Task 1

Tests for QKM2 (C482)

SOFTWARE 1 (JAVA)

WGU Evaluation

Table of Contents

Note to Students

These are the main tests that evaluation will be performing on your work. Please keep in mind that additional tests may be needed depending on what is found when we review your work. This document is meant to be a guide and not a definitive list of what the complete set of tests will be. In most cases if your submission passes all these tests your submission should pass. We will also use different values for our test cases. Don't code your submission to only work with these test cases. The evaluators may perform slightly different tests, combine tests, or run the tests in a different order than presented in this document.

Also, we want to remind you not to use work you found on the internet. While in production environments you will be modifying code from the internet, that is not what you are being tested on here. This assignment is testing your ability to write your own code from the ground up. We feel very bad when we see a good student fall into this trap. Remember the consequences for using work you found online could include being dismissed from the university. **Don't let this be you.**

A1: Main Form Tests

All tests must pass for this aspect to pass

Preconditions for all Tests

- The application needs to compile
- The application needs to run
- The application needs to open to the Main Form

Special Notes

a) Buttons for the search may not or may not be present. (Some designs auto search others use a button)

b) Not checking functionality on this aspect

c) Not checking exception controls on this aspect

Test 1 (GUI Components)

Steps:

1. Run the application

2. Examine the Main GUI form and ensure the following are present

3. Select present for each item that is found

Element	**Present**	**Not present**
Label for Main Form Title		
Label for Parts Table		
Label for Product Table		
Table for Parts		
Table for Products		
Button for Add Parts (accurate text on button)		
Button for Modify Parts (accurate text on button)		
Button for Delete Parts (accurate text on button)		
Button for Add Products (accurate text on button)		
Button for Modify Products (accurate text on button)		
Button for Delete Products (accurate text on button)		
Exit Button (accurate text on button)		
Search field for part		
Search field for product		

Expected results:

- **All the GUI components are found on the Main Form**
- **The grid above should show all aspects are present**

A2: Add Part Form Tests

All tests must pass for this aspect to pass

Preconditions for all Tests

- The application needs to compile
- The application needs to run
- The application needs to open to the Add Part Form

Special Notes

a) Not checking functionality on this aspect

b) Not checking exception controls on this aspect

Test 1 (GUI Components)

Steps:

1. **From the Main Form click the Add Part Button**
2. **Examine the Add Part GUI form and ensure the following are present**
3. **Select present for each item that is found**

Element	**Present**	**Not present**
Label for Add Part Form (accurate text on label)		
Radio Button for In-House		
Radio Button for Outsourced		
Label for ID (accurate text on label)		
Label for Name (accurate text on label)		
Label for Inventory (accurate text on label)		
Label for Price/Cost (accurate text on label)		
Label for Max (accurate text on label)		
Label for Min (accurate text on label)		
Label for MachineID/Company Name (The text does not need to switch for this aspect to pass)		
TextBox for ID (Does not need to be disabled and Auto Incremented for this aspect to pass)		
TextBox for Name		
TextBox for Inventory		
TextBox for Price/Cost		
TextBox for Max		
TextBox for Min		
TextBox for MachineID (Company Name)		
Button for Save (accurate text on button)		
Button for Cancel (accurate text on button)		

Expected results:

- **All the GUI components are found on the Add Part Form**
- **The grid above should show all aspects are present**

A3: Modify Part Form Tests

All tests must pass for this aspect to pass

Preconditions for all Tests

- The application needs to compile
- The application needs to run
- A Part needs to either be entered by the evaluator or a Part needs to be present when the application opens
- The application needs to open the Modify Part Form

Special Notes

a) Not checking functionality on this aspect

b) Not checking exception controls on this aspect

Test 1 (GUI Components)

Steps:

1. **From the Main Form select a Part from the Parts Table**
2. **Click the Modify Part Button**
3. **Examine the Modify Part GUI form and ensure the following are present**
4. **Select present for each item that is found**

Element	**Present**	**Not present**
Label for Modify Part Form (accurate text on label)		
Radio Button for In-House		
Radio Button for Outsourced		
Label for ID (accurate text on label)		
Label for Name (accurate text on label)		
Label for Inventory (accurate text on label)		
Label for Price/Cost (accurate text on label)		
Label for Max (accurate text on label)		
Label for Min (accurate text on label)		
Label for MachineID (The text does not need to switch for this aspect to pass)		
TextBox for ID (Does not need to be disabled and Auto Incremented for this aspect to pass)		
TextBox for Name		
TextBox for Inventory		
TextBox for Price/Cost		
TextBox for Max		
TextBox for Min		
TextBox for MachineID (Company Name)		
Button for Save (accurate text on button)		
Button for Cancel (accurate text on button)		

Expected results:

- **All the GUI components are found on the Modify Part Form**
- **The grid above should show all aspects are present**

A4: Add Product Form Tests

All tests must pass for this aspect to pass

Preconditions for all Tests

- The application needs to compile
- The application needs to run
- The application needs to open the Add Product Form

Special Notes

a) Not checking functionality on this aspect

b) Not checking exception controls on this aspect

Test 1 (GUI Components)

Steps:

1. **From the Main Form click the Add Product Button**
2. **Examine the Add Product GUI form and ensure the following are present**
3. **Select present for each item that is found**

Element	**Present**	**Not present**
Label for Add Product Form (accurate text on label)		
Label for ID (accurate text on label)		
Label for Name (accurate text on label)		
Label for Inventory (accurate text on label)		
Label for Price (accurate text on label)		
Label for Min (accurate text on label)		
Label for Max (accurate text on label)		
TextBox for ID (Does not need to be disabled and Auto Incremented for this aspect to pass)		
TextBox for Name		
TextBox for Inventory		
TextBox for Price		
TextBox for Min		
TextBox for Max		
TextBox for Search		
Grid for Finding Parts		
Grid for showing Parts Associated with the Current Product		
Button for Adding Part (accurate text on button)		
Button for Removing Part (accurate text on button)		
Button for Save (accurate text on button)		
Button for Cancel (accurate text on button)		

Expected results:

- **All the GUI components are found on the Add Product Form**
- **The grid above should show all aspects are present**

A5: Modify Product Form Tests

All tests must pass for this aspect to pass

Preconditions for all Tests

- The application needs to compile
- The application needs to run
- A Product needs to either be entered by the evaluator or a Product needs to be present when the application opens
- The application needs to open to the Modify Product Form

Special Notes

a) Not checking functionality on this aspect

b) Not checking exception controls on this aspect

Test 1 (GUI Components)

Steps:

1. **On the Main Form select a Product from the Products Table**
2. **Click the Modify Product Button**
3. **Examine the Modify Product GUI form and ensure the following are present**
4. **Select present for each item that is found**

Element	**Present**	**Not present**
Label for Modify Product Form (accurate text on label)		
Label for ID (accurate text on label)		
Label for Name (accurate text on label)		
Label for Inventory (accurate text on label)		
Label for Price (accurate text on label)		
Label for Min (accurate text on label)		
Label for Max (accurate text on label)		
TextBox for ID (Does not need to be disabled and Auto Incremented for this aspect to pass)		
TextBox for Name		
TextBox for Inventory		
TextBox for Price		
TextBox for Min		
TextBox for Max		
TextBox for Search		
Grid for Finding Parts		
Grid for showing Parts Associated with the Current Product		
Button for Adding Part (accurate text on button)		
Button for Removing Part (accurate text on button)		
Button for Save (accurate text on button)		
Button for Cancel (accurate text on button)		

Expected results: • **All the GUI components are found on the Modify Product Form** • **This the grid above should show all aspects are present**

B: JavaDoc Comments Tests

All tests must pass for this aspect to pass

Preconditions for all Tests

- None

Special Notes

a) Every project will have some small issue that the developer had to resolve. Comments that state there were no issues will be rejected for this aspect.

b) While evaluation doesn't have a percentage on the completed docs for the methods, we do look for solid coverage.

c) The comments for the methods do need to have some detail as to what the method is doing. This is where you show that you understand how the code works.

Test 1 (JavaDocs)
Steps:
1. Examine the JavaDocs and ensure the following are all found in them

Element	**Present**	**Not present**
Each method has a JavaDoc comment		
There is a comment that outlines errors that were addressed		
There is a comment that outlines future enhancements that would extend functionality		

Expected results:
• All three elements are complete in the JavaDocs

C: Class Structure Tests

All tests must pass for this aspect to pass

Preconditions for all Tests

None

Special Notes

a) Inheritance is checked in this aspect. Don't do things to the class structure to work around using proper inheritance in Java (i.e., Don't use a variable that tags if a Part is In-House or Outsourced).

b) Anything that attempts to get around the core competencies outlined in the task instructions will cause this aspect to not pass.

Test 1 (Classes)

Steps:

1. **Examine the classes and ensure the following are found**

Element	**True**	**False**
The part class is not modified		
The part class is abstract		
The In-House class inherits from the part class		
The Outsourced class inherits from the part class		
The In-House class includes the methods and variables from the UML Diagram		
The Outsourced class includes the methods and variables from the UML Diagram		
The Product class includes the methods and variables from the UML Diagram		
The Inventory class includes the methods and variables from the UML Diagram		
No extra variables or methods are created to store values for the part type (In-House/outsourced) or work around using proper inheritance		

Expected results:

- **All the elements are true in the classes**

D1: Parts Pane Tests

All tests must pass for this aspect to pass

Preconditions for all Tests

- The application needs to compile
- The application needs to run
- The application needs to open to the Main Form
- The Main Form GUI aspect (aspect A1) must be passing
- A Part needs to either be entered by the evaluator or a Part needs to be present when the application opens

Special Notes

a) Not checking exception controls on this aspect

b) The search can be by ID or Name (ID is the most common)

Test 1 (Add Part Button)
Steps: **1. Click the Add Part Button**
Expected results: **• The Add Part Form opens**

Test 2 (Modify Part Button)
Steps: **1. Highlight a Part from the Parts Table** **2. Click the Modify Part Button**
Expected results: **• The Modify Part Form opens**

Test 3 (Delete Part Button)
Steps: **1. Highlight a Part from the Parts Table** **2. Click the Delete Part Button**
Expected results: **• The highlighted Part from the Parts Table disappears** **OR** **• The part isn't deleted but a the application displays an error message in the UI or in a dialog box providing a detailed and valid reason for not deleting the part**

Test 4 (Search Part 1)

Steps:

1. **Enter text that matches a Part from the Parts Table into the Search TextBox (Or enter an ID that matches the ID of a Part from the Parts Table)**
2. **Press Search Button (If used)**
3. **Clear the text from the Search TextBox**
4. **Press Search Button (if used)**

Expected results:

- **Part matching text (or ID) is highlighted**

OR

- **Only the part matching the text (or ID) is shown in the table**

OR

- **Parts with partial text that matches the text (or ID) is shown in the table**

After step 4:

- **All the parts are returned to the table**

Test 5 (Search Part 2)

Steps:

1. **Enter text that does not matches a Part from the Parts Table into the Search TextBox**
2. **Press Search Button (If used)**

Expected results:

- **The application displays an error message in the UI or in a dialog box with a message that alerts the user that the Part could not be found**

D2: Products Pane Tests

All tests must pass for this aspect to pass

Preconditions for all Tests

- The application needs to compile
- The application needs to run
- The application needs to open to the Main Form
- The Main Form GUI aspect (aspect A1) must be passing
- A Product needs to either be entered by the evaluator or a Product needs to be present when the application opens

Special Notes

a) Not checking exception controls on this aspect

b) The search can be by ID or Name (ID is the most common)

Test 1 (Add Product Button)
Steps: **1. Click the Add Product Button**
Expected results: **• The Add Product Form opens**

Test 2 (Modify Product Button)
Steps: **1. Select a Product from the Main Form** **2. Click the Modify Product Button**
Expected results: **• The Modify Product Form opens**

Test 3 (Delete Product Button)
Steps: **1. Highlight a Product from the Product table** **2. Click the Delete Product Button**
Expected results: **• The highlighted product from the product table disappears** **OR** **• The product isn't deleted but the application displays an error message in the UI or in a dialog box providing a detailed and valid reason for not deleting the product**

Test 4 (Search Product 1)

Steps:

5. Enter text that matches a Product from the Products Table into the Search TextBox (Or enter an ID that matches the ID of a Product from the Products Table)
1.
2. Press Search Button (If used)
3. Clear the text from the Search TextBox
4. Press Search Button (if used)

Expected results:

- Product matching text (or ID) is highlighted

OR

- Only the product matching the text (or ID) is shown in the table

OR

- Products with partial text that matches the text (or ID) is shown in the table

After step 4:

- All the products are returned to the table

Test 5 (Search Product 2)

Steps:

1. Enter text (or ID) that does not matches a Product from the Products Table into the Search TextBox
2. Press Search Button (If used)

Expected results:

- The application displays an error message in the UI or in a dialog box with a message that alerts the user that the product could not be found

D3: Exit Button Tests

All tests must pass for this aspect to pass

Preconditions for all Tests

- The application needs to compile
- The application needs to run
- The application needs to open to the Main Form
- The Main Form GUI aspect (aspect A1) must be passing
- The classes (aspect C) must be passing

Special Notes

a) This aspect does not test exception controls

Test 1 (Exit Button)
Steps: **1. Press the Exit Button**
Expected results: • **The application should close**

E1: Add Part Form Functionality Tests

All tests must pass for this aspect to pass

Preconditions for all Tests

- The application needs to compile
- The application needs to run
- The application needs to open to the Add Part Form
- The Main Form GUI aspect (aspect A1) must be passing
- The Add Part Form GUI aspect (aspect A2) must be passing
- The Modify Part Form GUI aspect (aspect A3) must be passing
- The classes (aspect C) must be passing

Special Notes

a) This aspect does not test exception controls
b) All tests for this aspect will use valid data
c) Evaluators may use different values in the test (Red Text)
d) Inventory, min, max and machine ID will always be tested as whole numbers
e) Price will always be tested as a decimal number

Test 1 (Radio Buttons)

Steps:

1. **Click the Add Part Button on the Main Form**
2. **Select the In-House Radio Button**
3. **Note the text in the MachineID/Company Name Label**
4. **Select the Outsourced Radio Button**
5. **Note the text in the MachineID/Company Name Label**
6. **Select the In-House Radio Button**
7. **Note the text in the MachineID/Company Name Label**

Expected results:

- **When the In-House radio button is selected the text for the MachineID/Company Name Label should say Machine ID**
- **When the Outsourced radio button is selected the text for the machineID/Company Name Label should say Outsourced**

Test 2 (Part ID)

Steps:

1. **Click the Add Part Button on the Main Form**
2. **Inspect the Part ID Textbox**

Expected results:

- **The part ID is auto created, and the TextBox is disabled**

Test 3 (Adding a Part)

Steps:

1. **Click the Add Part Button on the Main Form**
2. **Select “In-House” from the Radio Buttons**
3. **Enter “Wheel” into the Part Name TextBox**
4. **Enter “5” into the Inventory TextBox**
5. **Enter “2.99” into the Price TextBox**
6. **Enter “10” into the Max TextBox**
7. **Enter “1” into the Min TextBox**
8. **Press the Save Button**
9. **Select the Part from the Parts Table on the Main Form**
10. **Click the Modify Part Button on the Main Form**
11. **Check the values on the Modify Part form against the values entered**

Expected results:

- **The TextBoxes allow the data to be entered**
- **No error messages or crashes happen when the Save Button is clicked**
- **The application returns the user to the Main Form when the Save Button is clicked**
- **The data inside the Modify Part Form matches the values entered**

Test 4 (Cancel Button)

Steps:

1. **Click the Add Part Button on the Main Form**
2. **Click the Cancel Button**

Expected results:

- **The Add Part Form closes, and the application returns to the Main Form**

E2: Modify Part Form Functionality Tests

All tests must pass for this aspect to pass

Preconditions for all Tests

- The application needs to compile
- The application needs to run
- The application needs to open to the Add Part Form
- The Main Form GUI aspect (aspect A1) must be passing
- The Modify Part Form GUI aspect (aspect A3) must be passing
- A Part needs to either be entered by the evaluator or a Part needs to be present when the application opens
- The classes (aspect C) must be passing

Special Notes

a) This aspect does not test exception controls
b) All tests for this aspect will use valid data
c) Evaluators may use different values in the test (Red Text)
f) Inventory, min, max and machine ID will always be tested as whole numbers
g) Price will always be tested as a decimal number
h) Inventory, min, max and machine ID will always be tested as whole numbers
i) Price will always be tested as a decimal number

Test 1 (Radio Buttons)

Steps:

1. **Select a Part from the parts Table on the Main Form**
2. **Click the Modify Part Button on the Main from**
3. **Select the In-House radio Button**
4. **Note the text in the MachineID/Company Name Label**
5. **Select the Outsourced Radio Button**
6. **Note the text in the MachineID/Company Name Label**
7. **Select the In-House Radio Button**
8. **Note the text in the MachineID/Company Name Label**

Expected results:

- **When the In-House Radio Button is selected the text for the MachineID/Company Name Label should say Machine ID**
- **When the Outsourced Radio Button is selected the text for the MachineID/Company Name Label should say Outsourced**

Test 2 (Part ID)

Steps:

1. **Select a Part from the Parts Table on the Main Form**
2. **Click the Modify Part Button on the Main Form**
3. **Inspect the Part ID TextBox**

Expected results:

- The TextBox is disabled but shows the part ID

Test 3 (Modify Existing)

Steps:

1. Select a Part from the Parts Table on the Main Form
2. Click the Modify Part Button on the Main from
3. Select "Outsourced" from the radio buttons
4. Enter "Brake" into the Part Name TextBox
5. Enter "15" into the Inventory TextBox
6. Enter "12.99" into the Price TextBox
7. Enter "120" into the Max TextBox
8. Enter "10" into the Min TextBox
9. Enter "PD-Brite" in the Company Name TextBox
10. Press the Save Button
11. Click the Modify Part button on the Main Form
12. Check the values on the Modify Part form against the values entered
13. Select "In-House" from the radio buttons
14. Enter "44" in the Company Name TextBox
15. Press the Save Button
16. Click the Modify Part button on the Main Form
17. Check the values on the Modify Part form against the values entered

Expected results:

- The TextBoxes allow the data to be entered
- No error messages or crashes happen when the save button is clicked
- The application returns the user to the Main Form when the Save Button is clicked
- The data inside the Modify Part Form matches the values entered for both step 12 and step 17

Test 4 (Cancel Button)

Steps:

1. Select a Part from the Parts Table on the Main Form
2. Click the Modify Part Button on the Main Form
3. Click the Cancel Button

Expected results:

- The Modify Part Form closes, and the application returns to the Main Form

F1: Add Product Form Functionality Tests

All tests must pass for this aspect to pass

Preconditions for all Tests

- The application needs to compile
- The application needs to run
- The application needs to open to the Add Part Form
- The Main Form GUI aspect (aspect A1) must be passing
- The Add Product Form GUI aspect (aspect A4) must be passing
- The Modify Product Form GUI aspect (aspect A5) must be passing
- A Part needs to either be entered by the evaluator or a Part needs to be present when the application opens
- The classes (aspect C) must be passing

Special Notes

a) This aspect does not test exception controls
b) All tests for this aspect will use valid data
c) Evaluators may use different values in the test (Red Text)
d) Inventory, min, and max will always be tested as whole numbers
e) Price will always be tested as a decimal number
f) Search can be by ID or Name

Test 1 (Product ID)

Steps:

1. **Click the Add Product Button on the Main Form**
2. **Inspect the Product ID textbox**

Expected results:

- **The TextBox is disabled but shows the part ID**

Test 2 (Adding a Part to a Product)

Steps:

1. **Click the Add Product Button from the Main Form**
2. **Enter "Bike" into the Product Name TextBox**
3. **Enter "7" into the Inventory TextBox**
4. **Enter "125.99" into the Price TextBox**
5. **Enter "10" into the Max TextBox**
6. **Enter "1" into the Min TextBox**
7. **Select a part from the Parts Table (top table)**
8. **Click the Add Part**
9. **Click Save**
10. **Select the Product you just added from the Products Table on the Main Form**
11. **Click the Modify Product Button**
12. **Check the values on the Modify Product form against the values entered**

Expected results:

- The TextBoxes allow the data to be entered
- No error messages or crashes happen when the Save Button is clicked
- When the Add Button is clicked, the Part from the top table is added to the bottom table
- Once the Save Button is clicked the application returns the user to the Main Form
- The new product is populated into the Products Table on the Main Form
- The data inside the Modify Product Form matches the values entered

Test 3 (Search Part 1)

Steps:

1. Click the Add Product Button from the Main Form
2. Enter text that matches a Part or ID from the top table into the Search TextBox
3. Press Search Button (If used)
4. Clear the text from the Search TextBox
5. Press Search Button (if used)

Expected results:

- Part or ID matching text is highlighted

OR

- Only the Part or ID matching the text is shown in the table

OR

- Parts or ID with partial text that matches the text is shown in the table

After step 4:

- All the Parts are returned to the table

Test 4 (Search Part 2)

Steps:

1. Click the Add Product Button from the Main Form
2. Enter text that does not matches a Part or ID from the top table into the Search TextBox
3. Press Search Button (If used)

Expected results:

- The application displays an error message in the UI or in a dialog box with a message that alerts the user that the Part could not be found

Test 5 (Remove)

Steps:

1. Click the Add Product Button from the Main Form
2. Select a Part from the Applied Parts Table (bottom table)
3. Click Remove Part

Expected results:

- The selected Part should be removed from the bottom table

Test 6 (Cancel)

Steps:

1. **Click the Add Product Button from the Main Form**
2. **Click Cancel**

Expected results:

- **The application should close the Add Product form and sent the user to the Main Form**

F2: Modify Product Form Functionality Tests

All tests must pass for this aspect to pass

Preconditions for all Tests

- The application needs to compile
- The application needs to run
- The application needs to open to the Add Part Form
- The Main Form GUI aspect (aspect A1) must be passing
- The Add Product Form GUI aspect (aspect A4) must be passing
- The Modify Product Form GUI aspect (aspect A5) must be passing
- A Part needs to either be entered by the evaluator or a Part needs to be present when the application opens
- A Product needs to either be entered by the evaluator or a Product needs to be present when the application opens
- The classes (aspect C) must be passing

Special Notes

a) This aspect does not test exception controls

b) All tests for this aspect will use valid data

c) Evaluators may use different values in the test (Red Text)

d) Inventory, min, and max will always be tested as whole numbers

e) Price will always be tested as a decimal number

f) Search can be by ID or Name

Test 1 (Product ID and Data Population)

Steps:

1. **Select a Product from the Products Table on the Main Form**
2. **Click the Modify Part Button on the Main from**
3. **Inspect the Product ID Textbox**
4. **Inspect the data fields**
5. **Inspect the bottom table**

Expected results:

- **The TextBox is disabled but shows the part ID**
- **The data fields be populated with the correct data**

Test 2 (Modify a Product)

Steps:

1. Select a Product from the Products Table on the Main Form
2. Click the Modify Part Button on the Main from
3. Enter "Car" into the Product Name TextBox
4. Enter "4" into the Inventory TextBox
5. Enter "90.55" into the Price TextBox
6. Enter "20" into the Max TextBox
7. Enter "2" into the Min TextBox
8. Select "Brake" from the Parts Table (bottom table)
9. Click the Add Part Button
10. Click the Save Button
11. Select the Product you just added from the Products Table on the Main Form
12. Click the Modify Part Button on the Main from
13. Inspect the data fields

Expected results:

- The TextBoxes allow the data to be entered
- No error messages or crashes happen when the Save Button is clicked
- Once the Save Button is clicked the application returns the user to the Main Form
- The updated product is populated into the products table on the Main Form
- In step 13 the data should be the same as the data entered in the previous steps

Test 3 (Search Part 1)

Steps:

1. Select a Product from the Products Table on the Main Form
2. Click the Modify Part Button on the Main Form
3. Enter text that matches a Part or ID from the top table into the Search TextBox
4. Press Search Button (If used)
5. Clear the text from the Search TextBox
6. Press Search Button (if used)

Expected results:

- Part or ID matching text is highlighted

OR

- Only the part or ID matching the text is shown in the table

OR

- Parts or IDs with partial text that matches the text is shown in the table

After step 4:

- All the parts are returned to the table

Test 5(Search Part 2)

Steps:

1. Select a Product from the Products Table on the Main Form
2. Click the Modify Part Button on the Main Form
3. Enter text that does not matches a Part from the top table into the Search TextBox
4. Press Search Button (If used)

Expected results:

- **The application displays an error message in the UI or in a dialog box with a message that alerts the user that the Part could not be found**

Test 5 (Remove)

Steps:

1. **Select a Product from the Products Table on the Main Form**
2. **Click the Modify Part Button on the Main from**
3. **Select a Part from the Applied Parts Table (bottom table)**
4. **Click Remove Part**

Expected results:

- **The selected Part should be removed from the bottom table**

Test 6 (Cancel)

Steps:

1. **Select a Product from the Products Table on the Main Form**
2. **Click the Modify Part Button on the Main from**
3. **Click Cancel**

Expected results:

- **The application should close the Modify Product Form and send the user to the Main Form**

G: Input Validation and Logical Errors Tests

All tests must pass for this aspect to pass

Preconditions for all Tests

- The application needs to compile
- The application needs to run
- The application needs to open to the Add Part Form
- The Main Form GUI aspect (aspect A1) must be passing
- The Add Part Form GUI aspect (aspect A2) must be passing
- The Modify Part Form GUI aspect (aspect A3) must be passing
- The Add Product Form GUI aspect (aspect A4) must be passing
- The Modify Product Form GUI aspect (aspect A5) must be passing
- The Add Part Functionality (aspect E1) must be passing
- The Add Product Functionality (aspect F1) must be passing
- The classes (aspect C) must be passing

Special Notes

a) Evaluators may use different values in the test (Red Text)

b) Inventory, min, max and machine ID will always be tested as whole numbers

c) Price will always be tested as a decimal number

Test 1 (Add Part Exception Controls)

Steps:

1. **Click the Add Part Button on the Main Form**
2. **Select "In-House" from the Radio Buttons**
3. **Enter "Wheel" into the Part Name TextBox**
4. **Enter "5" into the Inventory TextBox**
5. **Enter "2.99" into the Price TextBox**
6. **Enter "1" into the Max TextBox**
7. **Enter "10" into the Min TextBox**
8. **Press the Save Button**
9. **Enter "10" into the Max TextBox**
10. **Enter "1" into the Min TextBox**
11. **Enter "15" into the Inventory TextBox**
12. **Press the Save Button**

Expected results:

- **On step 8 an error message should come up alerting the user that the min is higher than the max**
- **On step 12 an error message should come up alerting the user that the inventory is outside the min/max range**
- **No parts were saved**

Test 2 (Modify Part Exception Controls)

Steps:

1. Select a Part from the Part table on the Main Form
2. Click the Modify Part Button on the Main From
3. Enter "12" into the Max TextBox
4. Enter "11" into the Min TextBox
5. Press the Save Button
6. Enter "10" into the Max TextBox
7. Enter "1" into the Min TextBox
8. Enter "25" into the Inventory TextBox
9. Press the Save Button

Expected results:

- On step 5 an error message should come up alerting the user that the min is higher than the max
- On step 9 an error message should come up alerting the user that the inventory is outside the min/max range
- No parts were saved

Test 3 (Add Product Exception Controls)

Steps:

1. Click Add Product from the Main Form
2. Enter "Home" into the Product Name TextBox
3. Enter "3" into the Inventory TextBox
4. Enter "15.99" into the Price TextBox
5. Enter "120" into the Max TextBox
6. Enter "200" into the Min TextBox
7. Select a Part from the Parts Table (top table)
8. Click the Add Part Button
9. Click the Save Button
10. Enter "10" into the Max TextBox
11. Enter "2" into the Min TextBox
12. Enter "30" into the Inventory TextBox
13. Click Save

Expected results:

- On step 9 an error message should come up alerting the user that the min is higher than the max
- On step 13 an error message should come up alerting the user that the inventory is outside the min/max range
- No products were saved

Test 4 (Modify Product Exception Controls)

Steps:

1. Select a Product from the Product Table on the Main Form
2. Click the Modify Product Button from the Main Form
3. Enter "1" into the Max TextBox

4. Enter “300” into the Min TextBox
5. Click the Save Button
6. Enter “10” into the Max TextBox
7. Enter “1” into the Min TextBox
8. Enter “320” into the Inventory TextBox
9. Click the Save Button

Expected results:

- On step 5 an error message should come up alerting the user that the min is higher than the max
- On step 9 an error message should come up alerting the user that the inventory is outside the min/max range
- No products were saved

Test 5 (Delete Part Exception Controls)

Steps:

1. Select a Part from the Part Table on the Main Form
2. Click the Delete Button

Expected results:

- A message should pop up asking the user to confirm

If the user clicks Confirm (Yes)

- The Part is deleted, and the user is taken back to the Main Form

If the user clicks Cancel (no)

- The user is taken back to the Main Form without the Part being deleted

Test 6 (Delete Product Exception Controls)

Steps:

1. Select a Product from the Product table on the Main form
2. Click Delete

Expected results:

If the Product has a Part assigned to it

- A message should pop up stating the product can’t be deleted because a part is assigned to it

If the Product does not have a Part assigned to it

- A message should pop up asking the user to confirm

If the user clicks Confirm (Yes)

- The Product is deleted, and the user is taken back to the Main Form

If the user clicks Cancel (no)

- The user is taken back to the Main Form without the Product being deleted

Test 7 (Remove Part Exception Controls)

Steps:

1. **Select a Product from the Product table on the Main Form**
2. **Click the Modify Part Button**
3. **Select a Part from the top table**
4. **Click the Add Button**
5. **Select the Part you just added from the bottom table**
6. **Click the Remove Button**

Expected results:

- **A message should pop up asking the user to confirm**

If the user clicks Confirm (Yes)

- **The Part is removed from the bottom table**

If the user clicks Cancel (no)

- **Nothing happens and the Part is still on the bottom table**

Test 8 (Add Part Valid Data Exception Controls)

Steps:

1. **Select Add Part Button from the Main Form**
2. **Select the In-House Radio Button**
3. **Enter "Light" in the name TextBox**
4. **Enter "8.99" in the price TextBox**
5. **Enter "4" in the Inventory TextBox**
6. **Enter "10" in the Max TextBox**
7. **Enter "1" in the Min TextBox**
8. **Enter "One" in the In-House TextBox**
9. **Click the Save Button**
10. **Enter "1" in the In-House TextBox**
11. **Enter "text" in the price TextBox**
12. **Click the Save Button**
13. **Enter "1.99" in the price TextBox**
14. **Enter "Cat" in the Inventory TextBox**
15. **Click the Save Button**
16. **Enter "5" in the Inventory TextBox**
17. **Enter "10times" in the Max TextBox**
18. **Click the Save Button**
19. **Enter "10" in the Max TextBox**
20. **Remove the text from the Min TextBox**
21. **Click the Save Button**

Expected results:

- **An error message should show up every time the Save Button is clicked that alerts the user to the data entry error**

- The text in the error message needs to alert the user as to where the data entry error is and what it is (Must be descriptive)

Test 9 (Modify Part Valid Data Exception Controls)

Steps:

1. Select a Part from the Parts Table on the Main Form
2. Click the Modify Part Button from the Main Form
3. Select the In-House Radio Button
4. Enter "MyPlace" in the In-House TextBox
5. Click the Save Button
6. Enter "1" in the In-House TextBox
7. Enter "text" in the price TextBox
8. Click the Save Button
9. Enter "1.99" in the price TextBox
10. Enter "Cat" in the Inventory TextBox
11. Click the Save Button
12. Enter "5" in the Inventory TextBox
13. Enter "10times" in the Max TextBox
14. Click the Save Button
15. Enter "10" in the Max TextBox
16. Remove the text from the Min TextBox
17. Click the Save Button

Expected results:

- An error message should show up every time the Save Button is clicked that alerts the user to the data entry error
- The text in the error message needs to alert the user as to where the data entry error is and what it is (Must be descriptive)

Test 10 (Add Product Valid Data Exception Controls)

Steps:

1. Click the Add Product Button from the Main Form
2. Enter "Headphones" in the Name TextBox
3. Enter "Too Much" in the Price TextBox
4. Enter "4" in the Inventory TextBox
5. Enter "10" in the Max TextBox
6. Enter "1" in the Min TextBox
7. Click the Save Button
8. Enter "1.99" in the Price TextBox
9. Enter "Out of Stock" in the Inventory TextBox
10. Click the Save Button
11. Enter "5" in the Inventory TextBox
12. Enter "x33" in the Max TextBox
13. Click the Save Button
14. Enter "10" in the Max TextBox
15. Remove the text from the Min TextBox

16. Click the Save Button

Expected results:

- An error message should show up every time the Save Button is clicked that alerts the user to the data entry error
- The text in the error message needs to alert the user as to where the data entry error is and what it is (Must be descriptive)

Test 11 (Modify Product Valid Data Exception Controls)

Steps:

1. Select a Product from the Products Table on the Main Form
2. Click the Modify Product Button from the Main Form
3. Enter “Slow” in the price TextBox
4. Click the Save Button
5. Enter “1.99” in the price TextBox
6. Enter “Low” in the Inventory TextBox
7. Click the Save Button
8. Enter “5” in the Inventory TextBox
9. Enter “x33+88+y” in the Max TextBox
10. Click the Save Button
11. Enter “10” in the Max TextBox
12. Remove the text from the Min TextBox
13. Click the Save Button

Expected results:

- An error message should show up every time the Save Button is clicked that alerts the user to the data entry error
- The text in the error message needs to alert the user as to where the data entry error is and what it is (Must be descriptive)

H: JavaDoc Folder Tests

All tests must pass for this aspect to pass

Preconditions for all Tests

- None

Special Notes

**None*

Test 1 (Folder)
Steps: **18. Look at the project for JavaDoc folder**
Expected results: • **The JavaDocs must be inside one folder (Sub-folders are allowed within that folder)**

I: Professional Communications

No tests for this section. However, if a submission is messy, or the code is chaotic, or the GUI is sloppy, this section may be marked as needs revision. This section is reserved for projects that appear to be slapped together. All other aspects may pass but if the project is a mess, you may be asked to clean it up and resubmit.