Check off the steps that you have completed.

Part 1

* Read the directions in Taskstream
* [Watch introductory video](https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=9740b11c-910b-4e97-9440-610ac2f77df1)
* [Read the FAQ](ttps://docs.google.com/a/wgu.edu/document/d/1DkWwPECpDpIi9o3VHki31pKaUmuCpweh1Sov6ZdM1z0/edit?usp=sharing)
* Using your WGU email address, access the support document [with this link](https://docs.google.com/a/wgu.edu/document/d/1xVhO8bfRWEwy1dP7xYwCgYj-YwcCmkBHBL0bRsf3woM/edit?usp=sharing).
* Downloaded [NetBeans for Java version 8.1](https://netbeans.org/community/releases/81/install.html) or later
* Install [Scene Builder](http://gluonhq.com/products/scene-builder/)
* Watch [Introduction to Scene Builder](https://wgu.adobeconnect.com/p7pl48k7zmf/)
* Read [JavaFX Tutorial](http://code.makery.ch/library/javafx-8-tutorial/part1/), parts 1-3.

Part 2

* Created a JavaFX FXML project in NetBeans. (Please put your name in the project name, otherwise there will be several hundred zipped projects all named InventorySystem.)
* In your Source Packages, create folders for View\_Controller and Model. Remove the FXMLDocument.FXML andFXMLDocumentController.java files.
* In the View\_Controller folder, build the following screens, by right clicking on the folder, choosing the New menu item, choosing Empty FXML, naming it, and clicking Next, and then checking the Use Java Controller and creating a new one. Then use Scene Builder to
  + Build the main screen
  + Build the Add/Modify Outsourced Part screen
  + Build the Add/Modify Inhouse Part screen
  + Build the Add/Modify Product Screen

Part 3

* Read [Using JavaFX Properties and Binding](http://docs.oracle.com/javafx/2/binding/jfxpub-binding.htm)
* In the Model folder, create the following classes, following the UML diagram and using properties
  + Inventory
  + Product
  + Part (this is an abstract class)
  + InhousePart
  + OutsourcedPart
* In the screens in the View\_Controller folder, give all the views FXIDs and all the buttons handlers

Part 4

* Using Scene Builder’s View menu’s Show Sample Controller Skeleton, update the controller file for each screen.
* Watch [Populating a TableView](https://wgu.adobeconnect.com/p5ko0grzeyc/)
* Read [Switching to Different Screens in JavaFX and FXML](http://www.javafxtutorials.com/tutorials/switching-to-different-screens-in-javafx-and-fxml/)
* Use the main screen’s add part button handler to open the Add/Modify Inhouse part screen and create an InhousePart and display it on the Part TableView . You will have to import some classes, and add a throws clause to the button handler.
* Enable the user to pick whether to create an InhousePart or an OutsourcedPart using the radio button
* Enable the user to cancel out of the Add/Modify Part screens
* Watch [Searching a TableView](https://wgu.adobeconnect.com/p3sia4yyjns/) and [Modifying a TableView](https://wgu.adobeconnect.com/p2lyz4njxrt/)
* Enable the user to search the Part TableView on the main screen.
* Enable the user to Modify a Part using the Modify button on the main screen, which loads the modify part screen. You will probably want to create a public static part in the Main screen’s controller, so the Add Part screen knows what part is chosen, and display the part’s values in the TextFields using the initialize method of the controller of the Add Part screen.
* Enable the user to Delete a Part using the Delete button on the main screen.
* Enable the user to Add a Product using the add product button on the main screen to open the Add Product Screen and create a product and display it on the Product TableView.
* Enable the user to Modify a Product using the Modify and Delete buttons on the main screen, and the modify, delete and save buttons on the Add/Modify a Product Screen.
* Enable the user to Delete a Product using the delete product button on the main screen.
* Enable the user to Exit the main screen.

Part 5

* Read [Dialog Boxes](http://code.makery.ch/blog/javafx-dialogs-official/)
* Write the following Exception Controls in methods in the Main Screen, Part and Product Controllers.
  + Inventory must be between the minimum or maximum value for that Part or Product
  + Maximum must have a value greater than minimum
  + Minimum must have a value less than maximum
  + Products must have at least one part
  + Products with at least one part cannot be deleted
  + Add a Confirm Dialogue for all Delete and Cancel buttons
  + Ensure that the price of product cannot be less than the cost of the parts
  + Ensure that a product must have a name, price and inventory level(default 0)
* Verify that everything looks good!