```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;

namespace QuiddichV2._0
{
    public static class Validator
    {
        private static string title = "Entry Error";

        public static string Title
        {
            get
            {
                return title;
            }
            set
            {
                title = value;
            }
        }

        public static bool IsPresent(TextBox textBox)
        {
            if (textBox.Text == "")
            {
                MessageBox.Show(textBox.Tag + " is a required field.", Title);
                textBox.Focus();
                return false;
            }
            return true;
        }

        public static bool IsPresent(PasswordBox passwordBox)
        {
            if (passwordBox.Password == "")
            {
                MessageBox.Show(passwordBox.Tag + " is a required field.", Title);
                passwordBox.Focus();
                return false;
            }
            return true;
        }

        public static bool IsDecimal(TextBox textBox)
        {
            decimal number = 0m;
            if (Decimal.TryParse(textBox.Text, out number))
            {
                return true;
            }
            else
            {
                MessageBox.Show(textBox.Tag + " must be a decimal value.", Title);
                textBox.Focus();
                return false;
            }
        }

        // The IsInt32 and IsWithinRange methods were omitted from figure 12-15.
        public static bool IsInt32(TextBox textBox)
        {
            int number = 0;
            if (Int32.TryParse(textBox.Text, out number))
            {
                return true;
            }
            else
            {
                MessageBox.Show(textBox.Tag + " must be an integer.", Title);
                textBox.Focus();
                return false;
            }
        }

        public static bool IsWithinRange(TextBox textBox, decimal min, decimal max)
        {
            decimal number = Convert.ToDecimal(textBox.Text);
            if (number < min || number > max)
            {
                MessageBox.Show(textBox.Tag + " must be between " + min
                    + " and " + max + ".", Title);
                textBox.Focus();
                return false;
            }
            return true;
        }

        public static bool IsValidZip(TextBox textBox)
        {
            bool valid = false;

            string pattern = @"^\d{5}(-\d{4})?$";
            Regex reg = new Regex(pattern);

            if(reg.Match(textBox.Text).Success)
            {
                valid = true;
            }
            else
            {
                MessageBox.Show(textBox.Tag + "is not a valid zip code." +
                    "\n Valid zip codes are in the format XXXXX or XXXXX-XXXX",
                    "Invalid Zip Code", MessageBoxButton.OK, MessageBoxImage.Error);
            }

            return valid;
        }
    }
}

```