California State Polytechnic University, Pomona
Computer Information Systems Department

Client/Server Programming with Visual Basic

Course: **CIS 338**
Instructor: **Hatim Mouissa**
Email: **hmouissa@cpp.edu**

Program Two

I. DUE DATE

See class schedule. The written part is due at the beginning of class no later than 8:15PM, the electronic part has to be sent to the instructor using Blackboard

II. OBJECTIVES

1. To learn how to break code into smaller more manageable procedures and functions
2. To learn how to use arrays / lists / collections to pass and retrieve data
3. To learn how to use exception handling to create error tolerant code
4. To learn how to use classes and OOP to separate Logic from the UI
5. To learn how to write to files
6. To learn how to code for MDI menus

III. REFERENCES

1. Textbook lecture chapters
2. Course Powerpoints and examples

IV. SOFTWARE REQUIRED

1. Visual Studio 2012
2. Blackboard.

V. ASSIGNMENT

Purpose

a) To learn how to design, write, and implement a Visual Basic program using a two-tier architecture. Your projects will be comprised of objects associated with both a graphical user interface and a business model.

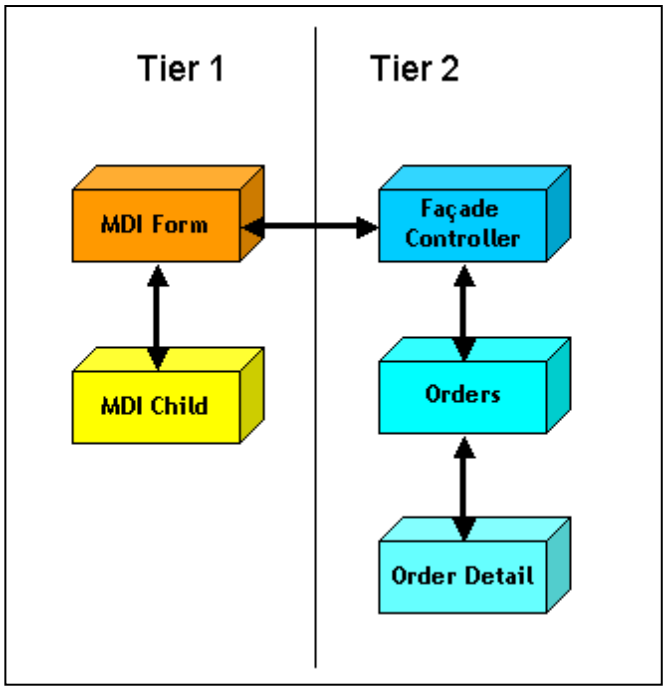
Problem

a) Enter, process, save, and retrieve multiple customer orders for the **HALOSTORE** project. All information must be saved and retrieved using classes associated with a business model. All business logic, including data validation, must occur within the corresponding business object.

b) The user interface will be connected to a business model that you will build and implement using Visual Basic class modules.

Program Design

a) This project consists of the following principal components. The Tier 1 layer uses (1) frmMain as the MDI parent and (2) frmOrder, as the order entry form. The Tier 2 layer uses (3) clsController, serving as the gateway/facade to the business model, (4) clsOrder, used to hold the general order information, (5) clsOrderDetail, used to hold the details for each type of item ordered.



b) As shown in the accompanying figure, this project has only **one connection** between the user interface and the business model. In this case, the connection will be implemented in the MDI parent form. Each new child form automatically links to this connection

MDI Form

a) Use the skeleton code that is provided, modify the project MDI File menu as follows:

1. New: Opens a new order form
2. Open: shows an input box prompting for the order number to open
3. Close: closes the current order form
4. Save: saves the current order form
5. Delete: deletes the current order
6. Exit.: exits the program

The buttons on the order form will do the following

7. Open: loads the order that corresponds to the number entered in the order form Order ID field
8. Delete: deletes the current order
9. Clear: will reset the form to its initial state by restoring all the defaults
10. Save: will save the current order

b) The following validation should be in place

- Order ID must be an integer value that can only increment in 100

Client/Server Programming with Visual Basic

Course: **CIS 338**
Instructor: **Hatim Mouissa**
Email: **hmouissa@cpp.edu**

- Customer Name cannot be blank
- Customer Phone has to be 10 digits
- Order date must be a valid date
- Quantity ordered must be an integer value greater than or equal to one.
- Price must be from the menu and cannot be entered by the user.
- Billing and shipping info cannot be blank depending on which option is selected
- All validation has to be performed by the appropriate business class
- Sales Tax is 7.825%.

c) For saving duplicate orders, provide a message box that asks for permission to overwrite the old values – include buttons for Yes and No (default).

d) Delete will remove the corresponding order from the collection and reset the Order form to a new order state.

Controller/Facade Class

Build a controller class (clsController) to function as the gateway between your user interface and the remainder of your business model. This class is responsible for managing all orders. Include functionality to retrieve, add, change, and delete orders as discussed in class.

Order Class

Build an Order class (clsOrder) to store, retrieve, and manage all information regarding the order, including order details (see below). All order data must be saved as private variables. Use public property variables, sub procedures, and functions, as necessary to access this data. This class will be responsible for validation its own data elements.

Order Detail Class

Build an Order Detail class (clsOrderDetail) to store, retrieve, and manage all information regarding the details for each type of item ordered. All order details must be saved as private variables. Use public property variables, sub procedures, and functions, as necessary to access this data. This class will be responsible for validating its own data elements.

Client/Server Programming with Visual Basic

Course: **CIS 338**
Instructor: **Hatim Mouissa**
Email: **hmouissa@cpp.edu**

VII. DELIVERABLES

1. Electronic:
   Name your project **xxxprogram2**, where xxx are your initials taken from the first three characters of your Cal Poly email address.

   Zip all the contents of your program folder and the word document listed in the next section and name it **xxxprogram2.zip**

   Your zip-file must be transferred via blackboard to your instructor. The zip-file must be sent by the beginning of class. After 8:15 PM on the due date the program is late and **will not be accepted** so make sure you turn-in whatever you have on time to get partial credit**.** (Do not wait until the last minute to upload your project). The clock on blackboard site is the official clock.

2. Word document (IN THE FOLLOWING ORDER)

   a. Use the Microsoft Word template provided with the assignment
   b. Add a table of content after the first page
   c. Add a completed Test Data table (Similar to the one included in the handout)
   d. Add Print Screens of your test inputs and outputs
   e. Program design (UML):
      i. Add Class diagram showing all instance variables, methods, procedures,functions, etc.
   f. Add a sheet Titled extra credit listing the extra credit (if any) that was done

VII. EXTRA CREDIT

- 5 points – Create a user-defined control and use it. The control will replace the detail line total labels and changes color from the default white to yellow when the total per line is greater than $20.00 and change back to white when the total per line is less than $20.00.
- 10 points – Load and Unload your controls dynamically for the order detail items. Add a button titled “Add Line” and a button titled “Remove Line”, when clicked the button adds a new detail line item to your panel (the detail line should have all your controls, line label / item combobox / qty textbox / price label / total label). The dynamic control should handle all

appropriate events. Only allow a line to be removed if it is blank of data entry.

Client/Server Programming with Visual Basic

Course: **CIS 338**
Instructor: **Hatim Mouissa**
Email: **hmouissa@cpp.edu**

VI. TEST DATA

Test No.	Item	Planned Result	Agree (Y/N)
1	Create and save Order No. 100 for 2 Banshees and 3.1 Ghosts	Error Invalid quantities.	
2	Change quantity to 3 Ghosts instead of 3.1 and save Order No. 100	Line total, sales tax, and order total are shown correctly. Order is saved.	
3	Close Order No. 100 and retrieve it again.	All values are shown correctly.	
4	Create and save Order No. 200 for quantity of 5 Hornets	Line total, sales tax, and order total are shown correctly. Order is saved.	
5	Close all orders.	Orders are closed.	
6	Open Order No.100 and Order No. 200 at the same time.	Orders display correctly.	
7	Open Order No. 300.	Error Order not found.	
8	Create and save Order No. 300 for 2 Choppers (in line 1 of the detail) and a quantity of 5 on the next line (line 2) with no item selected.	Error Missing item selection.	
9	Create and save Order No. 300 for Hornets and a quantity of 5 Choppers	Line total, sales tax, and order total are shown correctly. Order is saved.	
10	Delete Order No. 200.	Order is deleted. Form is reset to initial state.	
11	Delete Order No. 400	Error Order does not exist.	
12	Update Order No.100. Change the quantity for Cappuccino to 5.1.	Error Invalid qty and price.	
13	Update Order No.100. Change the quantity for Banshee to 5	Replace existing order? No Order is not saved. Replace existing order? Yes Order is saved.	
14	Open a blank form.	Form is reset. New date / time is shown.	

California State Polytechnic University, Pomona
Computer Information Systems Department

Client/Server Programming with Visual Basic

Course: **CIS 338**
Instructor: **Hatim Mouissa**
Email: **hmouissa@cpp.edu**

Name _____
(last name, first name)

Bronco #_____

Program Number TWO

Professionalism and following directions
o Assignment is neatly assembled 8 1/2 by 11.
o Project and Form and Zip file names are correct.
o Tabs between controls is in correct order
o Hot keys in buttons and menus are correct
o Test data screen prints

UML Diagrams
o Class diagram

Source Code
o Program source code listing matches code sent by Blackboard's digital dropbox.
o Multi-tier design / Classes properly used
o Code properly commented
o Code uses proper naming convention for variables and controls
o Indentation (formatting) is proper.
o Appropriate controls are used and their properties are set correctly

Electronic Run
o Program asks for all inputs
o All answers/outputs are present.
o All answers/outputs are correct.

Total points _____ **Xtra Credit** _____
Professor Comments:

Statement of Independent Effort:

I _____, attest that this entire project is my own work.

Signature _____