField = new javax.swing.JTextField();

248 isbnLabel = new javax.swing.JLabel();

249 isbnField = new javax.swing.JTextField();

250 authorField = new javax.swing.JTextField();

251 authorLabel = new javax.swing.JLabel();

252 publisherNameField = new javax.swing.JTextField();

253 publishedYearField = new javax.swing.JTextField();

254 publishedYearLabel = new javax.swing.JLabel();

255 bookTitleLabel = new javax.swing.JLabel();

256 bookTitleField = new javax.swing.JTextField();

257 publishedNameLabel = new javax.swing.JLabel();

258 keywordsField = new javax.swing.JTextField();

259 descriptionField = new javax.swing.JTextField();

260 descriptionLabel = new javax.swing.JLabel();

261 bookPriceLabel = new javax.swing.JLabel();

262 bookPriceField = new javax.swing.JTextField();

263 keywordsLabel = new javax.swing.JLabel();

264 descriptionCharactersLimitLabel = new javax.swing.JLabel();

265 bookPriceCurrencyLabel = new javax.swing.JLabel();

266 bindingLabel = new javax.swing.JLabel();

267 signatureLabel = new javax.swing.JLabel();

268 signatureComboBox = new javax.swing.JComboBox();

269 bindingComboBox = new javax.swing.JComboBox();

270 categoryLabel = new javax.swing.JLabel();

271 categoryField = new javax.swing.JTextField();

272 CustomerSignUpScrollPane = new javax.swing.JScrollPane();

273 CustomerSignUpPanel = new javax.swing.JPanel();

274 customerSignUpPanelTitle = new javax.swing.JLabel();

275 customerLastNameField = new javax.swing.JTextField();

276 PersonalProfileLabel = new javax.swing.JLabel();

277 customerFirstNameLabel = new javax.swing.JLabel();

278 customerLastNameLabel = new javax.swing.JLabel();

279 customerFirstNameField = new javax.swing.JTextField();

280 customerSignUpSubmitButton = new javax.swing.JButton();

281 customerStreetField = new javax.swing.JTextField();

282 customerGenderLabel = new javax.swing.JLabel();

283 customerStreetLabel = new javax.swing.JLabel();

284 customerCityLabel = new javax.swing.JLabel();

285 customerZipCodeLabel = new javax.swing.JLabel();

286 customerCityField = new javax.swing.JTextField();

287 customerZipCodeField = new javax.swing.JTextField();

288 customerCountryField = new javax.swing.JTextField();

289 customerCountryLabel = new javax.swing.JLabel();

290 customerPhoneNumberLabel = new javax.swing.JLabel();

291 customerEmailLabel = new javax.swing.JLabel();

292 customerPhoneNumberField = new javax.swing.JTextField();

293 customerEmailField = new javax.swing.JTextField();

294 customerAccountUsernameLabel = new javax.swing.JLabel();

295 customerAccountPasswordLabel = new javax.swing.JLabel();

296 customerAccountUsernameField = new javax.swing.JTextField();

297 customerSignUpPanelSeparator1 = new javax.swing.JSeparator();

298 customerSignUpPanelSeparator2 = new javax.swing.JSeparator();

299 AccountInfoLabel = new javax.swing.JLabel();

300 customerCreditCardNameLabel = new javax.swing.JLabel();

301 customerCreditCardNumberLabel = new javax.swing.JLabel();

302 customerCreditCardNameField = new javax.swing.JTextField();

303 customerCvcField = new javax.swing.JTextField();

304 customerCreditCardMonthLabel = new javax.swing.JLabel();

305 customerCvcLabel = new javax.swing.JLabel();

306 customerCreditCardNumberField = new javax.swing.JTextField();

307 creditCardProfileLabel = new javax.swing.JLabel();

308 customerCreditCardMonthField = new javax.swing.JTextField();

309 customerCreditCardYearLabel = new javax.swing.JLabel();

310 customerCreditCardYearField = new javax.swing.JTextField();

311 customerAccountPasswordField = new javax.swing.JPasswordField();

312 customerGenderComboBoxes = new javax.swing.JComboBox();

313

314 setDefaultCloseOperation(javax.swing.WindowConstants.*EXIT\_ON\_CLOSE*);

315

316 UserChoicePanel.setPreferredSize(new java.awt.Dimension(650, 449));

317

318 userChoicePanelTitle.setFont(new java.awt.Font("Tahoma", 0, 36)); // NOI18N

319 userChoicePanelTitle.setText("Picadilly Bookstore");

320

321 adminButton.setText("Admin");

322 adminButton.addActionListener(new java.awt.event.ActionListener() {

323 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

324 adminButtonActionPerformed(evt);

325 }

326 });

327

328 customerButton.setText("Customer");

329 customerButton.addActionListener(new java.awt.event.ActionListener() {

330 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

331 customerButtonActionPerformed(evt);

332 }

333 });

334

335 javax.swing.GroupLayout UserChoicePanelLayout = new javax.swing.GroupLayout(UserChoicePanel);

336 UserChoicePanel.setLayout(UserChoicePanelLayout);

337 UserChoicePanelLayout.setHorizontalGroup(

338 UserChoicePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

339 .addGroup(UserChoicePanelLayout.createSequentialGroup()

340 .addGap(175, 175, 175)

341 .addGroup(UserChoicePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

342 .addComponent(userChoicePanelTitle)

343 .addGroup(UserChoicePanelLayout.createSequentialGroup()

344 .addGap(94, 94, 94)

345 .addGroup(UserChoicePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

346 .addComponent(customerButton)

347 .addComponent(adminButton, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 105, javax.swing.GroupLayout.*PREFERRED\_SIZE*))))

348 .addContainerGap(361, Short.*MAX\_VALUE*))

349 );

350

351 UserChoicePanelLayout.linkSize(javax.swing.SwingConstants.*HORIZONTAL*, new java.awt.Component[] {adminButton, customerButton});

352

353 UserChoicePanelLayout.setVerticalGroup(

354 UserChoicePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

355 .addGroup(UserChoicePanelLayout.createSequentialGroup()

356 .addGap(106, 106, 106)

357 .addComponent(userChoicePanelTitle)

358 .addGap(45, 45, 45)

359 .addComponent(adminButton, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 46, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

360 .addGap(18, 18, 18)

361 .addComponent(customerButton)

362 .addContainerGap(1150, Short.*MAX\_VALUE*))

363 );

364

365 UserChoicePanelLayout.linkSize(javax.swing.SwingConstants.*VERTICAL*, new java.awt.Component[] {adminButton, customerButton});

366

367 AdminLoginPanel.setPreferredSize(new java.awt.Dimension(650, 449));

368

369 adminLoginPanelSubtitle.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

370 adminLoginPanelSubtitle.setText("Admin Login");

371

372 adminLoginPanelTitle.setFont(new java.awt.Font("Tahoma", 0, 36)); // NOI18N

373 adminLoginPanelTitle.setText("Picadilly Bookstore");

374

375 adminUsernameLabel.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

376 adminUsernameLabel.setText("Username:");

377

378 adminPasswordLabel.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

379 adminPasswordLabel.setText("Password:");

380

381 adminUsernameField.addActionListener(new java.awt.event.ActionListener() {

382 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

383 adminUsernameFieldActionPerformed(evt);

384 }

385 });

386

387 adminLoginButton.setText("Enter");

388 adminLoginButton.addActionListener(new java.awt.event.ActionListener() {

389 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

390 adminLoginButtonActionPerformed(evt);

391 }

392 });

393

394 javax.swing.GroupLayout AdminLoginPanelLayout = new javax.swing.GroupLayout(AdminLoginPanel);

395 AdminLoginPanel.setLayout(AdminLoginPanelLayout);

396 AdminLoginPanelLayout.setHorizontalGroup(

397 AdminLoginPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

398 .addGroup(AdminLoginPanelLayout.createSequentialGroup()

399 .addContainerGap(179, Short.*MAX\_VALUE*)

400 .addGroup(AdminLoginPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

401 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, AdminLoginPanelLayout.createSequentialGroup()

402 .addGroup(AdminLoginPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

403 .addComponent(adminLoginPanelTitle)

404 .addGroup(AdminLoginPanelLayout.createSequentialGroup()

405 .addGap(40, 40, 40)

406 .addGroup(AdminLoginPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*)

407 .addComponent(adminPasswordLabel)

408 .addComponent(adminUsernameLabel))

409 .addGap(18, 18, 18)

410 .addGroup(AdminLoginPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*, false)

411 .addComponent(adminUsernameField, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 110, Short.*MAX\_VALUE*)

412 .addComponent(adminPasswordField)))

413 .addGroup(AdminLoginPanelLayout.createSequentialGroup()

414 .addGap(114, 114, 114)

415 .addComponent(adminLoginPanelSubtitle)))

416 .addGap(175, 175, 175))

417 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, AdminLoginPanelLayout.createSequentialGroup()

418 .addComponent(adminLoginButton, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 89, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

419 .addGap(272, 272, 272))))

420 );

421 AdminLoginPanelLayout.setVerticalGroup(

422 AdminLoginPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

423 .addGroup(AdminLoginPanelLayout.createSequentialGroup()

424 .addGap(92, 92, 92)

425 .addComponent(adminLoginPanelTitle)

426 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

427 .addComponent(adminLoginPanelSubtitle)

428 .addGap(38, 38, 38)

429 .addGroup(AdminLoginPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

430 .addComponent(adminUsernameLabel)

431 .addComponent(adminUsernameField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

432 .addGap(18, 18, 18)

433 .addGroup(AdminLoginPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*BASELINE*)

434 .addComponent(adminPasswordLabel)

435 .addComponent(adminPasswordField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

436 .addGap(27, 27, 27)

437 .addComponent(adminLoginButton, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 38, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

438 .addContainerGap(107, Short.*MAX\_VALUE*))

439 );

440

441 AdminMenuPanel.setPreferredSize(new java.awt.Dimension(650, 449));

442

443 adminMenuPanelTitle.setFont(new java.awt.Font("Tahoma", 0, 36)); // NOI18N

444 adminMenuPanelTitle.setText("Picadilly Bookstore");

445

446 adminMenuPanelSubtitle.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

447 adminMenuPanelSubtitle.setText("Admin Menu");

448

449 adminMenuBooksButton.setText("Books");

450 adminMenuBooksButton.addActionListener(new java.awt.event.ActionListener() {

451 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

452 adminMenuBooksButtonActionPerformed(evt);

453 }

454 });

455

456 adminMenuCustomersButton.setText("Customers");

457 adminMenuCustomersButton.addActionListener(new java.awt.event.ActionListener() {

458 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

459 adminMenuCustomersButtonActionPerformed(evt);

460 }

461 });

462

463 adminMenuTransactionsButton.setText("Transactions");

464 adminMenuTransactionsButton.addActionListener(new java.awt.event.ActionListener() {

465 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

466 adminMenuTransactionsButtonActionPerformed(evt);

467 }

468 });

469

470 javax.swing.GroupLayout AdminMenuPanelLayout = new javax.swing.GroupLayout(AdminMenuPanel);

471 AdminMenuPanel.setLayout(AdminMenuPanelLayout);

472 AdminMenuPanelLayout.setHorizontalGroup(

473 AdminMenuPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

474 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, AdminMenuPanelLayout.createSequentialGroup()

475 .addContainerGap(179, Short.*MAX\_VALUE*)

476 .addGroup(AdminMenuPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

477 .addComponent(adminMenuPanelTitle)

478 .addGroup(AdminMenuPanelLayout.createSequentialGroup()

479 .addGap(94, 94, 94)

480 .addGroup(AdminMenuPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*)

481 .addComponent(adminMenuCustomersButton)

482 .addComponent(adminMenuBooksButton, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 97, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

483 .addComponent(adminMenuTransactionsButton)))

484 .addGroup(AdminMenuPanelLayout.createSequentialGroup()

485 .addGap(104, 104, 104)

486 .addComponent(adminMenuPanelSubtitle)))

487 .addGap(175, 175, 175))

488 );

489

490 AdminMenuPanelLayout.linkSize(javax.swing.SwingConstants.*HORIZONTAL*, new java.awt.Component[] {adminMenuBooksButton, adminMenuCustomersButton, adminMenuTransactionsButton});

491

492 AdminMenuPanelLayout.setVerticalGroup(

493 AdminMenuPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

494 .addGroup(AdminMenuPanelLayout.createSequentialGroup()

495 .addGap(77, 77, 77)

496 .addComponent(adminMenuPanelTitle)

497 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*RELATED*)

498 .addComponent(adminMenuPanelSubtitle)

499 .addGap(32, 32, 32)

500 .addComponent(adminMenuBooksButton, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 43, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

501 .addGap(18, 18, 18)

502 .addComponent(adminMenuCustomersButton)

503 .addGap(18, 18, 18)

504 .addComponent(adminMenuTransactionsButton)

505 .addContainerGap(108, Short.*MAX\_VALUE*))

506 );

507

508 AdminMenuPanelLayout.linkSize(javax.swing.SwingConstants.*VERTICAL*, new java.awt.Component[] {adminMenuBooksButton, adminMenuCustomersButton, adminMenuTransactionsButton});

509

510 AdminTransactionsListPanel.setPreferredSize(new java.awt.Dimension(650, 449));

511

512 AdminTransactionsListPanelTitle.setFont(new java.awt.Font("Tahoma", 0, 36)); // NOI18N

513 AdminTransactionsListPanelTitle.setText("Picadilly Bookstore");

514

515 AdminTransactionsListPanelSubtitle.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

516 AdminTransactionsListPanelSubtitle.setText("List of Transactions");

517

518 adminTransactionsListLogoutButton.setText("Logout");

519 adminTransactionsListLogoutButton.addActionListener(new java.awt.event.ActionListener() {

520 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

521 adminTransactionsListLogoutButtonActionPerformed(evt);

522 }

523 });

524

525 adminTransactionsListMenuButton.setText("Menu");

526 adminTransactionsListMenuButton.addActionListener(new java.awt.event.ActionListener() {

527 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

528 adminTransactionsListMenuButtonActionPerformed(evt);

529 }

530 });

531

532 adminTransactionTable.setModel(new javax.swing.table.DefaultTableModel(

533 new Object [][] {

534

535 },

536 new String [] {

537 "Transaction ID", "Customer ID", "ISBN", "Quantity", "Date", "Verification"

538 }

539 ));

540 jScrollPane2.setViewportView(adminTransactionTable);

541

542 javax.swing.GroupLayout AdminTransactionsListPanelLayout = new javax.swing.GroupLayout(AdminTransactionsListPanel);

543 AdminTransactionsListPanel.setLayout(AdminTransactionsListPanelLayout);

544 AdminTransactionsListPanelLayout.setHorizontalGroup(

545 AdminTransactionsListPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

546 .addGroup(AdminTransactionsListPanelLayout.createSequentialGroup()

547 .addContainerGap(18, Short.*MAX\_VALUE*)

548 .addGroup(AdminTransactionsListPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

549 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, AdminTransactionsListPanelLayout.createSequentialGroup()

550 .addComponent(AdminTransactionsListPanelSubtitle)

551 .addGap(248, 248, 248))

552 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, AdminTransactionsListPanelLayout.createSequentialGroup()

553 .addComponent(AdminTransactionsListPanelTitle)

554 .addGap(168, 168, 168))

555 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, AdminTransactionsListPanelLayout.createSequentialGroup()

556 .addComponent(adminTransactionsListMenuButton, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 89, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

557 .addGap(18, 18, 18)

558 .addComponent(adminTransactionsListLogoutButton, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 89, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

559 .addContainerGap())

560 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, AdminTransactionsListPanelLayout.createSequentialGroup()

561 .addComponent(jScrollPane2, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 626, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

562 .addContainerGap())))

563 );

564 AdminTransactionsListPanelLayout.setVerticalGroup(

565 AdminTransactionsListPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

566 .addGroup(AdminTransactionsListPanelLayout.createSequentialGroup()

567 .addContainerGap()

568 .addComponent(AdminTransactionsListPanelTitle)

569 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

570 .addComponent(AdminTransactionsListPanelSubtitle)

571 .addGap(28, 28, 28)

572 .addGroup(AdminTransactionsListPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*BASELINE*)

573 .addComponent(adminTransactionsListLogoutButton, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 37, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

574 .addComponent(adminTransactionsListMenuButton, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 37, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

575 .addGap(18, 18, 18)

576 .addComponent(jScrollPane2, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 139, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

577 .addContainerGap(143, Short.*MAX\_VALUE*))

578 );

579

580 AdminBooksListPanel.setPreferredSize(new java.awt.Dimension(650, 449));

581

582 AdminBooksListPanelTitle.setFont(new java.awt.Font("Tahoma", 0, 36)); // NOI18N

583 AdminBooksListPanelTitle.setText("Picadilly Bookstore");

584

585 AdminBooksListPanelSubtitle.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

586 AdminBooksListPanelSubtitle.setText("List of Books");

587

588 adminBooksListLogoutButton.setText("Logout");

589 adminBooksListLogoutButton.addActionListener(new java.awt.event.ActionListener() {

590 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

591 adminBooksListLogoutButtonActionPerformed(evt);

592 }

593 });

594

595 addBookButton.setText("Add New");

596 addBookButton.addActionListener(new java.awt.event.ActionListener() {

597 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

598 addBookButtonActionPerformed(evt);

599 }

600 });

601

602 adminBooksListMenuButton.setText("Menu");

603 adminBooksListMenuButton.addActionListener(new java.awt.event.ActionListener() {

604 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

605 adminBooksListMenuButtonActionPerformed(evt);

606 }

607 });

608

609 adminBookTable.setModel(new javax.swing.table.DefaultTableModel(

610 new Object [][] {

611

612 },

613 new String [] {

614 "ISBN", "Title", "Author", "Quantity", "Price (Rp.)"

615 }

616 ));

617 jScrollPane1.setViewportView(adminBookTable);

618

619 isbnToDeleteField.addActionListener(new java.awt.event.ActionListener() {

620 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

621 isbnToDeleteFieldActionPerformed(evt);

622 }

623 });

624

625 AdminBooksListISBNLabel.setText("ISBN:");

626

627 AdminDeleteBookButton.setText("Delete");

628 AdminDeleteBookButton.addActionListener(new java.awt.event.ActionListener() {

629 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

630 AdminDeleteBookButtonActionPerformed(evt);

631 }

632 });

633

634 javax.swing.GroupLayout AdminBooksListPanelLayout = new javax.swing.GroupLayout(AdminBooksListPanel);

635 AdminBooksListPanel.setLayout(AdminBooksListPanelLayout);

636 AdminBooksListPanelLayout.setHorizontalGroup(

637 AdminBooksListPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

638 .addGroup(AdminBooksListPanelLayout.createSequentialGroup()

639 .addGroup(AdminBooksListPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

640 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, AdminBooksListPanelLayout.createSequentialGroup()

641 .addContainerGap()

642 .addComponent(addBookButton, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 89, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

643 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*RELATED*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

644 .addComponent(adminBooksListMenuButton, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 89, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

645 .addGap(190, 190, 190)

646 .addComponent(adminBooksListLogoutButton, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 89, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

647 .addGroup(AdminBooksListPanelLayout.createSequentialGroup()

648 .addGroup(AdminBooksListPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

649 .addGroup(AdminBooksListPanelLayout.createSequentialGroup()

650 .addGap(182, 182, 182)

651 .addComponent(AdminBooksListPanelTitle))

652 .addGroup(AdminBooksListPanelLayout.createSequentialGroup()

653 .addGap(277, 277, 277)

654 .addComponent(AdminBooksListPanelSubtitle))

655 .addGroup(AdminBooksListPanelLayout.createSequentialGroup()

656 .addGap(18, 18, 18)

657 .addGroup(AdminBooksListPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

658 .addComponent(jScrollPane1, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 603, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

659 .addGroup(AdminBooksListPanelLayout.createSequentialGroup()

660 .addGap(30, 30, 30)

661 .addComponent(AdminBooksListISBNLabel)

662 .addGap(18, 18, 18)

663 .addComponent(isbnToDeleteField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 135, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

664 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

665 .addComponent(AdminDeleteBookButton)))))

666 .addGap(0, 23, Short.*MAX\_VALUE*)))

667 .addContainerGap())

668 );

669 AdminBooksListPanelLayout.setVerticalGroup(

670 AdminBooksListPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

671 .addGroup(AdminBooksListPanelLayout.createSequentialGroup()

672 .addContainerGap()

673 .addComponent(AdminBooksListPanelTitle)

674 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

675 .addComponent(AdminBooksListPanelSubtitle)

676 .addGap(29, 29, 29)

677 .addGroup(AdminBooksListPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*BASELINE*)

678 .addComponent(adminBooksListLogoutButton, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 37, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

679 .addComponent(adminBooksListMenuButton, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 37, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

680 .addComponent(addBookButton, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 37, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

681 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*RELATED*, 31, Short.*MAX\_VALUE*)

682 .addGroup(AdminBooksListPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*BASELINE*)

683 .addComponent(isbnToDeleteField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

684 .addComponent(AdminBooksListISBNLabel)

685 .addComponent(AdminDeleteBookButton))

686 .addGap(36, 36, 36)

687 .addComponent(jScrollPane1, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 140, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

688 .addGap(62, 62, 62))

689 );

690

691 AdminCustomersListPanel.setPreferredSize(new java.awt.Dimension(650, 449));

692

693 AdminCustomersListPanelTitle.setFont(new java.awt.Font("Tahoma", 0, 36)); // NOI18N

694 AdminCustomersListPanelTitle.setText("Picadilly Bookstore");

695

696 AdminCustomersListPanelSubtitle.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

697 AdminCustomersListPanelSubtitle.setText("List of Customers");

698

699 adminCustomersListLogoutButton.setText("Logout");

700 adminCustomersListLogoutButton.addActionListener(new java.awt.event.ActionListener() {

701 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

702 adminCustomersListLogoutButtonActionPerformed(evt);

703 }

704 });

705

706 adminCustomersListMenuButton.setText("Menu");

707 adminCustomersListMenuButton.addActionListener(new java.awt.event.ActionListener() {

708 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

709 adminCustomersListMenuButtonActionPerformed(evt);

710 }

711 });

712

713 adminCustomerTable.setModel(new javax.swing.table.DefaultTableModel(

714 new Object [][] {

715

716 },

717 new String [] {

718 "User ID", "Name", "Email", "Number Of Purchase", "Country"

719 }

720 ));

721 jScrollPane3.setViewportView(adminCustomerTable);

722

723 javax.swing.GroupLayout AdminCustomersListPanelLayout = new javax.swing.GroupLayout(AdminCustomersListPanel);

724 AdminCustomersListPanel.setLayout(AdminCustomersListPanelLayout);

725 AdminCustomersListPanelLayout.setHorizontalGroup(

726 AdminCustomersListPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

727 .addGroup(AdminCustomersListPanelLayout.createSequentialGroup()

728 .addGroup(AdminCustomersListPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

729 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, AdminCustomersListPanelLayout.createSequentialGroup()

730 .addContainerGap(javax.swing.GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

731 .addComponent(adminCustomersListMenuButton, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 89, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

732 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

733 .addComponent(adminCustomersListLogoutButton, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 89, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

734 .addGroup(AdminCustomersListPanelLayout.createSequentialGroup()

735 .addGap(174, 174, 174)

736 .addGroup(AdminCustomersListPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

737 .addComponent(AdminCustomersListPanelTitle)

738 .addGroup(AdminCustomersListPanelLayout.createSequentialGroup()

739 .addGap(78, 78, 78)

740 .addComponent(AdminCustomersListPanelSubtitle)))

741 .addGap(0, 0, Short.*MAX\_VALUE*))

742 .addGroup(AdminCustomersListPanelLayout.createSequentialGroup()

743 .addGap(20, 20, 20)

744 .addComponent(jScrollPane3, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 624, Short.*MAX\_VALUE*)))

745 .addContainerGap())

746 );

747 AdminCustomersListPanelLayout.setVerticalGroup(

748 AdminCustomersListPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

749 .addGroup(AdminCustomersListPanelLayout.createSequentialGroup()

750 .addContainerGap()

751 .addComponent(AdminCustomersListPanelTitle)

752 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

753 .addComponent(AdminCustomersListPanelSubtitle)

754 .addGap(30, 30, 30)

755 .addGroup(AdminCustomersListPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*BASELINE*)

756 .addComponent(adminCustomersListLogoutButton, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 37, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

757 .addComponent(adminCustomersListMenuButton, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 37, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

758 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

759 .addComponent(jScrollPane3, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 164, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

760 .addContainerGap(262, Short.*MAX\_VALUE*))

761 );

762

763 CustomerLoginPanel.setPreferredSize(new java.awt.Dimension(650, 449));

764

765 customerLoginPanelTitle.setFont(new java.awt.Font("Tahoma", 0, 36)); // NOI18N

766 customerLoginPanelTitle.setText("Picadilly Bookstore");

767

768 customerLoginPanelSubtitle.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

769 customerLoginPanelSubtitle.setText("Customer Login");

770

771 customerSignUpOptionButton.setText("Sign Up");

772 customerSignUpOptionButton.addActionListener(new java.awt.event.ActionListener() {

773 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

774 customerSignUpOptionButtonActionPerformed(evt);

775 }

776 });

777

778 customerUsernameLabel.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

779 customerUsernameLabel.setText("Username:");

780

781 customerPasswordLabel.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

782 customerPasswordLabel.setText("Password:");

783

784 customerUsernameField.addActionListener(new java.awt.event.ActionListener() {

785 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

786 customerUsernameFieldActionPerformed(evt);

787 }

788 });

789

790 customerLoginButton.setText("Enter");

791 customerLoginButton.addActionListener(new java.awt.event.ActionListener() {

792 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

793 customerLoginButtonActionPerformed(evt);

794 }

795 });

796

797 javax.swing.GroupLayout CustomerLoginPanelLayout = new javax.swing.GroupLayout(CustomerLoginPanel);

798 CustomerLoginPanel.setLayout(CustomerLoginPanelLayout);

799 CustomerLoginPanelLayout.setHorizontalGroup(

800 CustomerLoginPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

801 .addGroup(CustomerLoginPanelLayout.createSequentialGroup()

802 .addContainerGap(181, Short.*MAX\_VALUE*)

803 .addGroup(CustomerLoginPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

804 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, CustomerLoginPanelLayout.createSequentialGroup()

805 .addComponent(customerLoginPanelSubtitle)

806 .addGap(278, 278, 278))

807 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, CustomerLoginPanelLayout.createSequentialGroup()

808 .addGroup(CustomerLoginPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

809 .addGroup(CustomerLoginPanelLayout.createSequentialGroup()

810 .addComponent(customerLoginPanelTitle)

811 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*RELATED*, 141, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

812 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, CustomerLoginPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

813 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, CustomerLoginPanelLayout.createSequentialGroup()

814 .addGap(348, 348, 348)

815 .addComponent(customerSignUpOptionButton, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 89, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

816 .addGroup(CustomerLoginPanelLayout.createSequentialGroup()

817 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*RELATED*, 44, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

818 .addGroup(CustomerLoginPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

819 .addGroup(CustomerLoginPanelLayout.createSequentialGroup()

820 .addGap(60, 60, 60)

821 .addComponent(customerLoginButton, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 89, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

822 .addGroup(CustomerLoginPanelLayout.createSequentialGroup()

823 .addGroup(CustomerLoginPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*)

824 .addComponent(customerPasswordLabel)

825 .addComponent(customerUsernameLabel))

826 .addGap(18, 18, 18)

827 .addGroup(CustomerLoginPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*, false)

828 .addComponent(customerUsernameField, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 110, Short.*MAX\_VALUE*)

829 .addComponent(customerPasswordField)))))))

830 .addGap(32, 32, 32))))

831 );

832 CustomerLoginPanelLayout.setVerticalGroup(

833 CustomerLoginPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

834 .addGroup(CustomerLoginPanelLayout.createSequentialGroup()

835 .addGap(89, 89, 89)

836 .addComponent(customerLoginPanelTitle)

837 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*RELATED*)

838 .addComponent(customerLoginPanelSubtitle)

839 .addGap(38, 38, 38)

840 .addGroup(CustomerLoginPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

841 .addComponent(customerUsernameLabel)

842 .addComponent(customerUsernameField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

843 .addGap(18, 18, 18)

844 .addGroup(CustomerLoginPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*BASELINE*)

845 .addComponent(customerPasswordLabel)

846 .addComponent(customerPasswordField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

847 .addGap(27, 27, 27)

848 .addComponent(customerLoginButton, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 38, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

849 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*RELATED*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

850 .addComponent(customerSignUpOptionButton, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 38, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

851 .addGap(38, 38, 38))

852 );

853

854 CustomerMainMenuPanel.setPreferredSize(new java.awt.Dimension(650, 449));

855

856 CustomerMainMenuPanelTitle.setFont(new java.awt.Font("Tahoma", 0, 36)); // NOI18N

857 CustomerMainMenuPanelTitle.setText("Picadilly Bookstore");

858

859 CustomerMainMenuPanelSubtitle.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

860 CustomerMainMenuPanelSubtitle.setText("Customer Main Menu");

861

862 customerMainMenuLogoutButton.setText("Logout");

863 customerMainMenuLogoutButton.addActionListener(new java.awt.event.ActionListener() {

864 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

865 customerMainMenuLogoutButtonActionPerformed(evt);

866 }

867 });

868

869 customerBookTable.setModel(new javax.swing.table.DefaultTableModel(

870 new Object [][] {

871

872 },

873 new String [] {

874 "ISBN", "Title", "Author", "Quantity", "Price"

875 }

876 ));

877 jScrollPane4.setViewportView(customerBookTable);

878

879 CustomerISBNChoiceLabel.setText("ISBN:");

880

881 purchaseButton.setText("Purchase!");

882 purchaseButton.addActionListener(new java.awt.event.ActionListener() {

883 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

884 purchaseButtonActionPerformed(evt);

885 }

886 });

887

888 purchasingStatusLabel.setText("Status : ");

889

890 javax.swing.GroupLayout CustomerMainMenuPanelLayout = new javax.swing.GroupLayout(CustomerMainMenuPanel);

891 CustomerMainMenuPanel.setLayout(CustomerMainMenuPanelLayout);

892 CustomerMainMenuPanelLayout.setHorizontalGroup(

893 CustomerMainMenuPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

894 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, CustomerMainMenuPanelLayout.createSequentialGroup()

895 .addGroup(CustomerMainMenuPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*)

896 .addComponent(jScrollPane4, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 592, Short.*MAX\_VALUE*)

897 .addGroup(CustomerMainMenuPanelLayout.createSequentialGroup()

898 .addContainerGap(javax.swing.GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

899 .addComponent(customerMainMenuLogoutButton, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 89, javax.swing.GroupLayout.*PREFERRED\_SIZE*)))

900 .addGap(58, 58, 58))

901 .addGroup(CustomerMainMenuPanelLayout.createSequentialGroup()

902 .addGroup(CustomerMainMenuPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

903 .addGroup(CustomerMainMenuPanelLayout.createSequentialGroup()

904 .addGap(171, 171, 171)

905 .addComponent(CustomerMainMenuPanelTitle))

906 .addGroup(CustomerMainMenuPanelLayout.createSequentialGroup()

907 .addGap(236, 236, 236)

908 .addComponent(CustomerMainMenuPanelSubtitle))

909 .addGroup(CustomerMainMenuPanelLayout.createSequentialGroup()

910 .addContainerGap()

911 .addComponent(CustomerISBNChoiceLabel)

912 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

913 .addComponent(customerIsbnChoiceField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 158, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

914 .addGap(29, 29, 29)

915 .addComponent(purchaseButton))

916 .addGroup(CustomerMainMenuPanelLayout.createSequentialGroup()

917 .addContainerGap()

918 .addComponent(purchasingStatusLabel)))

919 .addContainerGap(javax.swing.GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*))

920 );

921 CustomerMainMenuPanelLayout.setVerticalGroup(

922 CustomerMainMenuPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

923 .addGroup(CustomerMainMenuPanelLayout.createSequentialGroup()

924 .addContainerGap()

925 .addComponent(CustomerMainMenuPanelTitle)

926 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

927 .addComponent(CustomerMainMenuPanelSubtitle)

928 .addGap(40, 40, 40)

929 .addComponent(customerMainMenuLogoutButton, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 37, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

930 .addGap(23, 23, 23)

931 .addGroup(CustomerMainMenuPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*BASELINE*)

932 .addComponent(purchaseButton)

933 .addComponent(CustomerISBNChoiceLabel)

934 .addComponent(customerIsbnChoiceField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

935 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*RELATED*)

936 .addComponent(purchasingStatusLabel)

937 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

938 .addComponent(jScrollPane4, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 144, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

939 .addContainerGap(57, Short.*MAX\_VALUE*))

940 );

941

942 CustomerEditProfilePanel.setPreferredSize(new java.awt.Dimension(650, 940));

943

944 jLabel51.setFont(new java.awt.Font("Tahoma", 0, 36)); // NOI18N

945 jLabel51.setText("Picadilly Bookstore");

946

947 jTextField26.addActionListener(new java.awt.event.ActionListener() {

948 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

949 jTextField26ActionPerformed(evt);

950 }

951 });

952

953 jLabel52.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

954 jLabel52.setText("Personal Profile");

955

956 jLabel53.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

957 jLabel53.setText("First Name:");

958

959 jLabel54.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

960 jLabel54.setText("Last Name:");

961

962 jTextField27.addActionListener(new java.awt.event.ActionListener() {

963 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

964 jTextField27ActionPerformed(evt);

965 }

966 });

967

968 jButton32.setText("Enter");

969

970 jTextField28.addActionListener(new java.awt.event.ActionListener() {

971 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

972 jTextField28ActionPerformed(evt);

973 }

974 });

975

976 jLabel55.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

977 jLabel55.setText("Gender:");

978

979 jLabel56.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

980 jLabel56.setText("Street:");

981

982 jLabel57.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

983 jLabel57.setText("City:");

984

985 jLabel58.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

986 jLabel58.setText("Zip Code:");

987

988 jTextField29.addActionListener(new java.awt.event.ActionListener() {

989 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

990 jTextField29ActionPerformed(evt);

991 }

992 });

993

994 jTextField30.addActionListener(new java.awt.event.ActionListener() {

995 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

996 jTextField30ActionPerformed(evt);

997 }

998 });

999

1000 jTextField31.addActionListener(new java.awt.event.ActionListener() {

1001 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1002 jTextField31ActionPerformed(evt);

1003 }

1004 });

1005

1006 jLabel59.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

1007 jLabel59.setText("Country:");

1008

1009 jLabel60.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

1010 jLabel60.setText("Phone Number:");

1011

1012 jLabel61.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

1013 jLabel61.setText("Email:");

1014

1015 jTextField32.addActionListener(new java.awt.event.ActionListener() {

1016 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1017 jTextField32ActionPerformed(evt);

1018 }

1019 });

1020

1021 jTextField33.addActionListener(new java.awt.event.ActionListener() {

1022 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1023 jTextField33ActionPerformed(evt);

1024 }

1025 });

1026

1027 jLabel62.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

1028 jLabel62.setText("Username:");

1029

1030 jLabel63.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

1031 jLabel63.setText("Old Password:");

1032

1033 jTextField34.addActionListener(new java.awt.event.ActionListener() {

1034 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1035 jTextField34ActionPerformed(evt);

1036 }

1037 });

1038

1039 jTextField35.addActionListener(new java.awt.event.ActionListener() {

1040 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1041 jTextField35ActionPerformed(evt);

1042 }

1043 });

1044

1045 jLabel64.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

1046 jLabel64.setText("Account Info");

1047

1048 jLabel65.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

1049 jLabel65.setText("Credit Card Holder:");

1050

1051 jLabel66.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

1052 jLabel66.setText("Credit Card Number:");

1053

1054 jTextField36.addActionListener(new java.awt.event.ActionListener() {

1055 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1056 jTextField36ActionPerformed(evt);

1057 }

1058 });

1059

1060 jTextField37.addActionListener(new java.awt.event.ActionListener() {

1061 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1062 jTextField37ActionPerformed(evt);

1063 }

1064 });

1065

1066 jLabel67.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

1067 jLabel67.setText("Month:");

1068

1069 jLabel68.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

1070 jLabel68.setText("CVC:");

1071

1072 jTextField38.addActionListener(new java.awt.event.ActionListener() {

1073 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1074 jTextField38ActionPerformed(evt);

1075 }

1076 });

1077

1078 jLabel69.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

1079 jLabel69.setText("Credit Card Profile");

1080

1081 jTextField39.addActionListener(new java.awt.event.ActionListener() {

1082 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1083 jTextField39ActionPerformed(evt);

1084 }

1085 });

1086

1087 jLabel70.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

1088 jLabel70.setText("Year:");

1089

1090 jTextField40.addActionListener(new java.awt.event.ActionListener() {

1091 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1092 jTextField40ActionPerformed(evt);

1093 }

1094 });

1095

1096 jLabel71.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

1097 jLabel71.setText("Edit Profile");

1098

1099 jCheckBox3.setText("Male");

1100 jCheckBox3.addActionListener(new java.awt.event.ActionListener() {

1101 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1102 jCheckBox3ActionPerformed(evt);

1103 }

1104 });

1105

1106 jCheckBox4.setText("Female");

1107 jCheckBox4.addActionListener(new java.awt.event.ActionListener() {

1108 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1109 jCheckBox4ActionPerformed(evt);

1110 }

1111 });

1112

1113 jLabel93.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

1114 jLabel93.setText("New Password:");

1115

1116 jTextField43.addActionListener(new java.awt.event.ActionListener() {

1117 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1118 jTextField43ActionPerformed(evt);

1119 }

1120 });

1121

1122 javax.swing.GroupLayout CustomerEditProfilePanelLayout = new javax.swing.GroupLayout(CustomerEditProfilePanel);

1123 CustomerEditProfilePanel.setLayout(CustomerEditProfilePanelLayout);

1124 CustomerEditProfilePanelLayout.setHorizontalGroup(

1125 CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1126 .addGroup(CustomerEditProfilePanelLayout.createSequentialGroup()

1127 .addGap(28, 28, 28)

1128 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1129 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1130 .addComponent(jSeparator3, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 588, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1131 .addGroup(CustomerEditProfilePanelLayout.createSequentialGroup()

1132 .addGap(201, 201, 201)

1133 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1134 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, CustomerEditProfilePanelLayout.createSequentialGroup()

1135 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*)

1136 .addComponent(jLabel54)

1137 .addComponent(jLabel53))

1138 .addGap(18, 18, 18)

1139 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1140 .addComponent(jTextField27, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 110, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1141 .addComponent(jTextField26, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 110, javax.swing.GroupLayout.*PREFERRED\_SIZE*)))

1142 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, CustomerEditProfilePanelLayout.createSequentialGroup()

1143 .addComponent(jLabel52)

1144 .addGap(63, 63, 63))))

1145 .addComponent(jSeparator4, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 588, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1146 .addGroup(CustomerEditProfilePanelLayout.createSequentialGroup()

1147 .addGap(260, 260, 260)

1148 .addComponent(jLabel69))

1149 .addGroup(CustomerEditProfilePanelLayout.createSequentialGroup()

1150 .addGap(128, 128, 128)

1151 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1152 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, CustomerEditProfilePanelLayout.createSequentialGroup()

1153 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*)

1154 .addComponent(jLabel66)

1155 .addComponent(jLabel65))

1156 .addGap(18, 18, 18)

1157 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1158 .addComponent(jTextField36, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 110, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1159 .addComponent(jTextField38, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 110, javax.swing.GroupLayout.*PREFERRED\_SIZE*)))

1160 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, CustomerEditProfilePanelLayout.createSequentialGroup()

1161 .addComponent(jLabel68)

1162 .addGap(18, 18, 18)

1163 .addComponent(jTextField37, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 110, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1164 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, CustomerEditProfilePanelLayout.createSequentialGroup()

1165 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*)

1166 .addComponent(jButton32, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 89, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1167 .addGroup(CustomerEditProfilePanelLayout.createSequentialGroup()

1168 .addComponent(jLabel67)

1169 .addGap(18, 18, 18)

1170 .addComponent(jTextField39, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 56, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1171 .addGap(18, 18, 18)

1172 .addComponent(jLabel70)))

1173 .addGap(18, 18, 18)

1174 .addComponent(jTextField40, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 56, javax.swing.GroupLayout.*PREFERRED\_SIZE*)))))

1175 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, CustomerEditProfilePanelLayout.createSequentialGroup()

1176 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*)

1177 .addGroup(CustomerEditProfilePanelLayout.createSequentialGroup()

1178 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*)

1179 .addComponent(jLabel56)

1180 .addComponent(jLabel55))

1181 .addGap(18, 18, 18)

1182 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1183 .addComponent(jTextField28, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 110, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1184 .addGroup(CustomerEditProfilePanelLayout.createSequentialGroup()

1185 .addComponent(jCheckBox3)

1186 .addGap(18, 18, 18)

1187 .addComponent(jCheckBox4))))

1188 .addGroup(CustomerEditProfilePanelLayout.createSequentialGroup()

1189 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*)

1190 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1191 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, CustomerEditProfilePanelLayout.createSequentialGroup()

1192 .addComponent(jLabel59)

1193 .addGap(18, 18, 18)

1194 .addComponent(jTextField31, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 110, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1195 .addGroup(CustomerEditProfilePanelLayout.createSequentialGroup()

1196 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*)

1197 .addComponent(jLabel58)

1198 .addComponent(jLabel57))

1199 .addGap(18, 18, 18)

1200 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1201 .addComponent(jTextField29, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 110, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1202 .addComponent(jTextField30, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 110, javax.swing.GroupLayout.*PREFERRED\_SIZE*))))

1203 .addGroup(CustomerEditProfilePanelLayout.createSequentialGroup()

1204 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*)

1205 .addComponent(jLabel61)

1206 .addComponent(jLabel60))

1207 .addGap(18, 18, 18)

1208 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1209 .addComponent(jTextField32, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 110, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1210 .addComponent(jTextField33, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 110, javax.swing.GroupLayout.*PREFERRED\_SIZE*))))

1211 .addGap(14, 14, 14)))

1212 .addGap(154, 154, 154))

1213 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, CustomerEditProfilePanelLayout.createSequentialGroup()

1214 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*)

1215 .addGroup(CustomerEditProfilePanelLayout.createSequentialGroup()

1216 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*)

1217 .addComponent(jLabel63)

1218 .addComponent(jLabel62))

1219 .addGap(18, 18, 18)

1220 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1221 .addComponent(jTextField34, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 110, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1222 .addComponent(jTextField35, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 110, javax.swing.GroupLayout.*PREFERRED\_SIZE*)))

1223 .addGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*, CustomerEditProfilePanelLayout.createSequentialGroup()

1224 .addGap(4, 4, 4)

1225 .addComponent(jLabel93)

1226 .addGap(18, 18, 18)

1227 .addComponent(jTextField43, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 110, javax.swing.GroupLayout.*PREFERRED\_SIZE*)))

1228 .addGap(168, 168, 168)))

1229 .addContainerGap(34, Short.*MAX\_VALUE*))

1230 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, CustomerEditProfilePanelLayout.createSequentialGroup()

1231 .addContainerGap(javax.swing.GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

1232 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1233 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, CustomerEditProfilePanelLayout.createSequentialGroup()

1234 .addComponent(jLabel51)

1235 .addGap(172, 172, 172))

1236 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, CustomerEditProfilePanelLayout.createSequentialGroup()

1237 .addComponent(jLabel71)

1238 .addGap(274, 274, 274))

1239 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, CustomerEditProfilePanelLayout.createSequentialGroup()

1240 .addComponent(jLabel64)

1241 .addGap(264, 264, 264))))

1242 );

1243 CustomerEditProfilePanelLayout.setVerticalGroup(

1244 CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1245 .addGroup(CustomerEditProfilePanelLayout.createSequentialGroup()

1246 .addContainerGap()

1247 .addComponent(jLabel51)

1248 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*RELATED*)

1249 .addComponent(jLabel71)

1250 .addGap(18, 18, 18)

1251 .addComponent(jLabel52)

1252 .addGap(18, 18, 18)

1253 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1254 .addComponent(jLabel53)

1255 .addComponent(jTextField27, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1256 .addGap(18, 18, 18)

1257 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1258 .addComponent(jLabel54)

1259 .addComponent(jTextField26, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1260 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1261 .addGroup(CustomerEditProfilePanelLayout.createSequentialGroup()

1262 .addGap(19, 19, 19)

1263 .addComponent(jLabel55))

1264 .addGroup(CustomerEditProfilePanelLayout.createSequentialGroup()

1265 .addGap(18, 18, 18)

1266 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*BASELINE*)

1267 .addComponent(jCheckBox3)

1268 .addComponent(jCheckBox4))))

1269 .addGap(18, 18, 18)

1270 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1271 .addComponent(jLabel56)

1272 .addComponent(jTextField28, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1273 .addGap(18, 18, 18)

1274 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1275 .addComponent(jLabel57)

1276 .addComponent(jTextField29, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1277 .addGap(18, 18, 18)

1278 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1279 .addComponent(jLabel58)

1280 .addComponent(jTextField30, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1281 .addGap(18, 18, 18)

1282 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1283 .addComponent(jLabel59)

1284 .addComponent(jTextField31, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1285 .addGap(18, 18, 18)

1286 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*BASELINE*)

1287 .addComponent(jTextField32, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1288 .addComponent(jLabel60))

1289 .addGap(20, 20, 20)

1290 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1291 .addComponent(jLabel61)

1292 .addComponent(jTextField33, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1293 .addGap(18, 18, 18)

1294 .addComponent(jSeparator3, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 8, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1295 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*RELATED*)

1296 .addComponent(jLabel64, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 17, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1297 .addGap(18, 18, 18)

1298 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*BASELINE*)

1299 .addComponent(jTextField34, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1300 .addComponent(jLabel62))

1301 .addGap(18, 18, 18)

1302 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1303 .addComponent(jLabel63)

1304 .addComponent(jTextField35, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1305 .addGap(18, 18, 18)

1306 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1307 .addComponent(jLabel93)

1308 .addComponent(jTextField43, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1309 .addGap(18, 18, 18)

1310 .addComponent(jSeparator4, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 8, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1311 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*RELATED*)

1312 .addComponent(jLabel69)

1313 .addGap(18, 18, 18)

1314 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1315 .addComponent(jLabel65)

1316 .addComponent(jTextField36, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1317 .addGap(18, 18, 18)

1318 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1319 .addComponent(jLabel66)

1320 .addComponent(jTextField38, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1321 .addGap(18, 18, 18)

1322 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1323 .addComponent(jLabel68)

1324 .addComponent(jTextField37, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1325 .addGap(18, 18, 18)

1326 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*)

1327 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1328 .addComponent(jTextField39, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1329 .addComponent(jLabel67))

1330 .addGroup(CustomerEditProfilePanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1331 .addComponent(jTextField40, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1332 .addComponent(jLabel70)))

1333 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*RELATED*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

1334 .addComponent(jButton32, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 38, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1335 .addGap(12, 12, 12))

1336 );

1337

1338 AdminAddBookPanel.setPreferredSize(new java.awt.Dimension(700, 520));

1339

1340 AdminAddBookPanelTitle.setFont(new java.awt.Font("Tahoma", 0, 36)); // NOI18N

1341 AdminAddBookPanelTitle.setText("Picadilly Bookstore");

1342

1343 AdminSubmitBook.setText("Enter");

1344 AdminSubmitBook.addActionListener(new java.awt.event.ActionListener() {

1345 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1346 AdminSubmitBookActionPerformed(evt);

1347 }

1348 });

1349

1350 AdminAddBookPanelSubtitle.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

1351 AdminAddBookPanelSubtitle.setText("Add Book");

1352

1353 bookQuantityLabel.setText("Quantity:");

1354

1355 isbnLabel.setText("ISBN:");

1356

1357 authorLabel.setText("Author:");

1358

1359 publishedYearLabel.setText("Published Year:");

1360

1361 bookTitleLabel.setText("Title:");

1362

1363 publishedNameLabel.setText("Publisher Name:");

1364

1365 descriptionLabel.setText("Description:");

1366

1367 bookPriceLabel.setText("Book Price:");

1368

1369 keywordsLabel.setText("Keywords:");

1370

1371 descriptionCharactersLimitLabel.setText("(max. 4000 characters)");

1372

1373 bookPriceCurrencyLabel.setText("In Rupiah");

1374

1375 bindingLabel.setText("Binding:");

1376

1377 signatureLabel.setText("Signature:");

1378

1379 signatureComboBox.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Signed", "Not Signed" }));

1380

1381 bindingComboBox.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Hard Cover", "Paperback" }));

1382

1383 categoryLabel.setText("Category:");

1384

1385 categoryField.addActionListener(new java.awt.event.ActionListener() {

1386 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1387 categoryFieldActionPerformed(evt);

1388 }

1389 });

1390

1391 javax.swing.GroupLayout AdminAddBookPanelLayout = new javax.swing.GroupLayout(AdminAddBookPanel);

1392 AdminAddBookPanel.setLayout(AdminAddBookPanelLayout);

1393 AdminAddBookPanelLayout.setHorizontalGroup(

1394 AdminAddBookPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1395 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, AdminAddBookPanelLayout.createSequentialGroup()

1396 .addGap(29, 29, 29)

1397 .addGroup(AdminAddBookPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1398 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, AdminAddBookPanelLayout.createSequentialGroup()

1399 .addComponent(isbnLabel)

1400 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

1401 .addComponent(isbnField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 215, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1402 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, AdminAddBookPanelLayout.createSequentialGroup()

1403 .addComponent(authorLabel)

1404 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

1405 .addComponent(authorField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 215, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1406 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, AdminAddBookPanelLayout.createSequentialGroup()

1407 .addComponent(bookTitleLabel)

1408 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

1409 .addComponent(bookTitleField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 215, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1410 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, AdminAddBookPanelLayout.createSequentialGroup()

1411 .addComponent(publishedNameLabel)

1412 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

1413 .addComponent(publisherNameField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 215, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1414 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, AdminAddBookPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*, false)

1415 .addGroup(AdminAddBookPanelLayout.createSequentialGroup()

1416 .addComponent(descriptionLabel)

1417 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

1418 .addGroup(AdminAddBookPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1419 .addComponent(descriptionCharactersLimitLabel)

1420 .addComponent(descriptionField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 215, javax.swing.GroupLayout.*PREFERRED\_SIZE*)))

1421 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, AdminAddBookPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1422 .addGroup(AdminAddBookPanelLayout.createSequentialGroup()

1423 .addComponent(bookPriceLabel)

1424 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

1425 .addComponent(bookPriceField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 101, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1426 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

1427 .addComponent(bookPriceCurrencyLabel))

1428 .addGroup(AdminAddBookPanelLayout.createSequentialGroup()

1429 .addComponent(keywordsLabel)

1430 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

1431 .addComponent(keywordsField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 215, javax.swing.GroupLayout.*PREFERRED\_SIZE*))))

1432 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, AdminAddBookPanelLayout.createSequentialGroup()

1433 .addComponent(publishedYearLabel)

1434 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

1435 .addComponent(publishedYearField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 215, javax.swing.GroupLayout.*PREFERRED\_SIZE*)))

1436 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*RELATED*, 27, Short.*MAX\_VALUE*)

1437 .addGroup(AdminAddBookPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*, false)

1438 .addGroup(AdminAddBookPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1439 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, AdminAddBookPanelLayout.createSequentialGroup()

1440 .addComponent(AdminSubmitBook, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 97, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1441 .addGap(75, 75, 75))

1442 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, AdminAddBookPanelLayout.createSequentialGroup()

1443 .addComponent(bindingLabel)

1444 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

1445 .addComponent(bindingComboBox, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 215, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1446 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, AdminAddBookPanelLayout.createSequentialGroup()

1447 .addComponent(signatureLabel)

1448 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

1449 .addComponent(signatureComboBox, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 215, javax.swing.GroupLayout.*PREFERRED\_SIZE*)))

1450 .addGroup(AdminAddBookPanelLayout.createSequentialGroup()

1451 .addComponent(categoryLabel)

1452 .addGap(18, 18, 18)

1453 .addComponent(categoryField)))

1454 .addGap(25, 25, 25))

1455 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, AdminAddBookPanelLayout.createSequentialGroup()

1456 .addContainerGap(javax.swing.GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

1457 .addGroup(AdminAddBookPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*)

1458 .addComponent(AdminAddBookPanelTitle, javax.swing.GroupLayout.*Alignment*.*LEADING*)

1459 .addGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*, AdminAddBookPanelLayout.createSequentialGroup()

1460 .addGap(109, 109, 109)

1461 .addComponent(AdminAddBookPanelSubtitle)))

1462 .addGap(198, 198, 198))

1463 .addGroup(AdminAddBookPanelLayout.createSequentialGroup()

1464 .addGap(68, 68, 68)

1465 .addComponent(bookQuantityLabel)

1466 .addGap(18, 18, 18)

1467 .addComponent(quantityField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 95, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1468 .addContainerGap(javax.swing.GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*))

1469 );

1470 AdminAddBookPanelLayout.setVerticalGroup(

1471 AdminAddBookPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1472 .addGroup(AdminAddBookPanelLayout.createSequentialGroup()

1473 .addContainerGap()

1474 .addComponent(AdminAddBookPanelTitle)

1475 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*RELATED*)

1476 .addComponent(AdminAddBookPanelSubtitle)

1477 .addGroup(AdminAddBookPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1478 .addGroup(AdminAddBookPanelLayout.createSequentialGroup()

1479 .addGap(51, 51, 51)

1480 .addGroup(AdminAddBookPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*BASELINE*)

1481 .addComponent(bindingLabel)

1482 .addComponent(bindingComboBox, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1483 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

1484 .addGroup(AdminAddBookPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*BASELINE*)

1485 .addComponent(signatureLabel)

1486 .addComponent(signatureComboBox, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1487 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

1488 .addGroup(AdminAddBookPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*BASELINE*)

1489 .addComponent(categoryLabel)

1490 .addComponent(categoryField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1491 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*RELATED*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

1492 .addComponent(AdminSubmitBook, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 38, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1493 .addGap(42, 42, 42))

1494 .addGroup(AdminAddBookPanelLayout.createSequentialGroup()

1495 .addGap(17, 17, 17)

1496 .addGroup(AdminAddBookPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*BASELINE*)

1497 .addComponent(bookQuantityLabel)

1498 .addComponent(quantityField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1499 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

1500 .addGroup(AdminAddBookPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*BASELINE*)

1501 .addComponent(isbnLabel)

1502 .addComponent(isbnField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1503 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

1504 .addGroup(AdminAddBookPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*BASELINE*)

1505 .addComponent(authorLabel)

1506 .addComponent(authorField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1507 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

1508 .addGroup(AdminAddBookPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*BASELINE*)

1509 .addComponent(bookTitleLabel)

1510 .addComponent(bookTitleField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1511 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

1512 .addGroup(AdminAddBookPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*BASELINE*)

1513 .addComponent(publishedNameLabel)

1514 .addComponent(publisherNameField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1515 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

1516 .addGroup(AdminAddBookPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*BASELINE*)

1517 .addComponent(publishedYearLabel)

1518 .addComponent(publishedYearField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1519 .addGap(51, 51, 51)

1520 .addGroup(AdminAddBookPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*BASELINE*)

1521 .addComponent(bookPriceLabel)

1522 .addComponent(bookPriceField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1523 .addComponent(bookPriceCurrencyLabel))

1524 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

1525 .addGroup(AdminAddBookPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*BASELINE*)

1526 .addComponent(keywordsLabel)

1527 .addComponent(keywordsField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1528 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

1529 .addGroup(AdminAddBookPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1530 .addComponent(descriptionField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 91, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1531 .addComponent(descriptionLabel))

1532 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*RELATED*)

1533 .addComponent(descriptionCharactersLimitLabel)

1534 .addGap(0, 0, Short.*MAX\_VALUE*))))

1535 );

1536

1537 CustomerSignUpPanel.setPreferredSize(new java.awt.Dimension(650, 900));

1538

1539 customerSignUpPanelTitle.setFont(new java.awt.Font("Tahoma", 0, 36)); // NOI18N

1540 customerSignUpPanelTitle.setText("Picadilly Bookstore");

1541

1542 customerLastNameField.addActionListener(new java.awt.event.ActionListener() {

1543 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1544 customerLastNameFieldActionPerformed(evt);

1545 }

1546 });

1547

1548 PersonalProfileLabel.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

1549 PersonalProfileLabel.setText("Personal Profile");

1550

1551 customerFirstNameLabel.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

1552 customerFirstNameLabel.setText("First Name:");

1553

1554 customerLastNameLabel.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

1555 customerLastNameLabel.setText("Last Name:");

1556

1557 customerFirstNameField.addActionListener(new java.awt.event.ActionListener() {

1558 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1559 customerFirstNameFieldActionPerformed(evt);

1560 }

1561 });

1562

1563 customerSignUpSubmitButton.setText("Enter");

1564 customerSignUpSubmitButton.addActionListener(new java.awt.event.ActionListener() {

1565 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1566 customerSignUpSubmitButtonActionPerformed(evt);

1567 }

1568 });

1569

1570 customerStreetField.addActionListener(new java.awt.event.ActionListener() {

1571 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1572 customerStreetFieldActionPerformed(evt);

1573 }

1574 });

1575

1576 customerGenderLabel.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

1577 customerGenderLabel.setText("Gender:");

1578

1579 customerStreetLabel.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

1580 customerStreetLabel.setText("Street:");

1581

1582 customerCityLabel.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

1583 customerCityLabel.setText("City:");

1584

1585 customerZipCodeLabel.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

1586 customerZipCodeLabel.setText("Zip Code:");

1587

1588 customerCityField.addActionListener(new java.awt.event.ActionListener() {

1589 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1590 customerCityFieldActionPerformed(evt);

1591 }

1592 });

1593

1594 customerZipCodeField.addActionListener(new java.awt.event.ActionListener() {

1595 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1596 customerZipCodeFieldActionPerformed(evt);

1597 }

1598 });

1599

1600 customerCountryField.addActionListener(new java.awt.event.ActionListener() {

1601 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1602 customerCountryFieldActionPerformed(evt);

1603 }

1604 });

1605

1606 customerCountryLabel.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

1607 customerCountryLabel.setText("Country:");

1608

1609 customerPhoneNumberLabel.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

1610 customerPhoneNumberLabel.setText("Phone Number:");

1611

1612 customerEmailLabel.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

1613 customerEmailLabel.setText("Email:");

1614

1615 customerPhoneNumberField.addActionListener(new java.awt.event.ActionListener() {

1616 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1617 customerPhoneNumberFieldActionPerformed(evt);

1618 }

1619 });

1620

1621 customerEmailField.addActionListener(new java.awt.event.ActionListener() {

1622 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1623 customerEmailFieldActionPerformed(evt);

1624 }

1625 });

1626

1627 customerAccountUsernameLabel.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

1628 customerAccountUsernameLabel.setText("Username:");

1629

1630 customerAccountPasswordLabel.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

1631 customerAccountPasswordLabel.setText("Password:");

1632

1633 customerAccountUsernameField.addActionListener(new java.awt.event.ActionListener() {

1634 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1635 customerAccountUsernameFieldActionPerformed(evt);

1636 }

1637 });

1638

1639 AccountInfoLabel.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

1640 AccountInfoLabel.setText("Account Info");

1641

1642 customerCreditCardNameLabel.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

1643 customerCreditCardNameLabel.setText("Credit Card Holder:");

1644

1645 customerCreditCardNumberLabel.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

1646 customerCreditCardNumberLabel.setText("Credit Card Number:");

1647

1648 customerCreditCardNameField.addActionListener(new java.awt.event.ActionListener() {

1649 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1650 customerCreditCardNameFieldActionPerformed(evt);

1651 }

1652 });

1653

1654 customerCvcField.addActionListener(new java.awt.event.ActionListener() {

1655 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1656 customerCvcFieldActionPerformed(evt);

1657 }

1658 });

1659

1660 customerCreditCardMonthLabel.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

1661 customerCreditCardMonthLabel.setText("Month:");

1662

1663 customerCvcLabel.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

1664 customerCvcLabel.setText("CVC:");

1665

1666 customerCreditCardNumberField.addActionListener(new java.awt.event.ActionListener() {

1667 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1668 customerCreditCardNumberFieldActionPerformed(evt);

1669 }

1670 });

1671

1672 creditCardProfileLabel.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

1673 creditCardProfileLabel.setText("Credit Card Profile");

1674

1675 customerCreditCardMonthField.addActionListener(new java.awt.event.ActionListener() {

1676 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1677 customerCreditCardMonthFieldActionPerformed(evt);

1678 }

1679 });

1680

1681 customerCreditCardYearLabel.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

1682 customerCreditCardYearLabel.setText("Year:");

1683

1684 customerCreditCardYearField.addActionListener(new java.awt.event.ActionListener() {

1685 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1686 customerCreditCardYearFieldActionPerformed(evt);

1687 }

1688 });

1689

1690 customerGenderComboBoxes.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Male", "Female" }));

1691 customerGenderComboBoxes.addActionListener(new java.awt.event.ActionListener() {

1692 public void **actionPerformed**(java.awt.event.ActionEvent evt) {

1693 customerGenderComboBoxesActionPerformed(evt);

1694 }

1695 });

1696

1697 javax.swing.GroupLayout CustomerSignUpPanelLayout = new javax.swing.GroupLayout(CustomerSignUpPanel);

1698 CustomerSignUpPanel.setLayout(CustomerSignUpPanelLayout);

1699 CustomerSignUpPanelLayout.setHorizontalGroup(

1700 CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1701 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, CustomerSignUpPanelLayout.createSequentialGroup()

1702 .addGap(0, 34, Short.*MAX\_VALUE*)

1703 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*)

1704 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*, false)

1705 .addGroup(CustomerSignUpPanelLayout.createSequentialGroup()

1706 .addGap(12, 12, 12)

1707 .addComponent(customerCreditCardNameLabel)

1708 .addGap(18, 18, 18)

1709 .addComponent(customerCreditCardNameField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 110, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1710 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, false)

1711 .addGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*, CustomerSignUpPanelLayout.createSequentialGroup()

1712 .addComponent(customerCreditCardMonthLabel)

1713 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*UNRELATED*)

1714 .addComponent(customerCreditCardMonthField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 56, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1715 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*RELATED*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

1716 .addComponent(customerCreditCardYearLabel)

1717 .addGap(18, 18, 18)

1718 .addComponent(customerCreditCardYearField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 56, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1719 .addGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*, CustomerSignUpPanelLayout.createSequentialGroup()

1720 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*)

1721 .addComponent(customerCvcLabel)

1722 .addComponent(customerCreditCardNumberLabel))

1723 .addGap(18, 18, 18)

1724 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1725 .addComponent(customerCreditCardNumberField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 110, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1726 .addComponent(customerCvcField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 110, javax.swing.GroupLayout.*PREFERRED\_SIZE*))))

1727 .addComponent(customerSignUpPanelSeparator2, javax.swing.GroupLayout.*Alignment*.*TRAILING*, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 588, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1728 .addComponent(customerSignUpPanelSeparator1, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 588, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1729 .addGap(28, 28, 28))

1730 .addGroup(CustomerSignUpPanelLayout.createSequentialGroup()

1731 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1732 .addGroup(CustomerSignUpPanelLayout.createSequentialGroup()

1733 .addGap(314, 314, 314)

1734 .addComponent(AccountInfoLabel))

1735 .addGroup(CustomerSignUpPanelLayout.createSequentialGroup()

1736 .addGap(239, 239, 239)

1737 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*)

1738 .addGroup(CustomerSignUpPanelLayout.createSequentialGroup()

1739 .addComponent(PersonalProfileLabel)

1740 .addGap(63, 63, 63))

1741 .addGroup(CustomerSignUpPanelLayout.createSequentialGroup()

1742 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*)

1743 .addComponent(customerLastNameLabel)

1744 .addComponent(customerFirstNameLabel)

1745 .addComponent(customerGenderLabel))

1746 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1747 .addGroup(CustomerSignUpPanelLayout.createSequentialGroup()

1748 .addGap(28, 28, 28)

1749 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*, false)

1750 .addComponent(customerFirstNameField, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 110, Short.*MAX\_VALUE*)

1751 .addComponent(customerLastNameField, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 110, Short.*MAX\_VALUE*)))

1752 .addGroup(CustomerSignUpPanelLayout.createSequentialGroup()

1753 .addGap(18, 18, 18)

1754 .addComponent(customerGenderComboBoxes, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 110, javax.swing.GroupLayout.*PREFERRED\_SIZE*))))))

1755 .addGroup(CustomerSignUpPanelLayout.createSequentialGroup()

1756 .addGap(207, 207, 207)

1757 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1758 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*)

1759 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1760 .addGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*, CustomerSignUpPanelLayout.createSequentialGroup()

1761 .addComponent(customerCountryLabel)

1762 .addGap(18, 18, 18)

1763 .addComponent(customerCountryField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 110, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1764 .addGroup(CustomerSignUpPanelLayout.createSequentialGroup()

1765 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*)

1766 .addComponent(customerZipCodeLabel)

1767 .addComponent(customerCityLabel)

1768 .addComponent(customerStreetLabel))

1769 .addGap(18, 18, 18)

1770 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1771 .addComponent(customerCityField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 110, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1772 .addComponent(customerZipCodeField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 110, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1773 .addComponent(customerStreetField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 110, javax.swing.GroupLayout.*PREFERRED\_SIZE*))))

1774 .addGroup(CustomerSignUpPanelLayout.createSequentialGroup()

1775 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*)

1776 .addComponent(customerEmailLabel)

1777 .addComponent(customerPhoneNumberLabel))

1778 .addGap(18, 18, 18)

1779 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1780 .addComponent(customerPhoneNumberField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 110, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1781 .addComponent(customerEmailField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 110, javax.swing.GroupLayout.*PREFERRED\_SIZE*)))

1782 .addGroup(CustomerSignUpPanelLayout.createSequentialGroup()

1783 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1784 .addComponent(customerAccountUsernameLabel, javax.swing.GroupLayout.*Alignment*.*TRAILING*)

1785 .addComponent(customerAccountPasswordLabel))

1786 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*RELATED*)

1787 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*, false)

1788 .addComponent(customerAccountPasswordField)

1789 .addComponent(customerAccountUsernameField, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 110, Short.*MAX\_VALUE*))))

1790 .addComponent(customerSignUpPanelTitle)))

1791 .addGroup(CustomerSignUpPanelLayout.createSequentialGroup()

1792 .addGap(300, 300, 300)

1793 .addComponent(creditCardProfileLabel))

1794 .addGroup(CustomerSignUpPanelLayout.createSequentialGroup()

1795 .addGap(310, 310, 310)

1796 .addComponent(customerSignUpSubmitButton, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 89, javax.swing.GroupLayout.*PREFERRED\_SIZE*)))

1797 .addContainerGap(javax.swing.GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*))

1798 );

1799 CustomerSignUpPanelLayout.setVerticalGroup(

1800 CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1801 .addGroup(CustomerSignUpPanelLayout.createSequentialGroup()

1802 .addContainerGap()

1803 .addComponent(customerSignUpPanelTitle)

1804 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*RELATED*)

1805 .addComponent(PersonalProfileLabel)

1806 .addGap(18, 18, 18)

1807 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1808 .addComponent(customerFirstNameLabel)

1809 .addComponent(customerFirstNameField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1810 .addGap(18, 18, 18)

1811 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1812 .addComponent(customerLastNameLabel)

1813 .addComponent(customerLastNameField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1814 .addGap(18, 18, 18)

1815 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*TRAILING*)

1816 .addComponent(customerGenderLabel)

1817 .addComponent(customerGenderComboBoxes, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1818 .addGap(11, 11, 11)

1819 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*BASELINE*)

1820 .addComponent(customerStreetField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1821 .addComponent(customerStreetLabel))

1822 .addGap(18, 18, 18)

1823 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1824 .addComponent(customerCityLabel)

1825 .addComponent(customerCityField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1826 .addGap(18, 18, 18)

1827 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1828 .addComponent(customerZipCodeLabel)

1829 .addComponent(customerZipCodeField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1830 .addGap(18, 18, 18)

1831 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1832 .addComponent(customerCountryLabel)

1833 .addComponent(customerCountryField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1834 .addGap(18, 18, 18)

1835 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*BASELINE*)

1836 .addComponent(customerPhoneNumberField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1837 .addComponent(customerPhoneNumberLabel))

1838 .addGap(20, 20, 20)

1839 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1840 .addComponent(customerEmailLabel)

1841 .addComponent(customerEmailField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1842 .addGap(18, 18, 18)

1843 .addComponent(customerSignUpPanelSeparator1, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 8, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1844 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*RELATED*)

1845 .addComponent(AccountInfoLabel, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 17, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1846 .addGap(16, 16, 16)

1847 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*BASELINE*)

1848 .addComponent(customerAccountUsernameField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1849 .addComponent(customerAccountUsernameLabel))

1850 .addGap(18, 18, 18)

1851 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1852 .addComponent(customerAccountPasswordField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1853 .addComponent(customerAccountPasswordLabel))

1854 .addGap(18, 18, 18)

1855 .addComponent(customerSignUpPanelSeparator2, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 8, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1856 .addPreferredGap(javax.swing.LayoutStyle.*ComponentPlacement*.*RELATED*)

1857 .addComponent(creditCardProfileLabel)

1858 .addGap(18, 18, 18)

1859 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*BASELINE*)

1860 .addComponent(customerCreditCardNameField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1861 .addComponent(customerCreditCardNameLabel))

1862 .addGap(18, 18, 18)

1863 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*BASELINE*)

1864 .addComponent(customerCreditCardNumberLabel)

1865 .addComponent(customerCreditCardNumberField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1866 .addGap(19, 19, 19)

1867 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*BASELINE*)

1868 .addComponent(customerCvcField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1869 .addComponent(customerCvcLabel))

1870 .addGap(18, 18, 18)

1871 .addGroup(CustomerSignUpPanelLayout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*BASELINE*)

1872 .addComponent(customerCreditCardMonthField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1873 .addComponent(customerCreditCardMonthLabel)

1874 .addComponent(customerCreditCardYearLabel)

1875 .addComponent(customerCreditCardYearField, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

1876 .addGap(18, 18, 18)

1877 .addComponent(customerSignUpSubmitButton, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 38, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

1878 .addContainerGap(javax.swing.GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*))

1879 );

1880

1881 CustomerSignUpScrollPane.setViewportView(CustomerSignUpPanel);

1882

1883 javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

1884 getContentPane().setLayout(layout);

1885 layout.setHorizontalGroup(

1886 layout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1887 .addComponent(UserChoicePanel, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 832, Short.*MAX\_VALUE*)

1888 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1889 .addGroup(layout.createSequentialGroup()

1890 .addContainerGap()

1891 .addComponent(AdminLoginPanel, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 820, Short.*MAX\_VALUE*)

1892 .addContainerGap()))

1893 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1894 .addGroup(layout.createSequentialGroup()

1895 .addContainerGap()

1896 .addComponent(AdminMenuPanel, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 820, Short.*MAX\_VALUE*)

1897 .addContainerGap()))

1898 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1899 .addGroup(layout.createSequentialGroup()

1900 .addContainerGap()

1901 .addComponent(AdminTransactionsListPanel, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 806, Short.*MAX\_VALUE*)

1902 .addGap(20, 20, 20)))

1903 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1904 .addGroup(layout.createSequentialGroup()

1905 .addContainerGap()

1906 .addComponent(AdminBooksListPanel, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 806, Short.*MAX\_VALUE*)

1907 .addGap(20, 20, 20)))

1908 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1909 .addGroup(layout.createSequentialGroup()

1910 .addContainerGap()

1911 .addComponent(AdminCustomersListPanel, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 806, Short.*MAX\_VALUE*)

1912 .addGap(20, 20, 20)))

1913 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1914 .addGroup(layout.createSequentialGroup()

1915 .addContainerGap()

1916 .addComponent(CustomerLoginPanel, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 806, Short.*MAX\_VALUE*)

1917 .addGap(20, 20, 20)))

1918 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1919 .addComponent(CustomerSignUpScrollPane))

1920 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1921 .addGroup(layout.createSequentialGroup()

1922 .addGap(20, 20, 20)

1923 .addComponent(CustomerMainMenuPanel, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 806, Short.*MAX\_VALUE*)

1924 .addContainerGap()))

1925 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1926 .addGroup(layout.createSequentialGroup()

1927 .addGap(20, 20, 20)

1928 .addComponent(CustomerEditProfilePanel, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 806, Short.*MAX\_VALUE*)

1929 .addContainerGap()))

1930 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1931 .addGroup(layout.createSequentialGroup()

1932 .addContainerGap()

1933 .addComponent(AdminAddBookPanel, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 730, Short.*MAX\_VALUE*)

1934 .addGap(96, 96, 96)))

1935 );

1936 layout.setVerticalGroup(

1937 layout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1938 .addComponent(UserChoicePanel, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 1455, Short.*MAX\_VALUE*)

1939 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1940 .addGroup(layout.createSequentialGroup()

1941 .addGap(10, 10, 10)

1942 .addComponent(AdminLoginPanel, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 1439, Short.*MAX\_VALUE*)

1943 .addContainerGap()))

1944 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1945 .addGroup(layout.createSequentialGroup()

1946 .addContainerGap()

1947 .addComponent(AdminMenuPanel, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 1443, Short.*MAX\_VALUE*)

1948 .addContainerGap()))

1949 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1950 .addGroup(layout.createSequentialGroup()

1951 .addContainerGap()

1952 .addComponent(AdminTransactionsListPanel, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 1436, Short.*MAX\_VALUE*)

1953 .addGap(13, 13, 13)))

1954 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1955 .addGroup(layout.createSequentialGroup()

1956 .addContainerGap()

1957 .addComponent(AdminBooksListPanel, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 1435, Short.*MAX\_VALUE*)

1958 .addGap(14, 14, 14)))

1959 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1960 .addGroup(layout.createSequentialGroup()

1961 .addContainerGap()

1962 .addComponent(AdminCustomersListPanel, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 1435, Short.*MAX\_VALUE*)

1963 .addGap(14, 14, 14)))

1964 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1965 .addGroup(layout.createSequentialGroup()

1966 .addContainerGap()

1967 .addComponent(CustomerLoginPanel, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 1434, Short.*MAX\_VALUE*)

1968 .addGap(15, 15, 15)))

1969 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1970 .addComponent(CustomerSignUpScrollPane))

1971 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1972 .addGroup(layout.createSequentialGroup()

1973 .addGap(21, 21, 21)

1974 .addComponent(CustomerMainMenuPanel, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 1431, Short.*MAX\_VALUE*)

1975 .addGap(3, 3, 3)))

1976 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1977 .addGroup(layout.createSequentialGroup()

1978 .addGap(21, 21, 21)

1979 .addComponent(CustomerEditProfilePanel, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 1429, Short.*MAX\_VALUE*)

1980 .addGap(5, 5, 5)))

1981 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.*Alignment*.*LEADING*)

1982 .addGroup(layout.createSequentialGroup()

1983 .addContainerGap()

1984 .addComponent(AdminAddBookPanel, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 944, Short.*MAX\_VALUE*)

1985 .addGap(505, 505, 505)))

1986 );

1987

1988 pack();

1989 }// </editor-fold>

1990

1991 private void **hideAllPanel**() {

1992 UserChoicePanel.setVisible(false);

1993 AdminLoginPanel.setVisible(false);

1994 AdminMenuPanel.setVisible(false);

1995 AdminTransactionsListPanel.setVisible(false);

1996 AdminBooksListPanel.setVisible(false);

1997 AdminCustomersListPanel.setVisible(false);

1998 CustomerLoginPanel.setVisible(false);

1999 CustomerSignUpScrollPane.setVisible(false);

2000 CustomerSignUpPanel.setVisible(false);

2001 CustomerMainMenuPanel.setVisible(false);

2002 CustomerEditProfilePanel.setVisible(false);

2003 AdminAddBookPanel.setVisible(false);

2004 }

2005 private void **adminButtonActionPerformed**(java.awt.event.ActionEvent evt) {

2006 this.hideAllPanel();

2007 AdminLoginPanel.setVisible(true);

2008 }

2009

2010 private void **adminUsernameFieldActionPerformed**(java.awt.event.ActionEvent evt) {

2011 this.hideAllPanel();

2012 CustomerLoginPanel.setVisible(true);

2013 }

2014

2015 private void **adminTransactionsListLogoutButtonActionPerformed**(java.awt.event.ActionEvent evt) {

2016 this.hideAllPanel();

2017 UserChoicePanel.setVisible(true);

2018 }

2019

2020 private void **adminBooksListLogoutButtonActionPerformed**(java.awt.event.ActionEvent evt) {

2021 this.hideAllPanel();

2022 UserChoicePanel.setVisible(true);

2023 }

2024

2025 private void **adminCustomersListLogoutButtonActionPerformed**(java.awt.event.ActionEvent evt) {

2026 this.hideAllPanel();

2027 UserChoicePanel.setVisible(true);

2028 }

2029

2030 private void **customerUsernameFieldActionPerformed**(java.awt.event.ActionEvent evt) {

2031 // TODO add your handling code here:

2032 }

2033

2034 private void **customerLastNameFieldActionPerformed**(java.awt.event.ActionEvent evt) {

2035 // TODO add your handling code here:

2036 }

2037

2038 private void **customerFirstNameFieldActionPerformed**(java.awt.event.ActionEvent evt) {

2039 // TODO add your handling code here:

2040 }

2041

2042 private void **customerStreetFieldActionPerformed**(java.awt.event.ActionEvent evt) {

2043 // TODO add your handling code here:

2044 }

2045

2046 private void **customerCityFieldActionPerformed**(java.awt.event.ActionEvent evt) {

2047 // TODO add your handling code here:

2048 }

2049

2050 private void **customerZipCodeFieldActionPerformed**(java.awt.event.ActionEvent evt) {

2051 // TODO add your handling code here:

2052 }

2053

2054 private void **customerCountryFieldActionPerformed**(java.awt.event.ActionEvent evt) {

2055 // TODO add your handling code here:

2056 }

2057

2058 private void **customerPhoneNumberFieldActionPerformed**(java.awt.event.ActionEvent evt) {

2059 // TODO add your handling code here:

2060 }

2061

2062 private void **customerEmailFieldActionPerformed**(java.awt.event.ActionEvent evt) {

2063 // TODO add your handling code here:

2064 }

2065

2066 private void **customerAccountUsernameFieldActionPerformed**(java.awt.event.ActionEvent evt) {

2067 // TODO add your handling code here:

2068 }

2069

2070 private void **customerCreditCardNameFieldActionPerformed**(java.awt.event.ActionEvent evt) {

2071 // TODO add your handling code here:

2072 }

2073

2074 private void **customerCvcFieldActionPerformed**(java.awt.event.ActionEvent evt) {

2075 // TODO add your handling code here:

2076 }

2077

2078 private void **customerCreditCardNumberFieldActionPerformed**(java.awt.event.ActionEvent evt) {

2079 // TODO add your handling code here:

2080 }

2081

2082 private void **customerCreditCardMonthFieldActionPerformed**(java.awt.event.ActionEvent evt) {

2083 // TODO add your handling code here:

2084 }

2085

2086 private void **customerCreditCardYearFieldActionPerformed**(java.awt.event.ActionEvent evt) {

2087 // TODO add your handling code here:

2088 }

2089

2090 private void **customerMainMenuLogoutButtonActionPerformed**(java.awt.event.ActionEvent evt) {

2091 this.hideAllPanel();

2092 UserChoicePanel.setVisible(true);

2093 }

2094

2095 private void **jTextField26ActionPerformed**(java.awt.event.ActionEvent evt) {

2096 // TODO add your handling code here:

2097 }

2098

2099 private void **jTextField27ActionPerformed**(java.awt.event.ActionEvent evt) {

2100 // TODO add your handling code here:

2101 }

2102

2103 private void **jTextField28ActionPerformed**(java.awt.event.ActionEvent evt) {

2104 // TODO add your handling code here:

2105 }

2106

2107 private void **jTextField29ActionPerformed**(java.awt.event.ActionEvent evt) {

2108 // TODO add your handling code here:

2109 }

2110

2111 private void **jTextField30ActionPerformed**(java.awt.event.ActionEvent evt) {

2112 // TODO add your handling code here:

2113 }

2114

2115 private void **jTextField31ActionPerformed**(java.awt.event.ActionEvent evt) {

2116 // TODO add your handling code here:

2117 }

2118

2119 private void **jTextField32ActionPerformed**(java.awt.event.ActionEvent evt) {

2120 // TODO add your handling code here:

2121 }

2122

2123 private void **jTextField33ActionPerformed**(java.awt.event.ActionEvent evt) {

2124 // TODO add your handling code here:

2125 }

2126

2127 private void **jTextField34ActionPerformed**(java.awt.event.ActionEvent evt) {

2128 // TODO add your handling code here:

2129 }

2130

2131 private void **jTextField35ActionPerformed**(java.awt.event.ActionEvent evt) {

2132 // TODO add your handling code here:

2133 }

2134

2135 private void **jTextField36ActionPerformed**(java.awt.event.ActionEvent evt) {

2136 // TODO add your handling code here:

2137 }

2138

2139 private void **jTextField37ActionPerformed**(java.awt.event.ActionEvent evt) {

2140 // TODO add your handling code here:

2141 }

2142

2143 private void **jTextField38ActionPerformed**(java.awt.event.ActionEvent evt) {

2144 // TODO add your handling code here:

2145 }

2146

2147 private void **jTextField39ActionPerformed**(java.awt.event.ActionEvent evt) {

2148 // TODO add your handling code here:

2149 }

2150

2151 private void **jTextField40ActionPerformed**(java.awt.event.ActionEvent evt) {

2152 // TODO add your handling code here:

2153 }

2154

2155 private void **jCheckBox3ActionPerformed**(java.awt.event.ActionEvent evt) {

2156 // TODO add your handling code here:

2157 }

2158

2159 private void **jCheckBox4ActionPerformed**(java.awt.event.ActionEvent evt) {

2160 // TODO add your handling code here:

2161 }

2162

2163 private void **jTextField43ActionPerformed**(java.awt.event.ActionEvent evt) {

2164 // TODO add your handling code here:

2165 }

2166

2167 private void **adminMenuBooksButtonActionPerformed**(java.awt.event.ActionEvent evt) {

2168 this.hideAllPanel();

2169 AdminBooksListPanel.setVisible(true);

2170 this.displayAdminBookList();

2171

2172 }

2173

2174 private void **adminLoginButtonActionPerformed**(java.awt.event.ActionEvent evt) {

2175 String username = adminUsernameField.getText();

2176 String password = adminPasswordField.getText();

2177 Admins adminList = (Admins) DAO.*getAllAdmins*();

2178 // sequential search. Complexity O(n)

2179 for (Admin adminChecked : adminList.getAdmins())

2180 {

2181 if (adminChecked.getUsername().equals(username)) {

2182 if (adminChecked.getPassword().equals(password)) {

2183 this.hideAllPanel();

2184 AdminMenuPanel.setVisible(true);

2185 }

2186 else {

2187 adminUsernameField.setText("");

2188 adminPasswordField.setText("");

2189 }

2190 }

2191 else {

2192 adminUsernameField.setText("");

2193 adminPasswordField.setText("");

2194 }

2195 }

2196 }

2197

2198 private void **customerLoginButtonActionPerformed**(java.awt.event.ActionEvent evt) {

2199 String username = customerUsernameField.getText();

2200 String password = customerPasswordField.getText();

2201 Customers customerList = (Customers) DAO.*getAllCustomers*();

2202 // sequential search. Complexity O(n)

2203 for (Customer customerChecked : customerList.getCustomers())

2204 {

2205 if (customerChecked.getUsername().equals(username)) {

2206 if (customerChecked.getPassword().equals(password)) {

2207 this.hideAllPanel();

2208 CustomerMainMenuPanel.setVisible(true);

2209 this.displayCustomerBookList();

2210 currentCustomer = DAO.*searchCustomer*(username);

2211 }

2212 else {

2213 customerUsernameField.setText("");

2214 customerPasswordField.setText("");

2215 }

2216 }

2217 else {

2218 customerUsernameField.setText("");

2219 customerPasswordField.setText("");

2220 }

2221 }

2222 }

2223

2224 private void **customerButtonActionPerformed**(java.awt.event.ActionEvent evt) {

2225 this.hideAllPanel();

2226 CustomerLoginPanel.setVisible(true);

2227 }

2228

2229 private void **customerSignUpOptionButtonActionPerformed**(java.awt.event.ActionEvent evt) {

2230 this.hideAllPanel();

2231 CustomerSignUpScrollPane.setVisible(true);

2232 CustomerSignUpPanel.setVisible(true);

2233 }

2234

2235 private void **customerGenderComboBoxesActionPerformed**(java.awt.event.ActionEvent evt) {

2236 // TODO add your handling code here:

2237 }

2238

2239 private void **customerSignUpSubmitButtonActionPerformed**(java.awt.event.ActionEvent evt) {

2240 Customer customer = new Customer();

2241 String fullName = customerFirstNameField.getText() + " " + customerLastNameField.getText();

2242 customer.setUserId((long) DAO.*numberOfCustomers*()+1);

2243 customer.setName(fullName);

2244

2245 String customerGender = (String) customerGenderComboBoxes.getSelectedItem();

2246 if (customerGender.equalsIgnoreCase("male")) {

2247 customer.setGender(Gender.*MALE*);

2248 }

2249 if (customerGender.equalsIgnoreCase("female")) {

2250 customer.setGender(Gender.*FEMALE*);

2251 }

2252

2253 customer.setUsername(customerAccountUsernameField.getText());

2254 customer.setPassword(customerAccountPasswordField.getText());

2255 customer.setEmail(customerEmailField.getText());

2256 customer.setPhoneNumber(customerPhoneNumberField.getText());

2257 customer.setNumberOfPurchase(0);

2258 customer.setStreetAddress(customerStreetField.getText());

2259 customer.setCity(customerCityField.getText());

2260 customer.setZipCode(customerZipCodeField.getText());

2261 customer.setCountry(customerCountryField.getText());

2262 customer.setCreditCardName(customerCreditCardNameField.getText());

2263 customer.setCreditCardNo(customerCreditCardNumberField.getText());

2264 customer.setCvc(customerCvcField.getText());

2265 customer.setCreditCardMonth(customerCreditCardMonthField.getText());

2266 customer.setCreditCardYear(customerCreditCardYearField.getText());

2267 DAO.*addCustomer*(customer);

2268 this.hideAllPanel();

2269 CustomerLoginPanel.setVisible(true);

2270

2271

2272 }

2273

2274 private void **adminMenuCustomersButtonActionPerformed**(java.awt.event.ActionEvent evt) {

2275 this.hideAllPanel();

2276 AdminCustomersListPanel.setVisible(true);

2277 this.displayAdminCustomerList();

2278 }

2279

2280 private void **adminMenuTransactionsButtonActionPerformed**(java.awt.event.ActionEvent evt) {

2281 this.hideAllPanel();

2282 AdminTransactionsListPanel.setVisible(true);

2283 this.displayAdminTransactionList();

2284 }

2285

2286 private void **addBookButtonActionPerformed**(java.awt.event.ActionEvent evt) {

2287 this.hideAllPanel();

2288 AdminAddBookPanel.setVisible(true);

2289 }

2290

2291 private void **categoryFieldActionPerformed**(java.awt.event.ActionEvent evt) {

2292 // TODO add your handling code here:

2293 }

2294

2295 private void **AdminSubmitBookActionPerformed**(java.awt.event.ActionEvent evt) {

2296 Book book = new Book();

2297

2298 book.setAuthor(authorField.getText());

2299 book.setTitle(bookTitleField.getText());

2300 book.setIsbn(isbnField.getText());

2301

2302 long price = Long.*valueOf*(bookPriceField.getText());

2303 book.setPrice(price);

2304

2305 long quantity = Long.*valueOf*(quantityField.getText());

2306 book.setQuantity(quantity);

2307

2308 book.setPublisherName(publisherNameField.getText());

2309

2310 int yearPublished = Integer.*parseInt*(publishedYearField.getText());

2311 book.setYearPublished(yearPublished);

2312

2313 book.setDescription(descriptionField.getText());

2314

2315 String bookBinding = (String) bindingComboBox.getSelectedItem();

2316 if (bookBinding.equalsIgnoreCase("paperback")) {

2317 book.setBinding(Binding.*PAPERBACK*);

2318 }

2319 if (bookBinding.equalsIgnoreCase("hard cover")) {

2320 book.setBinding(Binding.*HARDCOVER*);

2321 }

2322

2323

2324 String bookSignature = (String) signatureComboBox.getSelectedItem();

2325 if (bookSignature.equalsIgnoreCase("signed")) {

2326 book.setSigned(Signed.*SIGNED*);

2327 }

2328 if (bookSignature.equalsIgnoreCase("not signed")) {

2329 book.setSigned(Signed.*NOTSIGNED*);

2330 }

2331

2332

2333 book.setKeywords(Arrays.*asList*(keywordsField.getText()));

2334 book.setCategory(Arrays.*asList*(categoryField.getText()));

2335 DAO.*addBook*(book);

2336 this.hideAllPanel();

2337 AdminBooksListPanel.setVisible(true);

2338 this.displayAdminBookList();

2339

2340 }

2341

2342 private void **adminBooksListMenuButtonActionPerformed**(java.awt.event.ActionEvent evt) {

2343 this.hideAllPanel();

2344 AdminMenuPanel.setVisible(true);

2345 }

2346

2347 private void **AdminDeleteBookButtonActionPerformed**(java.awt.event.ActionEvent evt) {

2348 String deletedBookIsbn = isbnToDeleteField.getText();

2349 DAO.*deleteBook*(deletedBookIsbn);

2350 isbnToDeleteField.setText("");

2351 this.displayAdminBookList();

2352 }

2353

2354 private void **isbnToDeleteFieldActionPerformed**(java.awt.event.ActionEvent evt) {

2355 // TODO add your handling code here:

2356 }

2357

2358 private void **adminTransactionsListMenuButtonActionPerformed**(java.awt.event.ActionEvent evt) {

2359 this.hideAllPanel();

2360 AdminMenuPanel.setVisible(true);

2361 }

2362

2363 private void **adminCustomersListMenuButtonActionPerformed**(java.awt.event.ActionEvent evt) {

2364 this.hideAllPanel();

2365 AdminMenuPanel.setVisible(true);

2366 }

2367

2368 private void **purchaseButtonActionPerformed**(java.awt.event.ActionEvent evt) {

2369 String isbnOfBookPurchased = customerIsbnChoiceField.getText();

2370 Book bookInvolved = DAO.*searchBook*(isbnOfBookPurchased);

2371 PurchaseStatus thisPurchaseStatus = currentCustomer.purchaseBook(bookInvolved);

2372 purchasingStatusLabel.setText("Status : "+ thisPurchaseStatus.value());

2373 Date currentDate = new Date();

2374 Transaction currentTransaction = new Transaction();

2375 currentTransaction.setTransactionId(DAO.*numberOfTransaction*()+1);

2376 currentTransaction.setCustomerId(currentCustomer.getUserId());

2377 currentTransaction.setIsbn(bookInvolved.getIsbn());

2378 currentTransaction.setQuantity(1);

2379 currentTransaction.setDate(currentDate);

2380 if (thisPurchaseStatus== PurchaseStatus.*success*)

2381 {

2382 DAO.*replaceBookData*(bookInvolved);

2383 DAO.*replaceCustomerData*(currentCustomer);

2384 currentTransaction.setPaymentVerification(true);

2385 }

2386 else {

2387 currentTransaction.setPaymentVerification(false);

2388 }

2389 // Add currentTransaction to transaction.xml

2390 DAO.*addTransaction*(currentTransaction);

2391 this.displayCustomerBookList();

2392 }

2393

2394 /\*\*

2395 \* **@param** args the command line arguments

2396 \*/

2397 public static void ***main***(String args[]) {

2398 /\* Set the Nimbus look and feel \*/

2399 //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

2400 /\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

2401 \* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

2402 \*/

2403 try {

2404 for (javax.swing.UIManager.*LookAndFeelInfo* info : javax.swing.UIManager.*getInstalledLookAndFeels*()) {

2405 if ("Nimbus".equals(info.getName())) {

2406 javax.swing.UIManager.*setLookAndFeel*(info.getClassName());

2407 break;

2408 }

2409 }

2410 } catch (ClassNotFoundException ex) {

2411 java.util.logging.Logger.*getLogger*(PicadillyBookstoreGUI.class.getName()).log(java.util.logging.Level.*SEVERE*, null, ex);

2412 } catch (InstantiationException ex) {

2413 java.util.logging.Logger.*getLogger*(PicadillyBookstoreGUI.class.getName()).log(java.util.logging.Level.*SEVERE*, null, ex);

2414 } catch (IllegalAccessException ex) {

2415 java.util.logging.Logger.*getLogger*(PicadillyBookstoreGUI.class.getName()).log(java.util.logging.Level.*SEVERE*, null, ex);

2416 } catch (javax.swing.UnsupportedLookAndFeelException ex) {

2417 java.util.logging.Logger.*getLogger*(PicadillyBookstoreGUI.class.getName()).log(java.util.logging.Level.*SEVERE*, null, ex);

2418 }

2419 //</editor-fold>

2420

2421 /\* Create and display the form \*/

2422 java.awt.EventQueue.*invokeLater*(new Runnable() {

2423 public void **run**() {

2424 new PicadillyBookstoreGUI().setVisible(true);

2425 }

2426 });

2427 }

2428

2429 // Variables declaration - do not modify

2430 private javax.swing.JLabel AccountInfoLabel;

2431 private javax.swing.JPanel AdminAddBookPanel;

2432 private javax.swing.JLabel AdminAddBookPanelSubtitle;

2433 private javax.swing.JLabel AdminAddBookPanelTitle;

2434 private javax.swing.JLabel AdminBooksListISBNLabel;

2435 private javax.swing.JPanel AdminBooksListPanel;

2436 private javax.swing.JLabel AdminBooksListPanelSubtitle;

2437 private javax.swing.JLabel AdminBooksListPanelTitle;

2438 private javax.swing.JPanel AdminCustomersListPanel;

2439 private javax.swing.JLabel AdminCustomersListPanelSubtitle;

2440 private javax.swing.JLabel AdminCustomersListPanelTitle;

2441 private javax.swing.JButton AdminDeleteBookButton;

2442 private javax.swing.JPanel AdminLoginPanel;

2443 private javax.swing.JPanel AdminMenuPanel;

2444 private javax.swing.JButton AdminSubmitBook;

2445 private javax.swing.JPanel AdminTransactionsListPanel;

2446 private javax.swing.JLabel AdminTransactionsListPanelSubtitle;

2447 private javax.swing.JLabel AdminTransactionsListPanelTitle;

2448 private javax.swing.JPanel CustomerEditProfilePanel;

2449 private javax.swing.JLabel CustomerISBNChoiceLabel;

2450 private javax.swing.JPanel CustomerLoginPanel;

2451 private javax.swing.JPanel CustomerMainMenuPanel;

2452 private javax.swing.JLabel CustomerMainMenuPanelSubtitle;

2453 private javax.swing.JLabel CustomerMainMenuPanelTitle;

2454 private javax.swing.JPanel CustomerSignUpPanel;

2455 private javax.swing.JScrollPane CustomerSignUpScrollPane;

2456 private javax.swing.JLabel PersonalProfileLabel;

2457 private javax.swing.JPanel UserChoicePanel;

2458 private javax.swing.JButton addBookButton;

2459 private javax.swing.JTable adminBookTable;

2460 private javax.swing.JButton adminBooksListLogoutButton;

2461 private javax.swing.JButton adminBooksListMenuButton;

2462 private javax.swing.JButton adminButton;

2463 private javax.swing.JTable adminCustomerTable;

2464 private javax.swing.JButton adminCustomersListLogoutButton;

2465 private javax.swing.JButton adminCustomersListMenuButton;

2466 private javax.swing.JButton adminLoginButton;

2467 private javax.swing.JLabel adminLoginPanelSubtitle;

2468 private javax.swing.JLabel adminLoginPanelTitle;

2469 private javax.swing.JButton adminMenuBooksButton;

2470 private javax.swing.JButton adminMenuCustomersButton;

2471 private javax.swing.JLabel adminMenuPanelSubtitle;

2472 private javax.swing.JLabel adminMenuPanelTitle;

2473 private javax.swing.JButton adminMenuTransactionsButton;

2474 private javax.swing.JPasswordField adminPasswordField;

2475 private javax.swing.JLabel adminPasswordLabel;

2476 private javax.swing.JTable adminTransactionTable;

2477 private javax.swing.JButton adminTransactionsListLogoutButton;

2478 private javax.swing.JButton adminTransactionsListMenuButton;

2479 private javax.swing.JTextField adminUsernameField;

2480 private javax.swing.JLabel adminUsernameLabel;

2481 private javax.swing.JTextField authorField;

2482 private javax.swing.JLabel authorLabel;

2483 private javax.swing.JComboBox bindingComboBox;

2484 private javax.swing.JLabel bindingLabel;

2485 private javax.swing.JLabel bookPriceCurrencyLabel;

2486 private javax.swing.JTextField bookPriceField;

2487 private javax.swing.JLabel bookPriceLabel;

2488 private javax.swing.JLabel bookQuantityLabel;

2489 private javax.swing.JTextField bookTitleField;

2490 private javax.swing.JLabel bookTitleLabel;

2491 private javax.swing.ButtonGroup buttonGroup1;

2492 private javax.swing.JTextField categoryField;

2493 private javax.swing.JLabel categoryLabel;

2494 private javax.swing.JLabel creditCardProfileLabel;

2495 private javax.swing.JPasswordField customerAccountPasswordField;

2496 private javax.swing.JLabel customerAccountPasswordLabel;

2497 private javax.swing.JTextField customerAccountUsernameField;

2498 private javax.swing.JLabel customerAccountUsernameLabel;

2499 private javax.swing.JTable customerBookTable;

2500 private javax.swing.JButton customerButton;

2501 private javax.swing.JTextField customerCityField;

2502 private javax.swing.JLabel customerCityLabel;

2503 private javax.swing.JTextField customerCountryField;

2504 private javax.swing.JLabel customerCountryLabel;

2505 private javax.swing.JTextField customerCreditCardMonthField;

2506 private javax.swing.JLabel customerCreditCardMonthLabel;

2507 private javax.swing.JTextField customerCreditCardNameField;

2508 private javax.swing.JLabel customerCreditCardNameLabel;

2509 private javax.swing.JTextField customerCreditCardNumberField;

2510 private javax.swing.JLabel customerCreditCardNumberLabel;

2511 private javax.swing.JTextField customerCreditCardYearField;

2512 private javax.swing.JLabel customerCreditCardYearLabel;

2513 private javax.swing.JTextField customerCvcField;

2514 private javax.swing.JLabel customerCvcLabel;

2515 private javax.swing.JTextField customerEmailField;

2516 private javax.swing.JLabel customerEmailLabel;

2517 private javax.swing.JTextField customerFirstNameField;

2518 private javax.swing.JLabel customerFirstNameLabel;

2519 private javax.swing.JComboBox customerGenderComboBoxes;

2520 private javax.swing.JLabel customerGenderLabel;

2521 private javax.swing.JTextField customerIsbnChoiceField;

2522 private javax.swing.JTextField customerLastNameField;

2523 private javax.swing.JLabel customerLastNameLabel;

2524 private javax.swing.JButton customerLoginButton;

2525 private javax.swing.JLabel customerLoginPanelSubtitle;

2526 private javax.swing.JLabel customerLoginPanelTitle;

2527 private javax.swing.JButton customerMainMenuLogoutButton;

2528 private javax.swing.JPasswordField customerPasswordField;

2529 private javax.swing.JLabel customerPasswordLabel;

2530 private javax.swing.JTextField customerPhoneNumberField;

2531 private javax.swing.JLabel customerPhoneNumberLabel;

2532 private javax.swing.JButton customerSignUpOptionButton;

2533 private javax.swing.JSeparator customerSignUpPanelSeparator1;

2534 private javax.swing.JSeparator customerSignUpPanelSeparator2;

2535 private javax.swing.JLabel customerSignUpPanelTitle;

2536 private javax.swing.JButton customerSignUpSubmitButton;

2537 private javax.swing.JTextField customerStreetField;

2538 private javax.swing.JLabel customerStreetLabel;

2539 private javax.swing.JTextField customerUsernameField;

2540 private javax.swing.JLabel customerUsernameLabel;

2541 private javax.swing.JTextField customerZipCodeField;

2542 private javax.swing.JLabel customerZipCodeLabel;

2543 private javax.swing.JLabel descriptionCharactersLimitLabel;

2544 private javax.swing.JTextField descriptionField;

2545 private javax.swing.JLabel descriptionLabel;

2546 private javax.swing.Box.*Filler* filler1;

2547 private javax.swing.JTextField isbnField;

2548 private javax.swing.JLabel isbnLabel;

2549 private javax.swing.JTextField isbnToDeleteField;

2550 private javax.swing.JButton jButton32;

2551 private javax.swing.JCheckBox jCheckBox3;

2552 private javax.swing.JCheckBox jCheckBox4;

2553 private javax.swing.JLabel jLabel51;

2554 private javax.swing.JLabel jLabel52;

2555 private javax.swing.JLabel jLabel53;

2556 private javax.swing.JLabel jLabel54;

2557 private javax.swing.JLabel jLabel55;

2558 private javax.swing.JLabel jLabel56;

2559 private javax.swing.JLabel jLabel57;

2560 private javax.swing.JLabel jLabel58;

2561 private javax.swing.JLabel jLabel59;

2562 private javax.swing.JLabel jLabel60;

2563 private javax.swing.JLabel jLabel61;

2564 private javax.swing.JLabel jLabel62;

2565 private javax.swing.JLabel jLabel63;

2566 private javax.swing.JLabel jLabel64;

2567 private javax.swing.JLabel jLabel65;

2568 private javax.swing.JLabel jLabel66;

2569 private javax.swing.JLabel jLabel67;

2570 private javax.swing.JLabel jLabel68;

2571 private javax.swing.JLabel jLabel69;

2572 private javax.swing.JLabel jLabel70;

2573 private javax.swing.JLabel jLabel71;

2574 private javax.swing.JLabel jLabel93;

2575 private javax.swing.JScrollPane jScrollPane1;

2576 private javax.swing.JScrollPane jScrollPane2;

2577 private javax.swing.JScrollPane jScrollPane3;

2578 private javax.swing.JScrollPane jScrollPane4;

2579 private javax.swing.JSeparator jSeparator3;

2580 private javax.swing.JSeparator jSeparator4;

2581 private javax.swing.JTextField jTextField26;

2582 private javax.swing.JTextField jTextField27;

2583 private javax.swing.JTextField jTextField28;

2584 private javax.swing.JTextField jTextField29;

2585 private javax.swing.JTextField jTextField30;

2586 private javax.swing.JTextField jTextField31;

2587 private javax.swing.JTextField jTextField32;

2588 private javax.swing.JTextField jTextField33;

2589 private javax.swing.JTextField jTextField34;

2590 private javax.swing.JTextField jTextField35;

2591 private javax.swing.JTextField jTextField36;

2592 private javax.swing.JTextField jTextField37;

2593 private javax.swing.JTextField jTextField38;

2594 private javax.swing.JTextField jTextField39;

2595 private javax.swing.JTextField jTextField40;

2596 private javax.swing.JTextField jTextField43;

2597 private javax.swing.JTextField keywordsField;

2598 private javax.swing.JLabel keywordsLabel;

2599 private javax.swing.JLabel publishedNameLabel;

2600 private javax.swing.JTextField publishedYearField;

2601 private javax.swing.JLabel publishedYearLabel;

2602 private javax.swing.JTextField publisherNameField;

2603 private javax.swing.JButton purchaseButton;

2604 private javax.swing.JLabel purchasingStatusLabel;

2605 private javax.swing.JTextField quantityField;

2606 private javax.swing.JComboBox signatureComboBox;

2607 private javax.swing.JLabel signatureLabel;

2608 private javax.swing.JLabel userChoicePanelTitle;

2609 // End of variables declaration

2610 }

2611

Code in PicadillyBookStoreStepThree.java (inside picadillybookstore.step3 package):  
 1 package picadillybookstore.step3;

2

3 import java.util.ArrayList;

4 import java.util.List;

5 import java.util.Arrays;

6 import java.util.Date;

7

8 import picadillybookstore.domain.Admin;

9 import picadillybookstore.domain.Binding;

10 import picadillybookstore.domain.Book;

11 import picadillybookstore.domain.Customer;

12 import picadillybookstore.domain.Gender;

13 import picadillybookstore.domain.Signed;

14 import picadillybookstore.domain.Transaction;

15 import picadillybookstore.domain.PurchaseStatus;

16

17 /\*\*

18 \* **PicadillyBookStoreStepThree.java**

19 \* **This** **class** **is** **the** **container** **of** **main** **function, before** **using** **DAO**

20 \*

21 \* **@author** Albert Darmawan

22 \* **@version** 1.0

23 \* **@since** 2015-05-26

24 \*/

25 public class PicadillyBookStoreStepThree {

26 //------------

27 // Array List Initialization

28 //------------

29 public static ArrayList<Book> bookCatalog;

30 public static ArrayList<Customer> customerList;

31 public static ArrayList<Admin> adminList;

32 public static ArrayList<Transaction> transactionHistory;

33 public static int numberOfTransaction = 0;

34

35 /\*\*

36 \*

37 \* **@param** args

38 \*/

39 //------------

40 // Main function

41 //------------

42 public static void main(String[] args) {

43 // Object variable declaration

44 Book b1, b2, b3;

45 Customer c1, c2;

46 Admin a1, a2;

47

48 // Defining 3 books

49 b1 = new Book("George Berkowski", "How to Build A Billion Dollar App", "978-0-349-40137-9",(long) 246221, (long) 0, "Little, Brown Book Group",2015, "An accessible, step-by-step guide to building an app-based business—essential reading for anyone who has an idea for an app, but is unsure of where to start", Binding.PAPERBACK,Signed.NOTSIGNED, Arrays.asList("app", "business", "startup", "technology", "mobile"), Arrays.asList("business"));

50 b2 = new Book("Jean Hugard & Frederick Braue", "The Royal Road to Card Magic", "978-1-936759-01-9", (long) 134477, (long) 7, "Harper & Brothers Publishers", 2011, "With the help of this book, anyone can develop a versatile repertoire of first-rate card tricks. In fact, mastery of just the first chapter will enable you to perform a half-dozen astounding and entertaining sleights of hand. This exciting introduction to card conjuring will enable even beginners to develop professional-level skill and the ability to perform tricks guaranteed to astound family and friends.", Binding.PAPERBACK, Signed.NOTSIGNED, Arrays.asList("card", "magic", "houdini", "entertainment"), Arrays.asList("entertainment"));

51 b3 = new Book("William Poundstone", "Are You Smart Enough to Work at Google?", "978-0-316-09997-4", (long) 171770, (long) 10, "Little, Brown and Company", 2012, "Trick Questions, Zen-like Riddles, Insanely Difficult Puzzles, and Other Devious Interviewing Techniques You Need to Know to Get a Job Anywhere in the New Economy", Binding.PAPERBACK, Signed.NOTSIGNED, Arrays.asList("google", "smart", "intelligence", "work"), Arrays.asList("career"));

52

53 // Adding 3 books to Book array list

54 bookCatalog = new ArrayList<>();

55 bookCatalog.add(b1);

56 bookCatalog.add(b2);

57 bookCatalog.add(b3);

58

59 // Defining 2 customers

60 c1 = new Customer((long) 1, "Albert Darmawan", Gender.MALE, "darmawanalbert", "123456", "darmawanalbert@gmail.com", "081281123450", 0, "Jalan Citandui 1 Taman Tirta Golf", "Tangerang Selatan", "15310", "Indonesia", "Albert Darmawan", "1010202010101010", "145", "05", "18");

61 c2 = new Customer((long) 2, "Matthew Kharli", Gender.MALE, "matthewk", "098765", "matthewk@gmail.com", "08119001771", 0, "715 Slater Avenue", "Pleasant Hill", "94523", "USA", "Matthew Kharli", "1010001011110100", "010", "10", "10");

62

63 // Adding 2 customers to Customer array list

64 customerList = new ArrayList<>();

65 customerList.add(c1);

66 customerList.add(c2);

67

68 // Defining 2 admins

69 a1 = new Admin((long) 3, "Tri Asih Budiono", Gender.MALE, "tabudi", "ouch", "tabudi@gmail.com", "085632456732");

70 a2 = new Admin((long) 4, "Elbert Christopher Larosa", Gender.MALE, "elbert", "bejo", "elbertcl@yahoo.com", "0818833551");

71

72 // Adding 2 admins to Admin array list

73 adminList = new ArrayList<>();

74 adminList.add(a1);

75 adminList.add(a2);

76

77 // Creating Transaction array List

78 transactionHistory = new ArrayList<>();

79

80 // Output

81 System.out.println("================================================================");

82 System.out.println(" Welcome to Picadilly Book Store! **\n\t**Where customer's satisfaction is our main priority");

83 System.out.println("================================================================");

84

85 // Displaying Books List

86 System.out.println("================================================================");

87 System.out.println(" Book List");

88 System.out.println("================================================================");

89

90 for (Book eachBook : bookCatalog) {

91 eachBook.display();

92 }

93

94 // Displaying Customer List

95 System.out.println("================================================================");

96 System.out.println(" Customer List");

97 System.out.println("================================================================");

98

99 for (Customer eachCustomer : customerList) {

100 eachCustomer.display();

101 }

102

103 // Displaying Admin List

104 System.out.println("================================================================");

105 System.out.println(" Admin List");

106 System.out.println("================================================================");

107

108 for (Admin eachAdmin : adminList) {

109 eachAdmin.display();

110 }

111

112 // Customers start to purchase books

113 System.out.println("================================================================");

114 System.out.println(" Book Purchasing");

115 System.out.println("================================================================");

116 attemptToPurchaseBook(c1,b1);

117 attemptToPurchaseBook(c2,b2);

118

119 // Transaction History

120 System.out.println("================================================================");

121 System.out.println(" Transaction History");

122 System.out.println("================================================================");

123

124 for (Transaction eachTransaction : transactionHistory) {

125 eachTransaction.display();

126 }

127 }

128

129 private static void attemptToPurchaseBook(Customer customerInvolved, Book bookInvolved)

130 {

131 Transaction currentTransaction;

132 Date currentDate;

133

134 System.out.println(customerInvolved.getName()+" is trying to buy **\"**" + bookInvolved.getTitle() + "**\"**");

135 PurchaseStatus thisPurchaseStatus = customerInvolved.purchaseBook(bookInvolved);

136 System.out.println(thisPurchaseStatus.value());

137 numberOfTransaction++;

138 currentDate = new Date();

139 if (thisPurchaseStatus== PurchaseStatus.success)

140 {

141 currentTransaction = new Transaction(numberOfTransaction,customerInvolved.getUserId(), bookInvolved.getIsbn(), 1,currentDate,true);

142 System.out.println("Remaining books titled **\"**"+bookInvolved.getTitle()+"**\"**" + " : " + bookInvolved.getQuantity());

143 }

144 else {

145 currentTransaction = new Transaction(numberOfTransaction,customerInvolved.getUserId(), bookInvolved.getIsbn(), 1,currentDate,false);

146 }

147 transactionHistory.add(currentTransaction);

148 System.out.println("");

149 }

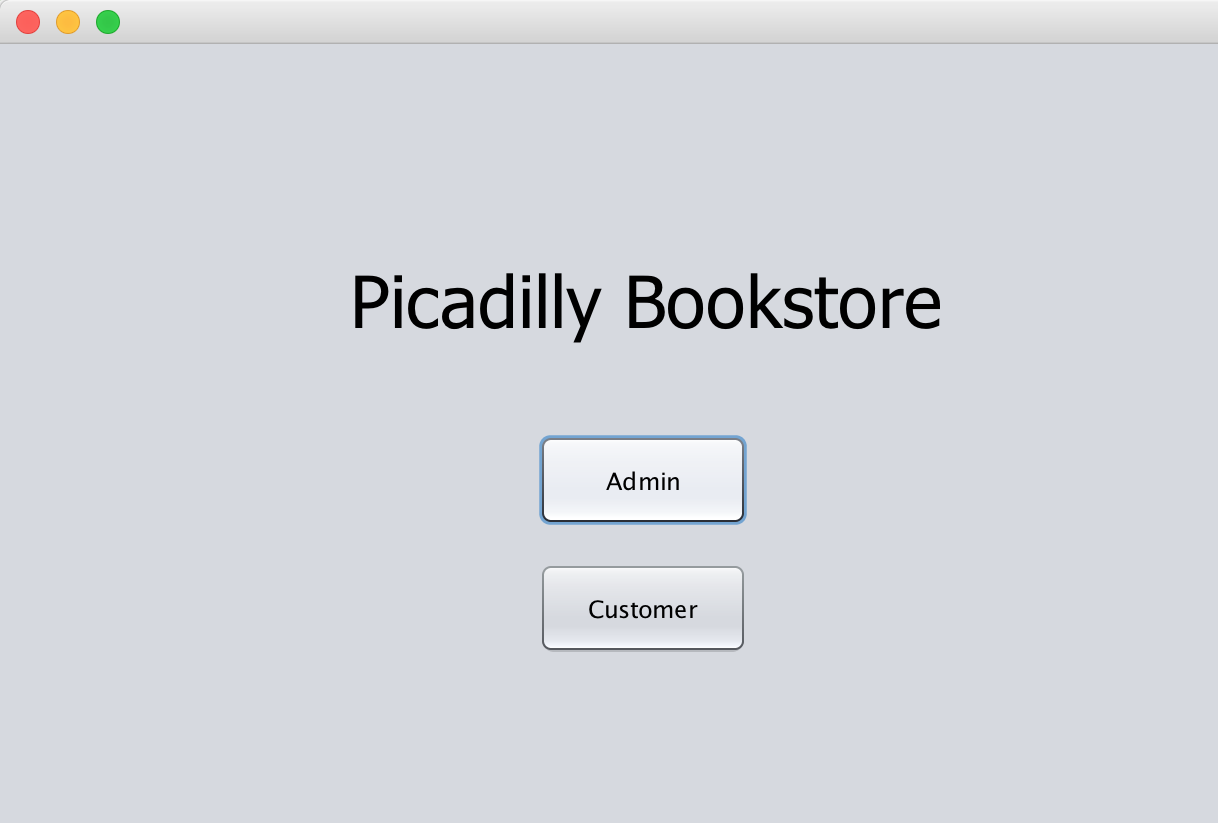
150 }

151

**Testing**

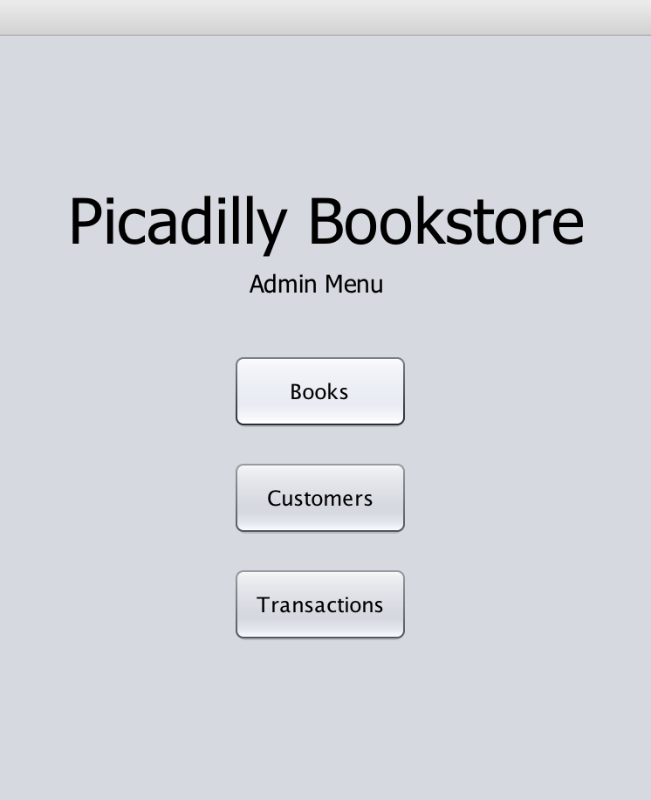
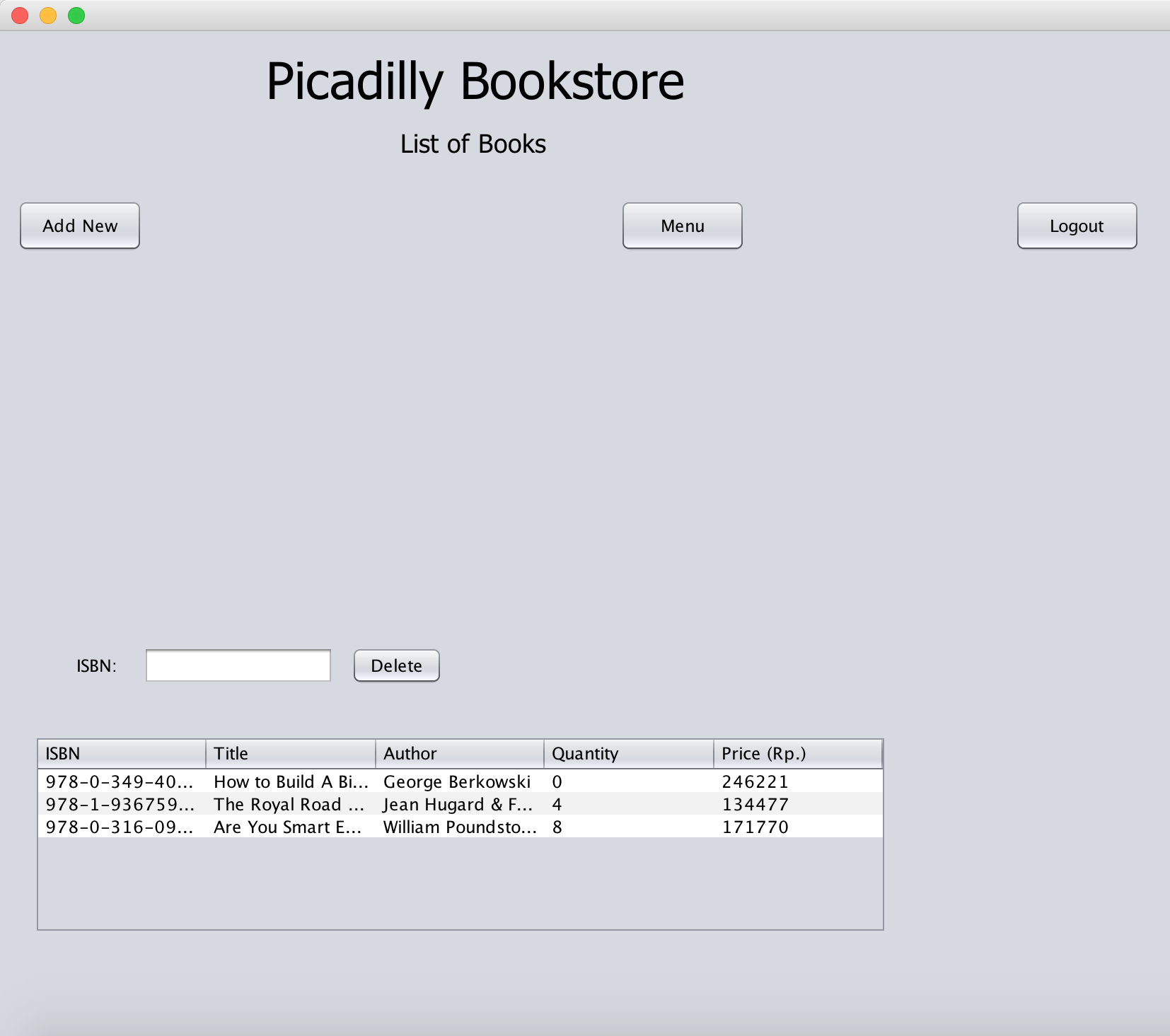
The testing strategy is divided into two, the first one is by inputting in the program correctly and by using the program the way it should be. The second one is by giving bad inputs and by using the program the way it should not be. Both strategies are going to be applied to both admin and customer.

 The results of the testing by using the first strategy as an Admin are shown below:



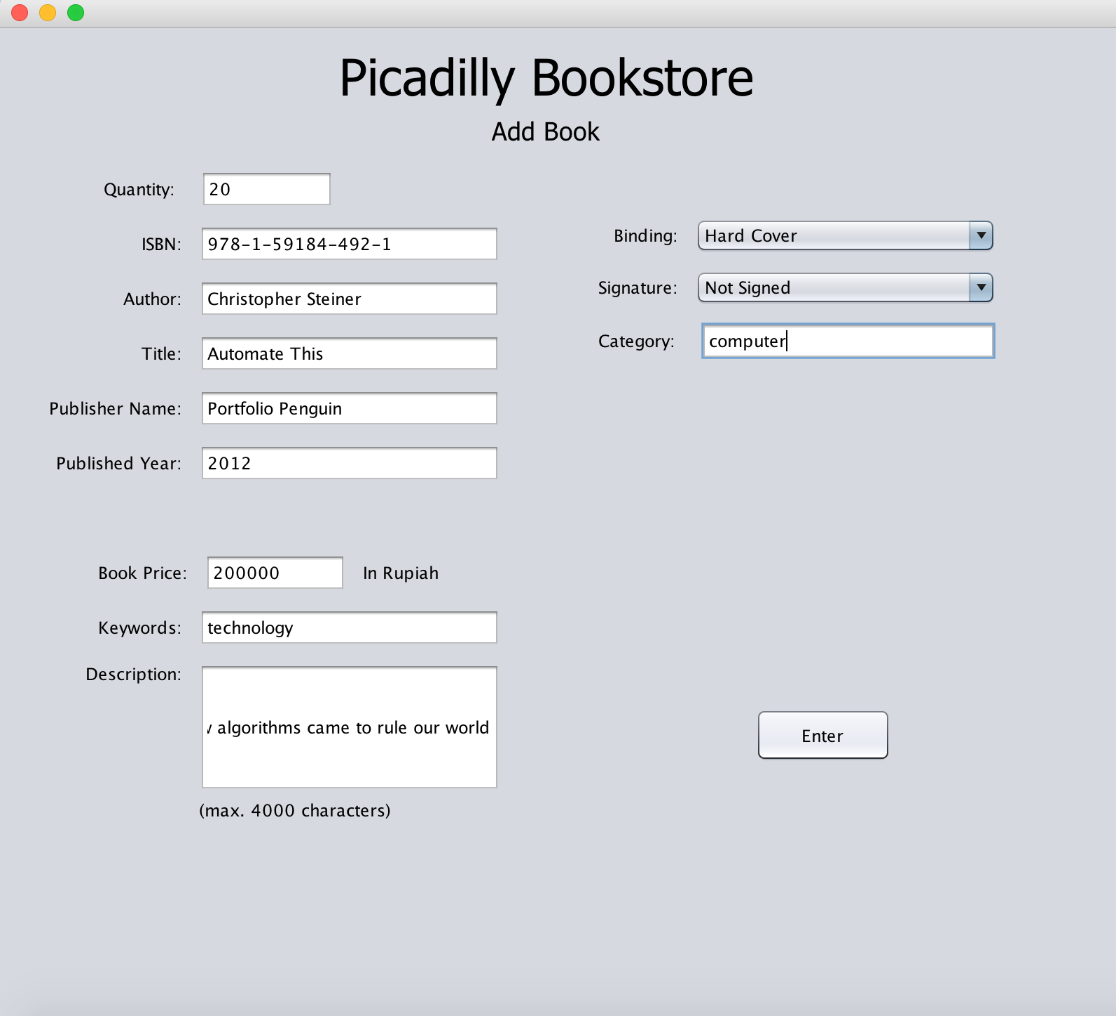
The Admin login worked as intended when the existing username and password as in the XML file are inputted

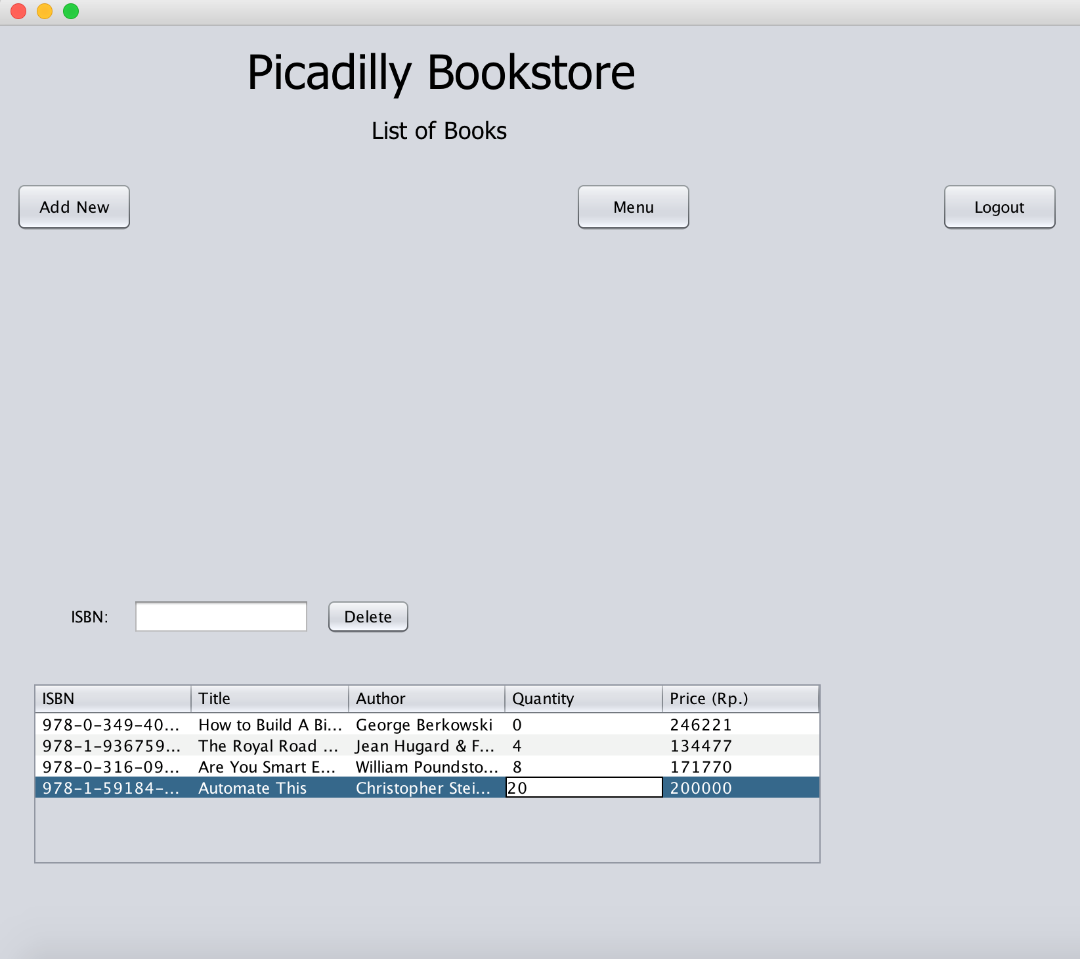
The User Choice panel successfully redirects to the Admin Login panel when choosing as an Admin



The Admin’s List of Books panel worked successfully in viewing books that have already been added

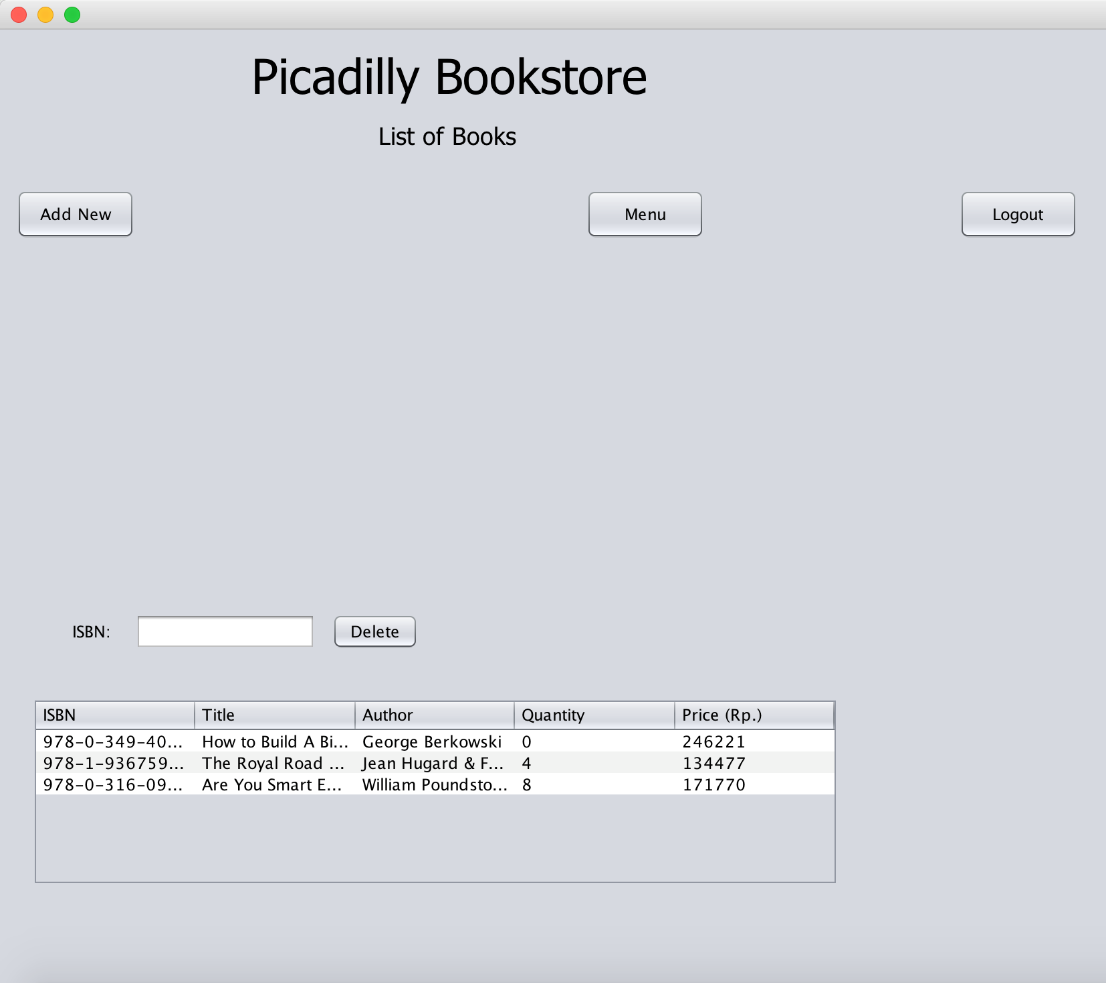
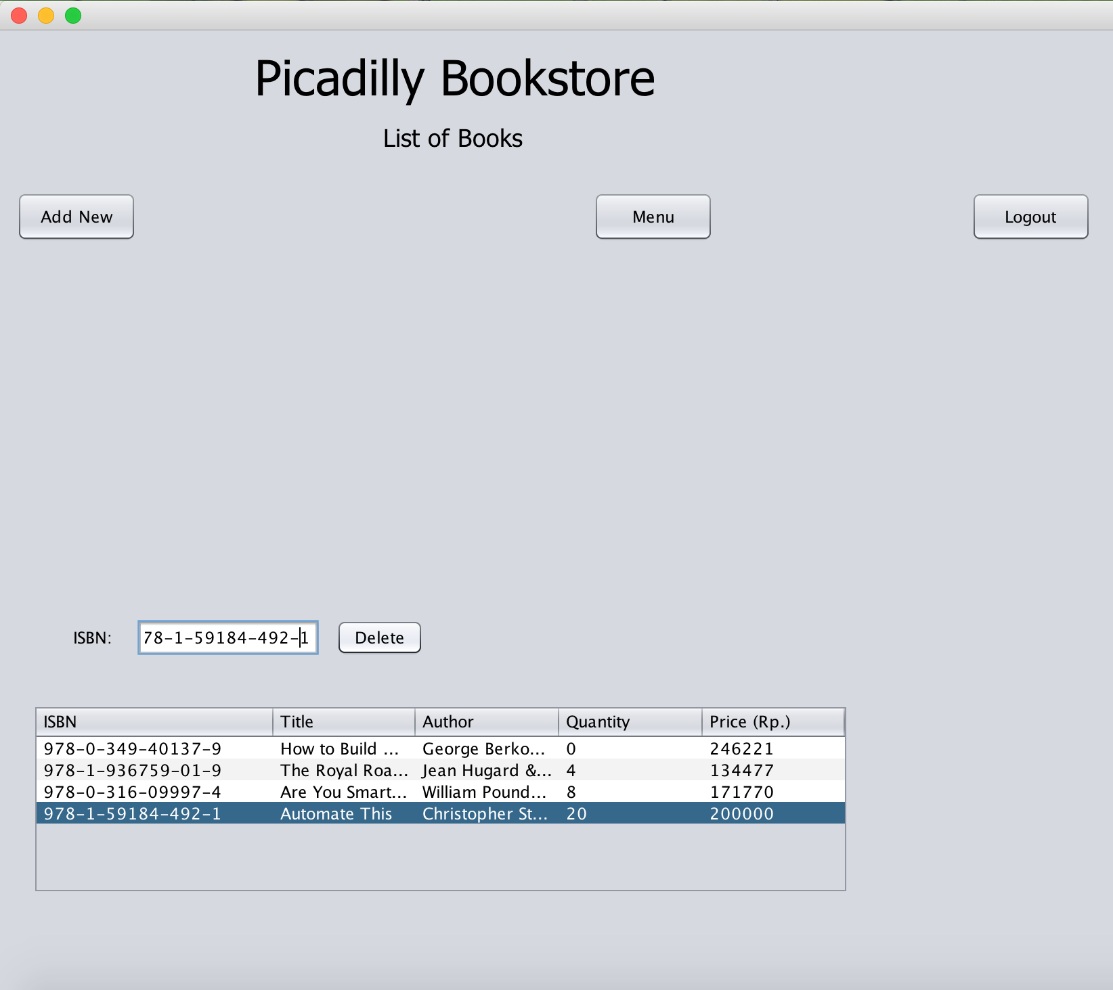
The Admin Menu panel worked as intended in redirecting to the Books list, Customers list, or the Transactions List





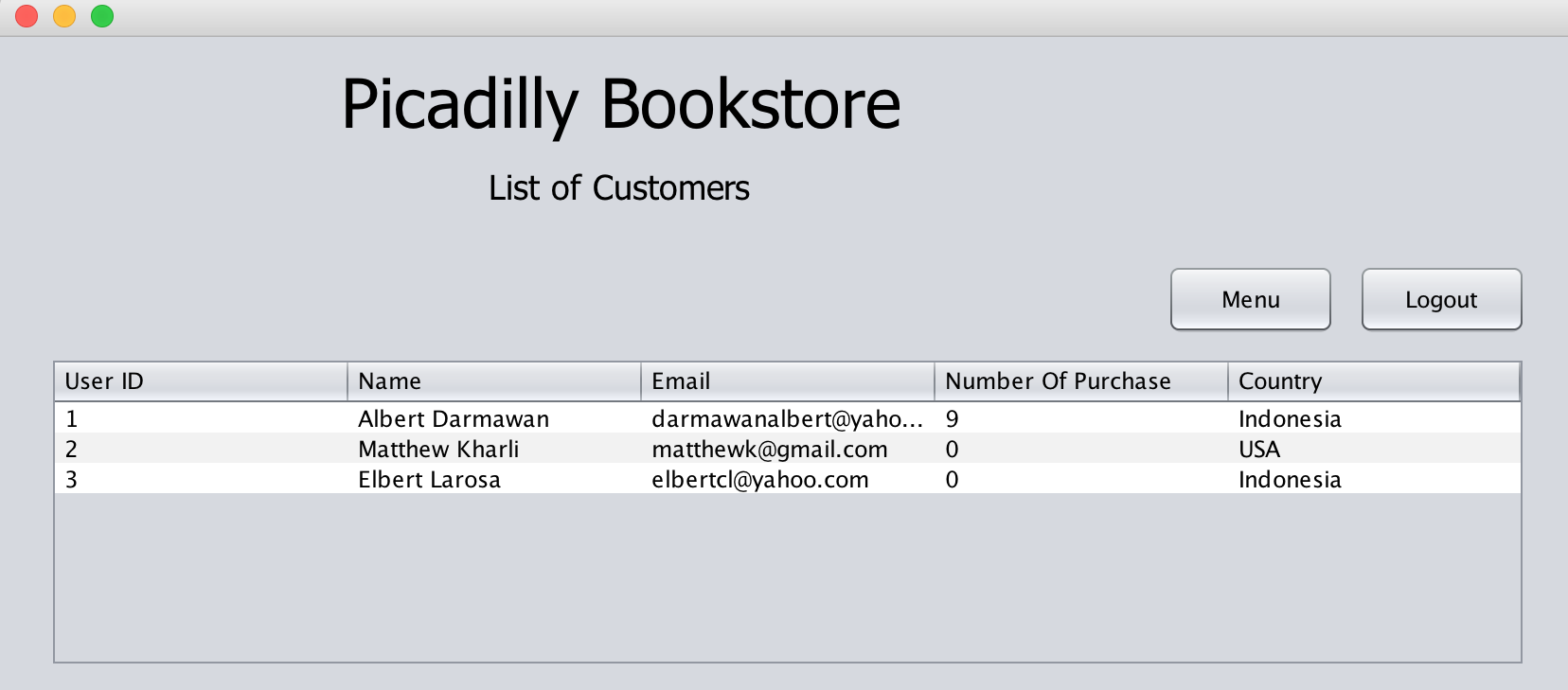
The previously added book is able to be viewed as intended in the List of Books panel

The Admin’s Add Book feature worked successfully in adding new book

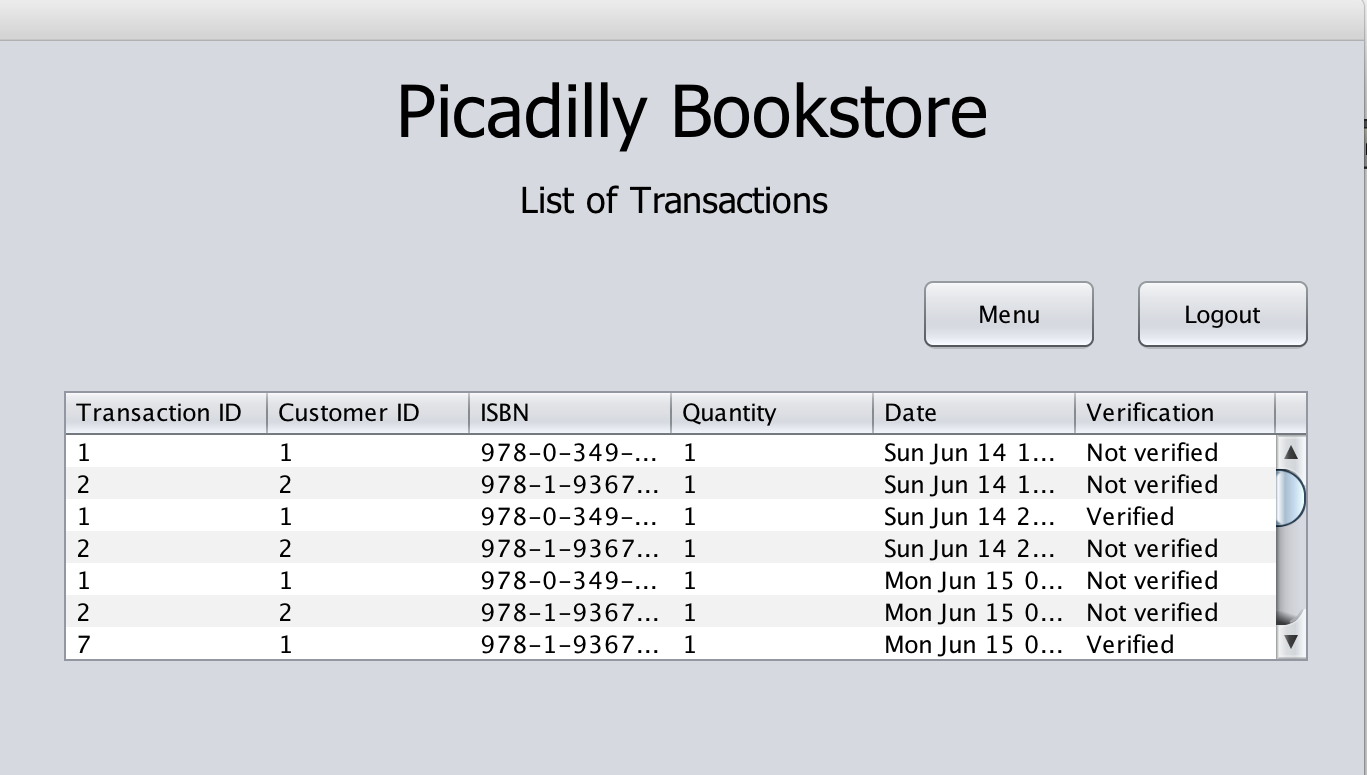


The delete book feature for Admin worked successfully when the correct ISBN is inputted

The previously deleted book is successfully removed from the List of Books

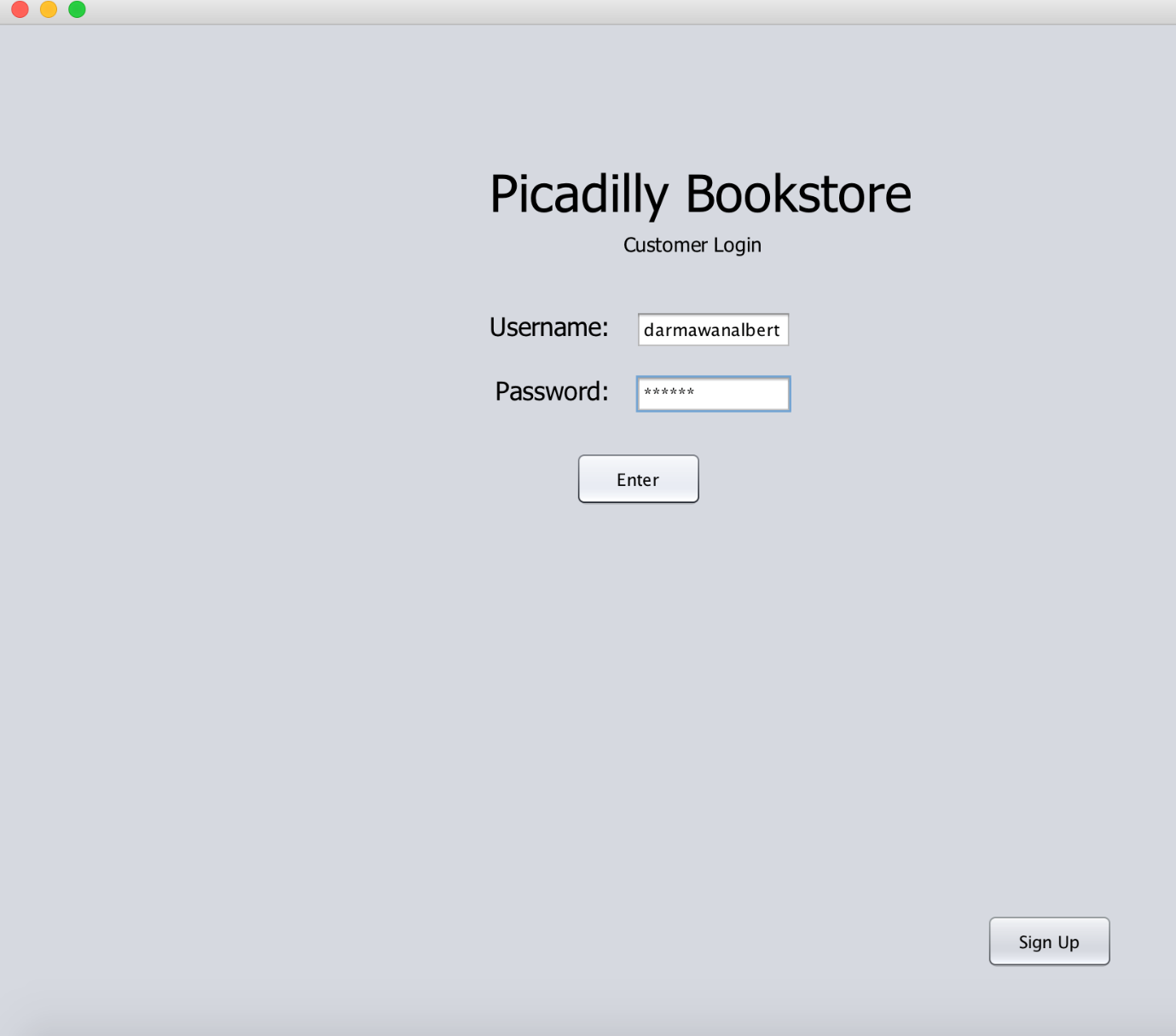


The List of Customers panel successfully shows the existing customers information

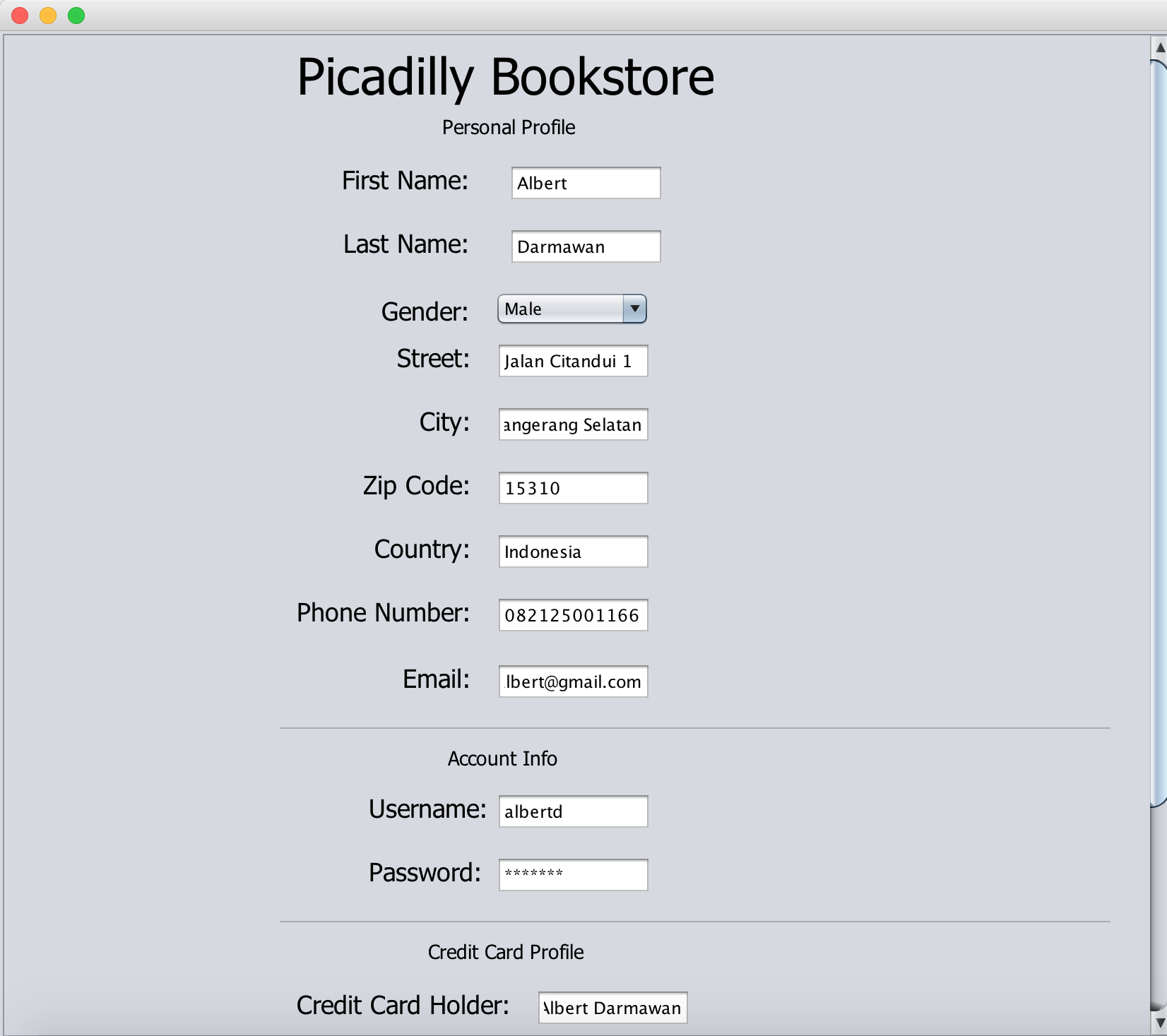
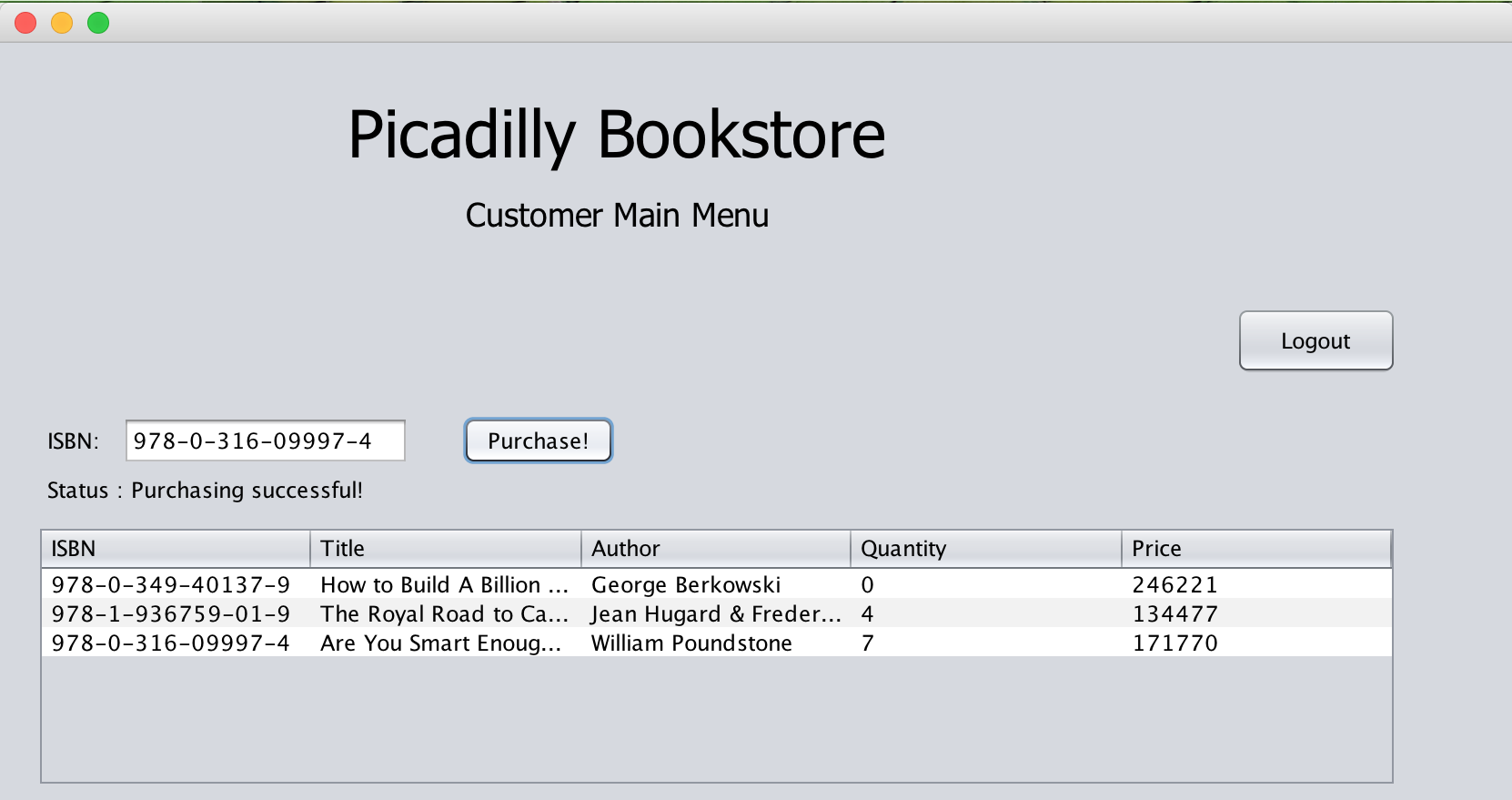


The List of Transactions panel successfully shows the transactions that have occurred previously

When the program is tested as an Admin by giving bad inputs and using the program the way it should not be (the second strategy), the program gave no changes. The program has been made to set the text field to blank when the incorrect input is given. This is why when incorrect input is given, the program gave no changes except the text field set to blank. When wrong inputs (such as wrong data type) is inputted when adding a new book, the bad inputs are still being stored in the database. This occurred because there is no input verification in that feature yet.

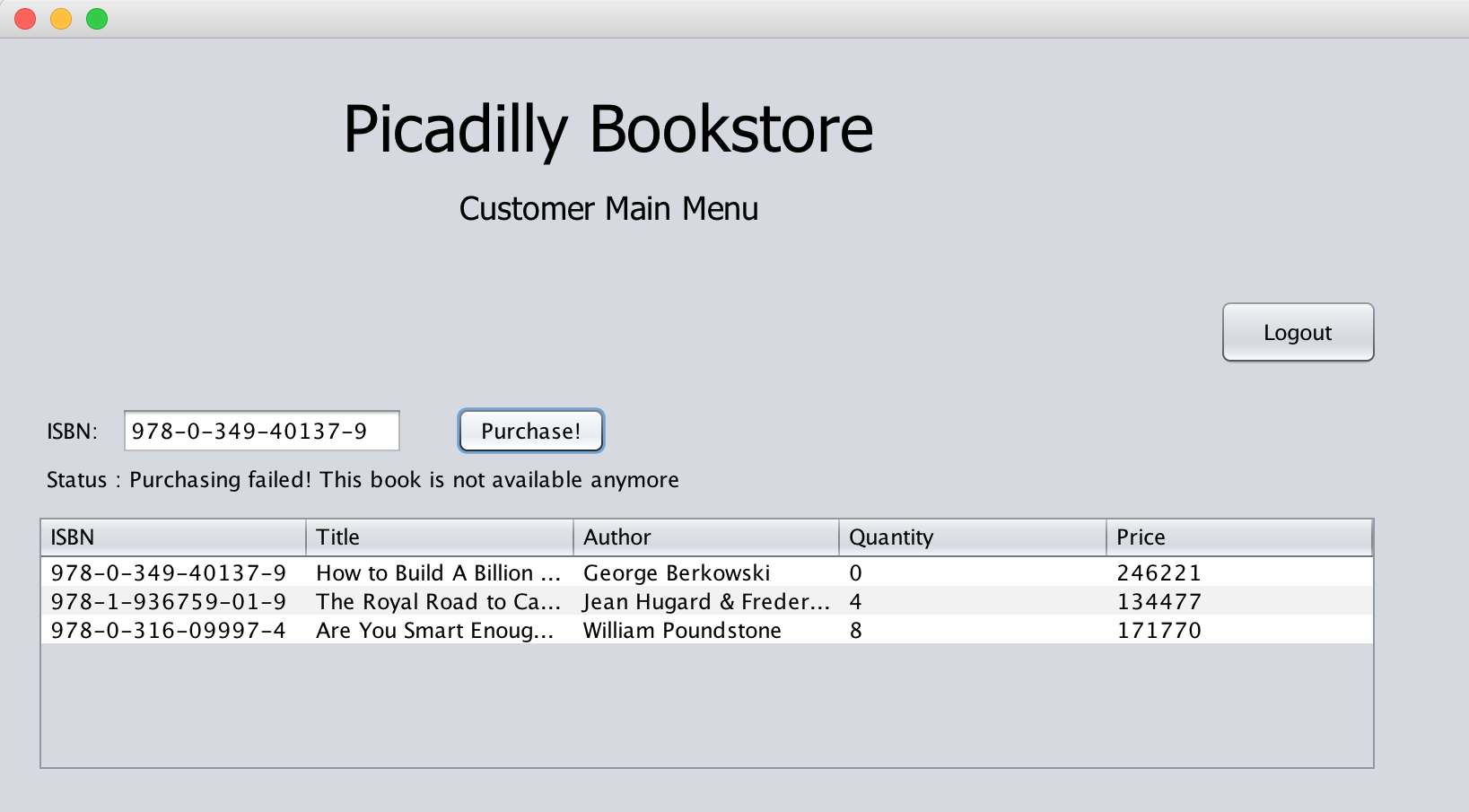
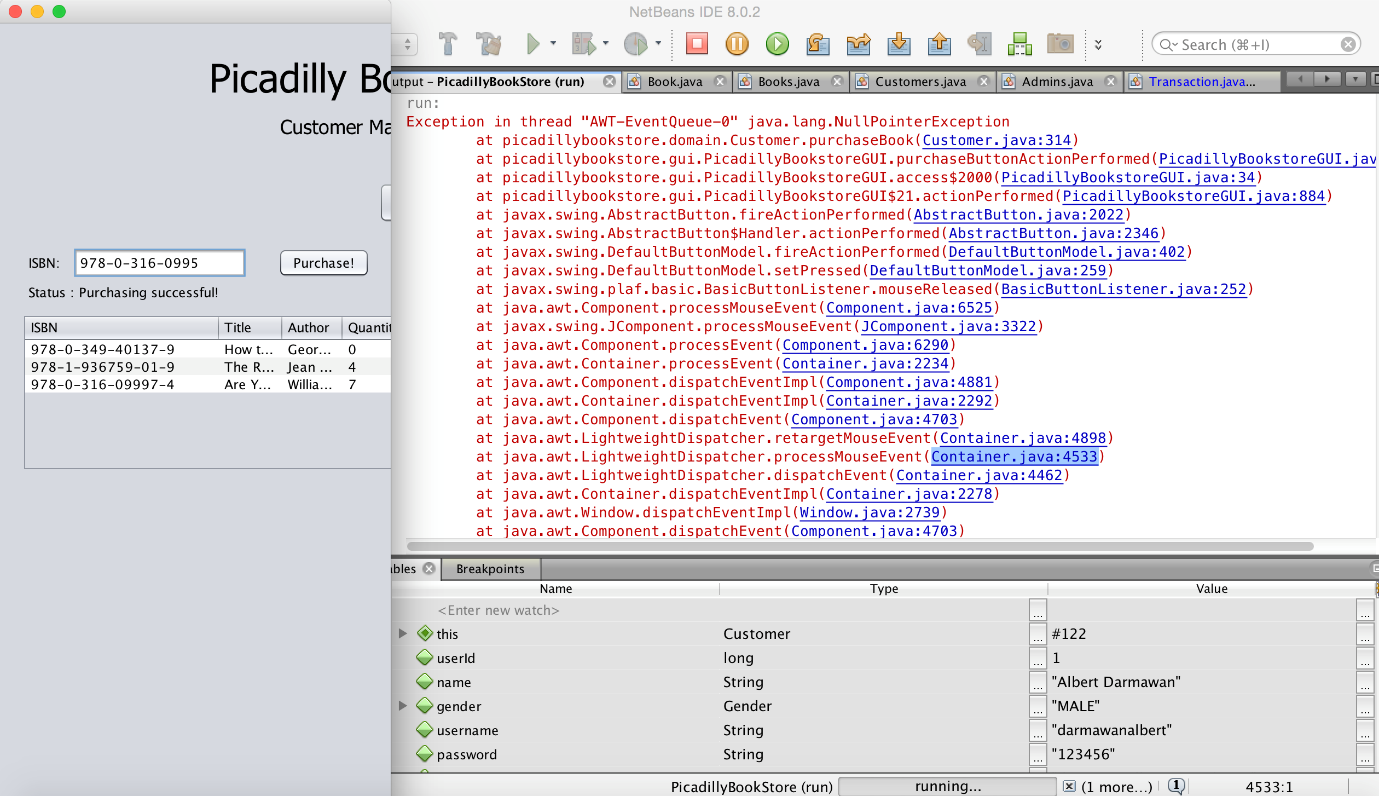
 The results of the testing by using the first strategy as customer are shown below:

The customer’s login feature worked as intended when the existing username and password in the XML file is inputted



The customer’s purchase book feature worked successfully in purchasing available book and keeping the transaction record into the XML file

The customer’s sign up feature worked successfully in adding new customer profile

The results of the testing by using the second strategy as customer are shown below:

A NullPointerException occurred when the incorrect ISBN is inputted in the purchase book feature

The purchase book feature worked successfully in preventing customer from buying book at are not available