**FINAL PROJECT – BCS 345 Java Programming**

**Due: 12/4/2019 @ 10:50am**

SUBMIT ALL PROJECTS NECESSARY TO RUN THIS PROGRAM. When you submit the assignment in the Blackboard dropbox make sure your zip file is attached to the submission.

IMPORTANT – Make sure you ***properly comment*** AND ***properly indent*** your program. The commenting and indenting documents are on Blackboard.

***VERY IMPORTANT – IF THE PROGRAM DOES NOT COMPILE THERE WILL BE MAJOR POINTS TAKEN OFF.***

***Overview***

You will create a JavaFX GUI application to display purchase data. You should use FXML to define the GUI. You need to do the following:

1. Write an FXML file that defines the GUI.
2. Write the PurchasesApplication class.
3. Write the PurchasesController class.
4. Write the PurchasesGraphicalUI class.
5. Add a new menu item to the main console UI menu.

**The GUI should be created using JavaFX and FXML. There should be a controller class to handle all window events (PurchasesController).**

Check the “BCS 345 Eclipse Create Project And Package” document on Blackboard for instructions on how to create projects and packages and how to import them.

***Specifications***

Use FXML to define the GUI. There should be one window in this application. This window should have a menu at the top and a tab pane control taking up the main part of the window. The tabs should contain the following data:

First Tab – Customer

1. First name
2. Last name
3. Address

Second Tab - Purchases

1. List of purchases (use a ListView).

**Class – PurchasesApplication**

Store in project/package: <your last name>.bcs345.purchases.presentation

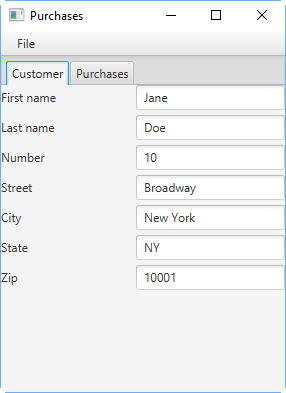
Should have an override of start. The start override should have code that loads the FXML file and that shows the window.

**Class – PurchasesController**

Store in project/package: <your last name>.bcs345.purchases.presentation

GUI event handlers should be defined in this class. This class should have an instance of PurchaseCollection as a member variable.

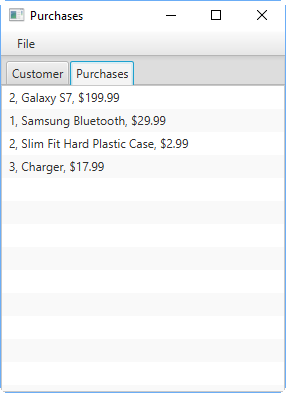
***Main Display***



**TabPane control**

**Customer Tab Visible**

The Customer tab shows the first name, last name, and address data.



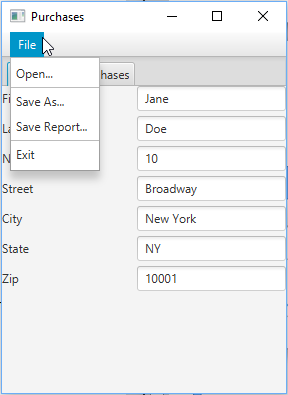
**Multiple items are showing in the ListView (not one large string). The red dashed lines will not appear in the ListView. The red dashed lines are there to accentuate the fact that they are different items in the ListView.**

**ListView conrol**

**Purchases Tab Visible**

The Purchases tab shows all the purchase data in a ListView.

The next screen shot shows what the menu should look like. You should add separators in between groups of related menu items as shown in the picture.



**Separator**

**Menu**

**Menu Item**

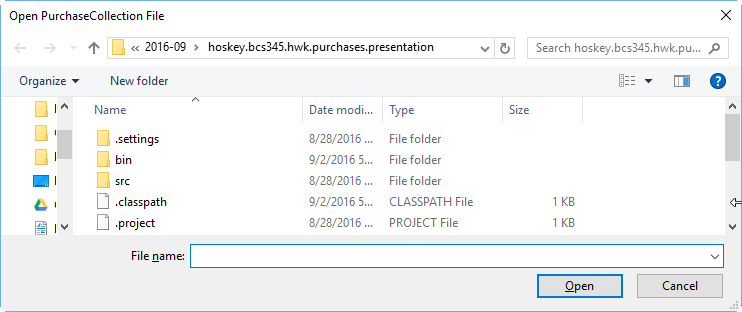
***Menu Event Handling***

**Open Menu Item**

When the user selects the Open menu item the program should display a FileChooser and let the user decide which file to read data from.

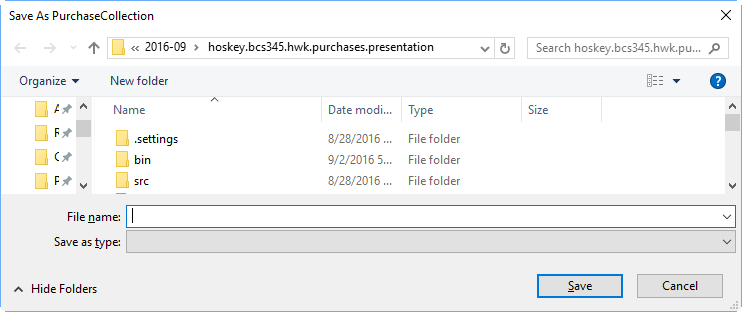
**The data from the selected file should be read into the PurchaseCollection instance and then be displayed in the appropriate controls. You should clear the purchases ListView items before adding the data that was just read from the file (the purchases from the file).**

As detailed above, the PurchaseCollection instance is declared in the PurchasesController class. The user should be able to cancel out of the open dialog and have it not affect the state of the program. To be clear, if the user cancels out of the open file dialog then the data that was being displayed and contained in the PurchaseCollection instance should not change. Here is a picture of the open file dialog:



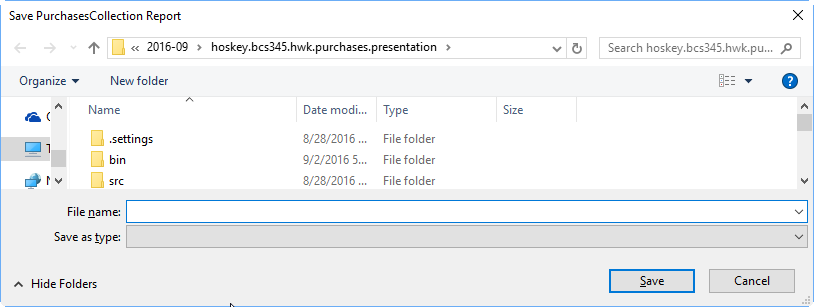
**Save As Menu Item**

When the User selects the Save As menu item the program should display a FileChooser and let the user decide which file to write data to. The program should then write data from the PurchasesCollection instance into the selected file. Just to be clear, this menu item writes data and not a report (no descriptive text of any kind). As with open, the user should be able to cancel out of the save file dialog and have it not change the state of the application. Here is a picture of the Save As dialog:



**Save Report Menu Item**

When the User selects the Save Report menu item the program should display a FileChooser and let the user decide which file to write a report to. The program should then write the report to the selected file. Just to be clear, this menu item writes a full report not just data. The format of the report is the exact same as for assignment 4 (check assignment 4 for the report format). The user should be able to cancel out of the save report dialog and have it not change the state of the application. Here is a picture of the Save Report dialog:



**Exit Menu Item**

When the user chooses Exit the menu item it should close the window and go back to the main menu console window prompt that created the window in the first place.

**Class – PurchasesGraphicalUI**

Store in project/package: <your last name>.bcs345.purchases.presentation

***Member Variables (all private)***

|  |  |  |
| --- | --- | --- |
| **Variable** | **Data Type** | **Description** |
| No member variables |  |  |

***Member Method Signatures and Descriptions (all public)***

|  |  |
| --- | --- |
| Signature | Description |
| ShowUI() | Shows the user interface. When this method is called it should launch Purchases GUI Application. |

**Class – Main (revise from previous assignment)**

Store in project/package: <your last name>.bcs345.purchases.presentation

***Member Variables (all private)***

|  |  |  |
| --- | --- | --- |
| **Variable** | **Data Type** | **Description** |
| No member variables |  |  |

***Member Method Signatures and Descriptions (all public)***

|  |  |
| --- | --- |
| **Signature** | **Description** |
| public static void main(String args[]) | Update the method so that the user is given another choice of which user interface to use. Update the menu from the previous assignment. Add a choice for PurchasesGraphicalUI. Look in the “Menu That Main Shows“ section below for the menu to display. The menu should keep showing until the user chooses to exit. |

***Deliverables***

***One*** winzip file containing **ALL** Eclipse projects necessary to run the program.

***Menu that Main Shows***

**Choose UI**

**---------**

**1 – CustomerPurchaseConsoleUI**

**2 – PurchaseCollectionConsoleUI**

**3 – PurchasesGraphicalUI**

**4 - Exit**

**Enter Choice:**