Capstone Comments

1. All code should be protected with exception handling. Minimum is Main where the application begins execution. Also, it would be best in all event handlers where the ability to communicate with the user is still possible. Below is a recommended approach that works to capture exceptions in Main:

Create your own Main method to startup your WPF application. Add a class to your project and enter the code as shown below within the StartUpClass. Enhance the exception handling to catch errors and all of their info. Then open the Properties window for your project => Application tab => set the Startup Object to your new class with Main. Now you have a way to capture any exceptions raised that are not caught anywhere in your application.

class StartUpClass

{

[System.STAThreadAttribute()]

static void Main()

{

try

{

App.Main();

}

catch

{

}

}

}

1. Should use recursion to capture all exceptions, even if your app does not currently throw a new exception in a catch block that has the caught exception in the InnerException property. This recursive method could be called by all catch blocks. Yes, executing the ToString() method on an exception will show all info, but that is not in a usable format for storing info. Using the ToString() method is not appropriate for a real world application – only for testing purposes.
2. Make sure you have adequate comments throughout. The idea of regionalizing your code is great. However, if there are areas where additional comments would be helpful, that would also be strongly suggested.
3. Make sure all controls where data entry of any kind is being made, has appropriate labels that identify what the entry control is for. I noticed this mostly with dropdown controls.
4. It would be nice to display phone numbers in a phone number format: (nnn) nnn-nnnn.
5. Make sure all labels are showing completely.