INTRODUCTION

Throughout your career in software design and development, you will be asked to create applications with various features and functionality based on business requirements. When a new system is developed, typically the process begins with a business analyst gathering and writing these business requirements, with the assistance of subject matter experts from the business. Then a system analyst works with several application team members and others to formulate a solution based on the requirements. As a developer, you would then create a design document from the solution and finally develop the system based on your design document.  
  
For this assessment, you will create a C# application using the solution statements provided in the requirements section.  
  
The skills you showcase in your completed application will be useful in responding to technical interview questions for future employment. This application may also be added to your portfolio to show to future employers.  
  
Your submission should include a zip file with all the necessary code files to compile, support, and run your application.  
  
**Note: The preferred integrated development environment (IDE) for this assignment is Visual Studio. Refer to your course of study for instructions on how to install and use this application. If you choose to use another IDE, you must export your project into Visual Studio format for submission.**  
  
**Your submission should include a zip file with all the necessary code files to compile, support, and run your application. The zip file submission must also keep the project file and folder structure intact for the Visual Studio IDE.**

SCENARIO

You are working for a small manufacturing organization that has outgrown its current inventory system. They have been using a spreadsheet program to manually enter inventory additions, deletions, and other data from a paper-based system but would now like you to develop a more sophisticated inventory program.  
  
They have provided you with a mock-up of the user interface to use in the design and development of the system (see the attached “GUI Mock-Up”) and a class diagram to assist you in your work (see the attached “UML Class Diagram”). The organization also has specific business requirements that must be included as part of the application. A system analyst from your company created the solution statements outlined in the requirements section based on the manufacturing organization’s business requirements. You will use these solution statements to develop your application.

REQUIREMENTS

***I. User Interface***

Create a C# application with a graphical user interface (GUI) based on the attached “GUI Mock-Up.” Write code to display each of the following forms in the GUI:

A.  A main form, 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 form, showing the following controls:

•  radio buttons for “In-House” and “Outsourced” parts

•  buttons for “Save” and “Cancel”

•  text boxes 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 form, with fields that populate with data from an existing Part, showing the following controls:

•  radio buttons for “In-House” and “Outsourced” parts

•  buttons for “Save” and “Cancel”

•  text boxes 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 form, showing the following controls:

•  buttons for “Save,” “Cancel,” “Add” part, and “Delete” part

•  text boxes 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 grid view for all parts

•  a grid view for parts associated with the product

•  a “Search” button and a text field with an associated list for displaying the results of the search

E.  A modify product form, with fields that populate with data from an existing product, showing the following controls:

•  buttons for “Save,” “Cancel,” “Add” part, and “Delete” part

•  text boxes 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 “all candidate parts”

•  a grid view for parts associated with the product

•  a “Search” button and a text box 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 all associated properties

•  variables are accessible/modifiable through properties

G.  Add the following functionalities to the main form, using the methods provided in the attached “UML Class Diagram”:

•  redirect the user to the “Add Part,” “Modify Part,” “Add Product,” or “Modify Product” forms

•  delete a selected part or product from the grid view

•  search for a part or product and display matching results

•  exit the main form

H.  Add the following functionalities to the part forms, using the methods provided in the attached “UML Class Diagram”:

1.   “Add Part” form

•  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 form

•  cancel or exit out of this form and go back to the main form

2.   “Modify Part” form

•  select “In-House” or “Outsourced”

•  modify or change data values

•  save modifications to the data and then redirect to the main form

•  cancel or exit out of this form and go back to the main form

I.  Add the following functionalities to the product forms, using the methods provided in the attached “UML Class Diagram”:

1.   “Add Product” form

•  enter name, inventory level, price, and max and min values

•  save the data and then redirect to the main form

•  associate one or more parts with a product

•  remove or disassociate a part from a product

•  cancel or exit out of this form and go back to the main form

2.   “Modify Product” form

•  modify or change data values

•  save modifications to the data and then redirect to the main form

•  associate one or more parts with a product

•  remove or disassociate a part from a product

•  cancel or exit out of this form and go back to the main form

J.  Write code to address the following conditions with exception handling code:

•  Detect non-numeric values in textboxes that expect numeric values

•  Min should be less than Max; and Inv should be between those two values

•  Prevent the user from deleting a product that has a Part associated with it

•  Confirm “Delete” actions