CENTENNIAL COLLEGE COMP 228: JAVA PROGRAMMING

FINAL PROJECT

NAME: JOKKU SABU KATTAKKAYAM

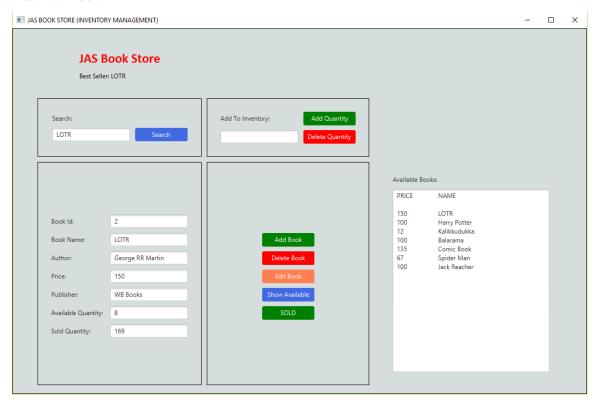
STUDENT ID NO.: 301013603

Screenshots

Login



Main Window



Code Window

```
import javax.swing.JOptionPane;
import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.GridPane;
import javafx.stage.Stage;
import javafx.scene.text.Text;
import javafx.stage.StageStyle;
//For Database
import java.sql.*;
public class ApplicationGUI extends Application
   Text bestSellerText = new Text("Best Seller: ");
   Label title= new Label("JAS Book Store");
   //For Login Window
   //creating label email
   Label emailLbl= new Label("Email");
   //creating label password
   Label passwordLbl= new Label("Password");
   //Creating Text Filed for email
   TextField email = new TextField();
```

```
//Creating Text Filed for password
  PasswordField password = new PasswordField();
  //Creating Buttons
  Button b1 = new Button("Login");
  Button b2 = new Button("Cancel");
  //For Main Window
  Label bookIdLbl = new Label ("Book Id: ");
  TextField bookIdTextField = new TextField();
  Label bookNameLbl = new Label ("Book Name: ");
  TextField bookNameTextField = new TextField();
  Label bookAuthorLbl = new Label ("Author: ");
  TextField bookAuthorTextField = new TextField();
  Label bookPriceLbl = new Label ("Price: ");
  TextField bookPriceTextField = new TextField();
  Label bookPublisherLbl = new Label("Publisher: ");
  TextField bookPublisherTextField = new TextField();
  Label quantityAvailableLbl = new Label("Available Quantity: ");
  TextField quantityAvailableTextField = new TextField();
  Label quantitySoldLbl = new Label("Sold Quantity: ");
  TextField quantitySoldTextField = new TextField();
  Label addToInventoryLbl = new Label("Add To Inventory: ");
  TextField addToInventoryTextField = new TextField();
  Label searchFromInventoryLbl = new Label("Search: ");
  TextField searchFromInventoryTextField = new TextField();
  Label availableBooksLbl = new Label("Available Books: ");
  TextArea availableBooksTextArea = new TextArea();
//Label TemporaryLbl = new Label("THIS IS A TEMPORARY LABEL.");
  Button searchBtn = new Button("Search");
  Button createBtn = new Button("Add Book");
  Button deleteBtn = new Button("Delete Book");
  Button editBtn = new Button("Edit Book");
  Button saveBtn = new Button("Save Book info");
  Button viewAllBtn = new Button("View Existing Books");
  Button addQuantityBtn = new Button("Add Quantity");
  Button removeQuantityBtn = new Button("Delete Quantity");
  Button showAvailableBooksBtn = new Button("Show Available");
  Button soldBtn = new Button("SOLD");
  Stage window;
                     //Taking instance of a stage as Global stage variable.
```

```
@Override
  public void start(Stage primaryStage) throws Exception
     Stage popupwindow=new Stage();
     popupwindow.setTitle("Login window");
     b1.setOnAction(e -> LoginBtnClicked(popupwindow));
     b2.setOnAction(e -> System.exit(0));
     //Creating a Grid Pane
     GridPane gridPane = new GridPane();
     //Setting size for the pane
     gridPane.setMinSize(400, 300);
     //Setting the padding
     gridPane.setPadding(new Insets(10, 10, 10, 10));
     //Setting the vertical and horizontal gaps between the columns
     gridPane.setVgap(20);
     gridPane.setHgap(10);
     //Setting the Grid alignment
     gridPane.setAlignment(Pos.CENTER);
     //Arranging all the nodes in the grid
     gridPane.add(emailLbl, 0, 1);
     gridPane.add(email, 1, 1);
     gridPane.add(passwordLbl, 0, 2);
     gridPane.add(password, 1, 2);
     gridPane.add(b1, 0, 3);
     gridPane.add(b2, 1, 3);
     gridPane.add(title, 0, 0);
      //Styling nodes
     b1.setStyle("-fx-background-color: green; -fx-text-fill: white;");
     b2.setStyle("-fx-background-color: red; -fx-text-fill: white;");
     emailLbl.setStyle(" -fx-text-fill: white; -fx-font: normal bold 15px
'serif'");
     passwordLbl.setStyle(" -fx-text-fill: white; -fx-font: normal bold 15px
'serif' ");
     title.setStyle(" -fx-text-fill: gold; -fx-font: normal bold 25px
'calibri' ");
     gridPane.setStyle("-fx-background-color: darkslategrey;");
     // Creating a scene object
     Scene scene1 = new Scene(gridPane);
     popupwindow.setScene(scene1);
     popupwindow.initStyle(StageStyle.UNDECORATED);
     popupwindow.showAndWait();
     popupwindow.setOnCloseRequest(e->System.exit(0));
     popupwindow.close();
```

```
//STAGE: Declaration and Configuration of a stage. Start >>>
     window = primaryStage;
     window.setTitle("JAS BOOK STORE (INVENTORY MANAGEMENT)");
     //STAGE: Declaration and Configuration of a stage. End <<<
     //GridPane: Declaration and Configuration of rootGridPane. Start
>>>>>>>>>>>>>>>
     GridPane rootGridPane = new GridPane();
     rootGridPane.setAlignment(Pos.CENTER);
     rootGridPane.setStyle("-fx-background-color: #d5dddd; -fx-border-style:
solid inside; -fx-border-width: 1 ");
     rootGridPane.setHgap(10);
     rootGridPane.setVgap(10);
     //GridPane: Declaration and Configuration of rootGridPane. End
<<<<<<<<
     //GridPane: Declaration and Configuration of firstGridPane. Start
>>>>>>>>>>>>
     GridPane firstGridPane = new GridPane();
     firstGridPane.setAlignment(Pos.CENTER);
     firstGridPane.setStyle("-fx-background-color: #d5dddd; -fx-border-style:
solid inside; -fx-border-width: 1 ");
     firstGridPane.setHgap(10);
     firstGridPane.setVgap(10);
     firstGridPane.setPadding(new Insets(25, 25, 25, 25));
     firstGridPane.add(bookIdLbl, 0, 1);
     firstGridPane.add(bookIdTextField, 1, 1);
     firstGridPane.add(bookNameLbl, 0, 2);
     firstGridPane.add(bookNameTextField, 1, 2);
     firstGridPane.add(bookAuthorLbl, 0, 3);
     firstGridPane.add(bookAuthorTextField, 1, 3);
     firstGridPane.add(bookPriceLbl, 0, 4);
     firstGridPane.add(bookPriceTextField, 1, 4);
     firstGridPane.add(bookPublisherLbl, 0, 5);
     firstGridPane.add(bookPublisherTextField, 1, 5);
     firstGridPane.add(quantityAvailableLbl, 0, 6);
     firstGridPane.add(quantityAvailableTextField, 1, 6);
     firstGridPane.add(quantitySoldLbl, 0, 7);
     firstGridPane.add(quantitySoldTextField, 1, 7);
     //GridPane: Declaration and Configuration of firstGridPane. End
<<<<<<<<
     //GridPane: Declaration and Configuration of secondGridPane. Start
>>>>>>>>>>
     GridPane secondGridPane = new GridPane();
     secondGridPane.setAlignment(Pos.CENTER);
     secondGridPane.setStyle("-fx-background-color: #d5dddd; -fx-border-style:
solid inside; -fx-border-width: 1");
     secondGridPane.setHgap(10);
     secondGridPane.setVgap(10);
```

```
secondGridPane.setPadding(new Insets(25, 25, 25, 25));
     secondGridPane.add(addToInventoryLbl, 0, 0);
     secondGridPane.add(addToInventoryTextField, 0, 1);
     secondGridPane.add(addQuantityBtn, 1, 0);
     secondGridPane.add(removeQuantityBtn, 1, 1);
     //GridPane: Declaration and Configuration of secondGridPane. End
<<<<<<<<<<></></></></></></></></
     //GridPane: Declaration and Configuration of thirdGridPane. Start
>>>>>>>>>>
     GridPane thirdGridPane = new GridPane();
     thirdGridPane.setAlignment(Pos. CENTER);
     thirdGridPane.setStyle("-fx-background-color: #d5dddd; -fx-border-style:
solid inside; -fx-border-width: 1");
     thirdGridPane.setHgap(10);
     thirdGridPane.setVgap(10);
     thirdGridPane.setPadding(new Insets(25, 25, 25, 25));
     thirdGridPane.add(searchFromInventoryLbl, 0, 0);
     thirdGridPane.add(searchFromInventoryTextField, 0, 1);
     thirdGridPane.add(searchBtn, 1, 1);
     //GridPane: Declaration and Configuration of thirdGridPane. End
<<<<<<<<
     //GridPane: Declaration and Configuration of fourthGridPane. Start
GridPane fourthGridPane = new GridPane();
     fourthGridPane.setAlignment(Pos.CENTER);
     fourthGridPane.setStyle("-fx-background-color: #d5dddd; -fx-border-style:
solid inside; -fx-border-width: 1");
     fourthGridPane.setHgap(10);
     fourthGridPane.setVgap(10);
     fourthGridPane.setPadding(new Insets(25, 25, 25, 25));
     //>>>>
     //GridPane: Declaration and Configuration of titleGridPane. Start
>>>>>>>>>>>
     GridPane titleGridPane = new GridPane();
     titleGridPane.setAlignment(Pos. CENTER);
     titleGridPane.setStyle("-fx-background-color: #d5dddd;");
     title.setStyle(" -fx-text-fill: red; -fx-font: normal bold 25px
'calibri' ");
     titleGridPane.setHgap(10);
     titleGridPane.setVgap(10);
     titleGridPane.setPadding(new Insets(25, 25, 25, 25));
     //>>>>
     //GridPane: Declaration and Configuration of textAreaGrid. Start
     GridPane textAreaGrid = new GridPane();
     textAreaGrid.setAlignment(Pos. CENTER);
     textAreaGrid.setStyle("-fx-background-color: #d5dddd;");
     textAreaGrid.setHgap(10);
     textAreaGrid.setVgap(10);
     textAreaGrid.setPadding(new Insets(25, 25, 25, 25));
```

```
titleGridPane.add(title, 0, 0);
     titleGridPane.add(bestSellerText, 0, 1);
     textAreaGrid.add(availableBooksLbl, 0, 0);
     textAreaGrid.add(availableBooksTextArea, 0, 1);
      //GridPane: Declaration and Configuration of titleGridPane. End
<<<<<<<<
     fourthGridPane.add(createBtn, 0, 1);
     fourthGridPane.add(deleteBtn, 0, 2);
     fourthGridPane.add(editBtn, 0, 3);
     fourthGridPane.add(showAvailableBooksBtn, 0, 4);
     fourthGridPane.add(soldBtn, 0, 5);
     //GridPane: Declaration and Configuration of fourthGridPane. End
<<<<<<<<
     //Buttons and associated click events
     searchBtn.setOnAction(e->SearchBtnClicked());
     searchBtn.setStyle("-fx-pref-width: 100px; -fx-background-
color:royalblue; -fx-text-fill: #ffffff");
     createBtn.setOnAction(e->CreateBtnClicked());
     createBtn.setStyle("-fx-pref-width: 100px; -fx-background-color:green; -
fx-text-fill: #ffffff");
     deleteBtn.setOnAction(e->DeleteBtnClicked());
     deleteBtn.setStyle("-fx-pref-width: 100px; -fx-background-color:red; -fx-
text-fill: #ffffff");
     editBtn.setOnAction(e->EditBtnClicked());
     editBtn.setStyle("-fx-pref-width: 100px; -fx-background-color:coral; -fx-
text-fill: #ffffff");
     viewAllBtn.setStyle("-fx-pref-width: 100px");
     addQuantityBtn.setOnAction(e->AddQuantityBtnClicked());
     addQuantityBtn.setStyle("-fx-pref-width: 100px; -fx-background-
color:green; -fx-text-fill: #ffffff");
     removeQuantityBtn.setOnAction(e->RemoveQuantityBtnClicked());
     removeQuantityBtn.setStyle("-fx-pref-width: 100px; -fx-background-
color:red; -fx-text-fill: #ffffff");
      //saveBtn.setOnAction(e->SaveBtnClicked());
     saveBtn.setStyle("-fx-pref-width: 100px; -fx-background-color:green; -fx-
text-fill: #ffffff");
     showAvailableBooksBtn.setOnAction(e->ShowBtnClicked());
     showAvailableBooksBtn.setStyle("-fx-pref-width: 100px; -fx-background-
color:royalblue; -fx-text-fill: #ffffff; -fx-font-alignment:center;");
     soldBtn.setStyle("-fx-pref-width: 100px; -fx-background-color:green; -fx-
text-fill: #ffffff; -fx-font-alignment:center;");
```

```
soldBtn.setOnAction(e->SoldBtnClicked());
      availableBooksTextArea.setStyle("-fx-pref-width: 300px; -fx-pref-height:
350px");
      rootGridPane.add(titleGridPane, 0, 0);
      rootGridPane.add(thirdGridPane, 0, 1);
      rootGridPane.add(firstGridPane, 0, 2);
      rootGridPane.add(secondGridPane, 1, 1);
      rootGridPane.add(fourthGridPane, 1, 2);
      rootGridPane.add(textAreaGrid, 3, 2);
      Scene objectScene = new Scene(rootGridPane, 1100, 700); //SCENE:
Declaration and configuring scene and putting scene gridPane into scene
      window.setScene(objectScene);
      window.show();
   }
  public static void main(String[] args)
     Application. launch (args);
   }
   //Handler Classes inside the ApplicationGUI class.
   //To search a book using name
  public void SearchBtnClicked()
      //clear all textfields
      clearTextFields();
      //loads the driver in memory
      try {
         Class.forName("oracle.jdbc.OracleDriver");
         //Connection connection =
DriverManager.getConnection("jdbc:oracle:thin:@oracle1.centennialcollege.ca:152 1:SQLD", "COMP228SY_F18_41", "password");//connection details
         Connection connection =
DriverManager.getConnection("jdbc:oracle:thin:@199.212.26.208:1521:SQLD",
"COMP228SY F18 41", "password");//connection details
         //String selected= cmbID.getSelectionModel().getSelectedItem();
         Statement showBooksStatement = connection.createStatement();
         ResultSet resultSet = showBooksStatement.executeQuery("select * from
Books where NAME = '"+searchFromInventoryTextField.getText()+"' FETCH FIRST 1
ROWS ONLY");
         // fetch all rows from the table
         // get ResultSet's meta data
         ResultSetMetaData metaData = resultSet.getMetaData();
```

```
int numberOfColumns = metaData.getColumnCount();
         //To print the first book
         while (resultSet.next()) {
            displayDetails(resultSet);
         setEditableFalse();
      catch(ClassNotFoundException e) {
         JOptionPane.showMessageDialog(null, e.getMessage());
         e.printStackTrace();
      catch(SQLException e) {
         JOptionPane.showMessageDialog(null, e.getMessage());
         e.printStackTrace();
      }
   }
   //To add a new book
  public void CreateBtnClicked()
      //To get the new ID
      int nextId=0;
      //loads the driver in memory
      try {
         Class.forName("oracle.jdbc.OracleDriver");
         //Connection connection =
DriverManager.getConnection("jdbc:oracle:thin:@oracle1.centennialcollege.ca:152
1:SQLD", "COMP228SY F18 41", "password");//connection details
        Connection connection =
DriverManager.getConnection("jdbc:oracle:thin:@199.212.26.208:1521:SQLD",
"COMP228SY F18 41", "password");//connection details
         //String selected= cmbID.getSelectionModel().getSelectedItem();
         Statement showBooksStatement = connection.createStatement();
         ResultSet resultSet = showBooksStatement.executeQuery("select * from
BOOKS where BOOKID = (select max(BOOKID) from BOOKS)");
         // fetch all rows from the table
         // get ResultSet's meta data
         ResultSetMetaData metaData = resultSet.getMetaData();
         // display query results
         while (resultSet.next())
         {
            nextId = Integer.parseInt(resultSet.getString("BOOKID"));
      catch(ClassNotFoundException e) {
         JOptionPane.showMessageDialog(null, e.getMessage());
         e.printStackTrace();
      catch(SQLException e) {
        JOptionPane.showMessageDialog(null, e.getMessage());
         e.printStackTrace();
```

```
}
      if(createBtn.getText().equals("Save"))
         setEditableTrue();
         nextId = nextId +1;
         addToInventoryLbl.setText("ItsWorking");
         //loads the driver in memory
         try {
            Class.forName("oracle.jdbc.OracleDriver");
            //Connection connection =
DriverManager.getConnection("jdbc:oracle:thin:@oracle1.centennialcollege.ca:152
1:SQLD", "COMP228SY F18 41", "password");//connection details
            Connection connection =
DriverManager.getConnection("jdbc:oracle:thin:@199.212.26.208:1521:SQLD",
"COMP228SY_F18_41", "password");//connection details
            //\overline{S}tring selected= cmbID.getSelectionModel().getSelectedItem();
            Statement showBooksStatement = connection.createStatement();
            showBooksStatement.executeQuery("insert into Books(BOOKID, NAME,
AUTHOR, PRICE, PUBLISHER, AVAILABLEQUANTITY, SOLDQUANTITY) values (
"+nextId+",'"+ bookNameTextField.getText()
                  +"','"+bookAuthorTextField.getText()+"',
"+Integer.parseInt(bookPriceTextField.getText())+",
'"+bookPublisherTextField.getText()+"',"+
                  Integer.parseInt(quantityAvailableTextField.getText())+",
"+Integer.parseInt(quantitySoldTextField.getText())+")");
            // fetch all rows from the table
            // get ResultSet's meta data
            //ResultSetMetaData metaData = resultSet.getMetaData();
         catch (ClassNotFoundException e) {
            JOptionPane.showMessageDialog(null, e.getMessage());
            e.printStackTrace();
         catch(SQLException e) {
            JOptionPane.showMessageDialog(null, e.getMessage());
            e.printStackTrace();
         }
         clearTextFields();
         createBtn.setText("Add Book");
         quantitySoldTextField.setDisable(false);
        bookIdTextField.setDisable(false);
      else
         clearTextFields();
         setEditableTrue();
        bookIdTextField.setDisable(true);
        bookNameTextField.requestFocus();
         createBtn.setText("Save");
         quantitySoldTextField.setText("0");
         quantitySoldTextField.setDisable(true);
```

```
}
  //To delete a book
  public void DeleteBtnClicked()
      //For the deletion of the selected book
     if (bookIdTextField.getText().isEmpty())
        Alert alert = new Alert(Alert.AlertType.ERROR);
        alert.setTitle("Error Dialog");
        alert.setHeaderText("Look, an Error Dialog");
        alert.setContentText("No Books selected!");
        alert.showAndWait();
     }
     else
      {
        try {
            Class.forName("oracle.jdbc.OracleDriver");
            //Connection connection =
DriverManager.getConnection("jdbc:oracle:thin:@oracle1.centennialcollege.ca:152
1:SQLD", "COMP228SY F18 41", "password");//connection details
            Connection connection =
DriverManager.getConnection("jdbc:oracle:thin:@199.212.26.208:1521:SQLD",
"COMP228SY_F18_41", "password");//connection details
            //String selected= cmbID.getSelectionModel().getSelectedItem();
           Statement showBooksStatement = connection.createStatement();
            ResultSet resultSet = showBooksStatement.executeQuery("delete from
BOOKS where BOOKID = "+Integer.parseInt(bookIdTextField.getText()));
         catch(ClassNotFoundException e) {
            JOptionPane.showMessageDialog(null, e.getMessage());
            e.printStackTrace();
         catch(SQLException e) {
            JOptionPane.showMessageDialog(null, e.getMessage());
            e.printStackTrace();
        clearTextFields();
        ShowBtnClicked();
  }
  //To Edit the details of the book
  public void EditBtnClicked()
     if (bookIdTextField.getText().isEmpty())
        Alert alert = new Alert(Alert.AlertType. ERROR);
        alert.setTitle("Error Dialog");
        alert.setHeaderText("Look, an Error Dialog");
        alert.setContentText("No Books selected!");
        alert.showAndWait();
     else
        if (editBtn.getText().equals("Save"))
```

```
{
            try {
               Class.forName("oracle.jdbc.OracleDriver");
               //Connection connection =
DriverManager.getConnection("jdbc:oracle:thin:@oracle1.centennialcollege.ca:152
1:SQLD", "COMP228SY_F18_41", "password");//connection details
               Connection connection =
DriverManager.getConnection("jdbc:oracle:thin:@199.212.26.208:1521:SQLD",
"COMP228SY F18 41", "password");//connection details
                //String selected= _cmbID.getSelectionModel().getSelectedItem();
               Statement showBooksStatement = connection.createStatement();
               ResultSet resultSet = showBooksStatement.executeQuery("update
BOOKS set NAME = '"+bookNameTextField.getText()+"',
AUTHOR='"+bookAuthorTextField.getText()+"'," +
                     " PRICE =
"+Integer.parseInt(bookPriceTextField.getText())+",
PUBLISHER='"+bookPublisherTextField.getText()+"', AVAILABLEQUANTITY="+Integer.pa
rseInt(quantityAvailableTextField.getText())
                     +",
SOLDQUANTITY="+Integer.parseInt(quantitySoldTextField.getText())+" where
BOOKID="+Integer.parseInt(bookIdTextField.getText()));
            catch(ClassNotFoundException e) {
               JOptionPane.showMessageDialog(null, e.getMessage());
               e.printStackTrace();
            catch(SQLException e) {
               JOptionPane.showMessageDialog(null, e.getMessage());
               e.printStackTrace();
            clearTextFields();
            ShowBtnClicked();
            clearTextFields();
            editBtn.setText("Edit Book");
            quantitySoldTextField.setDisable(false);
            bookIdTextField.setDisable(false);
         else
            setEditableTrue();
            bookIdTextField.setDisable(true);
            bookNameTextField.requestFocus();
            editBtn.setText("Save");
         }
      }
   }
   //For login
  public void LoginBtnClicked(Stage s)
      if(email.getText().isEmpty() || password.getText().isEmpty())
        Alert alert = new Alert(Alert.AlertType.ERROR);
         alert.setTitle("Error Dialog");
         alert.setHeaderText("Look, an Error Dialog");
        alert.setContentText("Please enter your Email ID and Password!");
        alert.showAndWait();
```

```
else
         if (email.getText().equals("admin@gmail.com") &&
password.getText().equals("admin@books"))
            s.close();
         }
         else
            Alert alert = new Alert(Alert.AlertType.ERROR);
            alert.setTitle("Error Dialog");
            alert.setHeaderText("Look, an Error Dialog");
            alert.setContentText("Invalid Email ID or Password!");
            alert.showAndWait();
      }
   }
   //To Add quantity
  public void AddQuantityBtnClicked()
      if (bookIdTextField.getText().isEmpty())
         Alert alert = new Alert(Alert.AlertType.ERROR);
         alert.setTitle("Error Dialog");
         alert.setHeaderText("Look, an Error Dialog");
         alert.setContentText("No Books selected!");
         alert.showAndWait();
      else
         try {
            Class.forName("oracle.jdbc.OracleDriver");
            //Connection connection =
DriverManager.getConnection("jdbc:oracle:thin:@oracle1.centennialcollege.ca:152
1:SQLD", "COMP228SY F18 41", "password");//connection details
            Connection connection =
DriverManager.getConnection("jdbc:oracle:thin:@199.212.26.208:1521:SQLD",
"COMP228SY_F18_41", "password");//connection details
            //String selected= cmbID.getSelectionModel().getSelectedItem();
            Statement showBooksStatement = connection.createStatement();
            ResultSet resultSet = showBooksStatement.executeQuery("update BOOKS
set AVAILABLEQUANTITY = AVAILABLEQUANTITY +
"+Integer.parseInt(addToInventoryTextField.getText())+"where
BOOKID="+Integer.parseInt(bookIdTextField.getText()));
         catch(ClassNotFoundException e) {
            JOptionPane.showMessageDialog(null, e.getMessage());
            e.printStackTrace();
         catch(SQLException e) {
            JOptionPane.showMessageDialog(null, e.getMessage());
            e.printStackTrace();
         clearTextFields();
         ShowBtnClicked();
   }
```

```
//To remove quantity of a book
  public void RemoveQuantityBtnClicked()
      if (bookIdTextField.getText().isEmpty())
         Alert alert = new Alert(Alert.AlertType. ERROR);
         alert.setTitle("Error Dialog");
         alert.setHeaderText("Look, an Error Dialog");
         alert.setContentText("No Books selected!");
         alert.showAndWait();
      else
         try {
            Class.forName("oracle.jdbc.OracleDriver");
            //Connection connection =
\textit{DriverManager.getConnection("jdbc:oracle:thin:@oracle1.centennialcollege.ca:152")} \\
1:SQLD", "COMP228SY F18 41", "password");//connection details
            Connection connection =
DriverManager.getConnection("jdbc:oracle:thin:@199.212.26.208:1521:SQLD",
"COMP228SY F18 41", "password");//connection details
            //String selected= cmbID.getSelectionModel().getSelectedItem();
            Statement showBooksStatement = connection.createStatement();
            ResultSet resultSet = showBooksStatement.executeQuery("update BOOKS
set AVAILABLEQUANTITY = AVAILABLEQUANTITY -
"+Integer.parseInt(addToInventoryTextField.getText())+"where
BOOKID="+Integer.parseInt(bookIdTextField.getText()));
         catch(ClassNotFoundException e) {
            JOptionPane.showMessageDialog(null, e.getMessage());
            e.printStackTrace();
         catch(SQLException e) {
            JOptionPane.showMessageDialog(null, e.getMessage());
            e.printStackTrace();
         clearTextFields();
         ShowBtnClicked();
   }
   //To sell one book
  public void SoldBtnClicked()
      if (bookIdTextField.getText().isEmpty())
         Alert alert = new Alert(Alert.AlertType.ERROR);
         alert.setTitle("Error Dialog");
         alert.setHeaderText("Look, an Error Dialog");
         alert.setContentText("No Books selected!");
         alert.showAndWait();
      else
        try {
            Class.forName("oracle.jdbc.OracleDriver");
            //Connection connection =
DriverManager.getConnection("jdbc:oracle:thin:@oracle1.centennialcollege.ca:152
1:SQLD", "COMP228SY_F18_41", "password");//connection details
```

```
Connection connection =
DriverManager.getConnection("jdbc:oracle:thin:@199.212.26.208:1521:SQLD",
"COMP228SY F18 41", "password");//connection details
            //\overline{S}tring selected= _cmbID.getSelectionModel().getSelectedItem();
            Statement showBooksStatement = connection.createStatement();
            ResultSet resultSet = showBooksStatement.executeQuery("update BOOKS
set AVAILABLEQUANTITY = AVAILABLEQUANTITY - 1 where
BOOKID="+Integer.parseInt(bookIdTextField.getText()));
         catch(ClassNotFoundException e) {
            JOptionPane.showMessageDialog(null, e.getMessage());
            e.printStackTrace();
         catch(SQLException e) {
            JOptionPane.showMessageDialog(null, e.getMessage());
            e.printStackTrace();
         clearTextFields();
         ShowBtnClicked();
      }
   //Method to show all the available books from database
  public void ShowBtnClicked()
      availableBooksTextArea.clear();//clear text area before showing result
     bestSellerText.setText("Best Seller: ");
      //loads the driver in memory
      try {
         Class.forName("oracle.jdbc.OracleDriver");
         //Connection connection =
DriverManager.getConnection("jdbc:oracle:thin:@oracle1.centennialcollege.ca:152
1:SQLD", "COMP228SY F18 41", "password");//connection details
        Connection connection =
DriverManager.getConnection("jdbc:oracle:thin:@199.212.26.208:1521:SQLD",
"COMP228SY F18 41", "password");//connection details
         //String selected= cmbID.getSelectionModel().getSelectedItem();
         Statement showBooksStatement = connection.createStatement();
         //ResultSet resultSet = showBooksStatement.executeQuery("select
NAME, AVAILABLEQUANTITY from Books where AVAILABLEQUANTITY > 0 ");
         ResultSet resultSet = showBooksStatement.executeQuery("select PRICE,
NAME from Books order by SOLDQUANTITY desc ");
         // fetch all rows from the table
         // get ResultSet's meta data
         ResultSetMetaData metaData = resultSet.getMetaData();
         int numberOfColumns = metaData.getColumnCount();
         // display the names of the columns in the ResultSet
         for (int i = 1; i <= numberOfColumns; i++) {</pre>
            availableBooksTextArea.appendText(String.format("%-
10s\t\t", metaData.getColumnName(i)));
         availableBooksTextArea.appendText("\n\n");
```

```
Label tempLabel= new Label("");
         //VBox vbox = new VBox();
         //vbox.setPadding(new Insets(10));
         //vbox.setSpacing(8);
         // display query results
         int j=1;
         while (resultSet.next())
            if(j==1)
               bestSellerText.setText(bestSellerText.getText() +
resultSet.getString("NAME"));
            for (int i = 1; i <= numberOfColumns; i++)</pre>
               availableBooksTextArea.appendText(String.format("%s\t\t\t",
resultSet.getObject(i)));
            availableBooksTextArea.appendText("\n");
            j++;
         }
      }
      catch(ClassNotFoundException e) {
         JOptionPane.showMessageDialog(null, e.getMessage());
         e.printStackTrace();
      catch(SQLException e) {
        JOptionPane.showMessageDialog(null, e.getMessage());
        e.printStackTrace();
      }
   //Method to clear all the text fields
  public void clearTextFields()
     bookIdTextField.clear();
     bookAuthorTextField.clear();
     bookNameTextField.clear();
     bookPriceTextField.clear();
     bookPublisherTextField.clear();
     quantityAvailableTextField.clear();
      quantitySoldTextField.clear();
   //Method to print details
  public void displayDetails(ResultSet resultSet)
        bookIdTextField.appendText(resultSet.getString("BOOKID"));
        bookAuthorTextField.appendText(resultSet.getString("AUTHOR"));
        bookNameTextField.appendText(resultSet.getString("NAME"));
        bookPriceTextField.appendText(resultSet.getString("PRICE"));
        bookPublisherTextField.appendText(resultSet.getString("PUBLISHER"));
         if(Integer.parseInt(resultSet.getString("AVAILABLEQUANTITY"))<0)</pre>
            quantityAvailableTextField.appendText("0");
```

```
else
quantityAvailableTextField.appendText(resultSet.getString("AVAILABLEQUANTITY"))
         quantitySoldTextField.appendText(resultSet.getString("SOLDQUANTITY"));
      catch(SQLException e) {
         JOptionPane.showMessageDialog(null, e.getMessage());
         e.printStackTrace();
      }
   }
   //To set the text boxes as read only
  public void setEditableFalse()
     bookIdTextField.setEditable(false);
     bookNameTextField.setEditable(false);
     bookAuthorTextField.setEditable(false);
     bookPriceTextField.setEditable(false);
     bookPublisherTextField.setEditable(false);
      quantitySoldTextField.setEditable(false);
      quantityAvailableTextField.setEditable(false);
   }
   //To set the text fields as editable
  public void setEditableTrue()
     bookIdTextField.setEditable(true);
     bookNameTextField.setEditable(true);
     bookAuthorTextField.setEditable(true);
     bookPriceTextField.setEditable(true);
     bookPublisherTextField.setEditable(true);
     quantitySoldTextField.setEditable(true);
      quantityAvailableTextField.setEditable(true);
```