REQUIREMENTS

Your submission must be your original work. No more than a combined total of 30% of the submission and no more than a 10% match to any one individual source can be directly quoted or closely paraphrased from sources, even if cited correctly. An originality report is provided when you submit your task that can be used as a guide.

You must use the rubric to direct the creation of your submission because it provides detailed criteria that will be used to evaluate your work. Each requirement below may be evaluated by more than one rubric aspect. The rubric aspect titles may contain hyperlinks to relevant portions of the course.

I. User Interface

Create a JavaFX application with a graphical user interface (GUI) based on the attached "GUI Mock-Up". Write code to display *each* of the following screens in the GUI:

A. A main screen, showing the following controls:
- buttons for "Add", "Modify", "Delete", "Search" for parts and products, and "Exit"
- lists for parts and products
- text boxes for searching for parts and products
- title labels for parts, products, and the application title

B. An add part screen, showing the following controls:
- radio buttons for "In-House" and "Outsourced" parts
- buttons for "Save" and "Cancel"
- text fields for ID, name, inventory level, price, max and min values, and company name or machine ID
- labels for ID, name, inventory level, price/cost, max and min values, the application title, and company name or machine ID

C. A modify part screen, with fields that populate with presaved data, showing the following controls:
- radio buttons for "In-House" and "Outsourced" parts
- buttons for "Save" and "Cancel"
- text fields for ID, name, inventory level, price, max and min values, and company name or machine ID
- labels for ID, name, inventory level, price, max and min values, the application title, and company name or machine ID

D. An add product screen, showing the following controls:
- buttons for "Save", "Cancel", "Add" part, and "Delete" part
- text fields for ID, name, inventory level, price, and max and min values
- labels for ID, name, inventory level, price, max and min values, and the application
- a list for associated parts and their products
- a "Search" button and a text field with an associated list for displaying the results of the search

E. A modify product screen, with fields that populate with presaved data, showing the following controls:
- buttons for "Save", "Cancel", "Add" part, and "Delete" part
- text fields for ID, name, inventory level, price, and max and min values
- labels for ID, name, inventory level, price, max and min values, and the application
- a list for associated parts and their products
- a "Search" button and a text field with associated list for displaying the results of the search

II. Application

Now that you've created the GUI, write code to create the class structure provided in the attached "UML (unified modeling language) Class Diagram". Enable *each* of the following capabilities in the application:

F. Using the attached "UML Class Diagram", create appropriate classes and instance variables with the following criteria:
- five classes with the 16 associated instance variables
- variables are only accessible through getter methods
- variables are only modifiable through setter methods

G. Add the following functionalities to the main screen, using the methods provided in the attached "UML Class Diagram":
- redirect the user to the "Add Part", "Modify Part", "Add Product", or "Modify Product" screens
- delete a selected part or product from the list
- search for a part or product and display matching results
- exit the main screen

H. Add the following functionalities to the part screens, using the methods provided in the attached "UML Class Diagram":
1. "Add Part" screen
   - select "In-House" or "Outsourced"
   - enter name, inventory level, price, max and min values, and company name or machine ID
   - save the data and then redirect to the main screen
   - cancel or exit out of this screen and go back to the main screen
2. "Modify Part" screen
   - select "In-House" or "Outsourced"
   - modify or change data values
   - save modifications to the data and then redirect to the main screen
   - cancel or exit out of this screen and go back to the main screen

I. Add the following functionalities to the product screens, using the methods provided in the attached "UML Class Diagram":
1. "Add Product" screen
   - enter name, inventory level, price, and max and min values
   - save the data and then redirect to the main screen
   - associate one or more parts with a product
   - remove or disassociate a part from a product
   - cancel or exit out of this screen and go back to the main screen
2. "Modify Product" screen
   - modify or change data values
   - save modifications to the data and then redirect to the main screen
   - associate one or more parts with a product
   - remove or disassociate a part from a product
   - cancel or exit out of this screen and go back to the main screen

J. Write code to implement exception controls with custom error messages for one requirement out of *each*of the following sets (pick one from each):
1. Set 1
   - entering an inventory value that exceeds the minimum or maximum value for that part or product
   - preventing the minimum field from having a value above the maximum field
   - preventing the maximum field from having a value below the minimum field

    - ensuring that a product must always have at least one part

2. Set 2
    - including a confirm dialogue for all “Delete” and “Cancel” buttons
    - ensuring that the price of a product cannot be less than the cost of the parts
    - ensuring that a product must have a name, price, category, and inventory level (default 0)

K. Demonstrate professional communication in the content and presentation of your submission.