The Problem

Included is a Visual Studio solution for a WPF application that allows the user to create a list of triangles, calculate the area of their triangle list, and submit the area to a remote server. The feedback has been good, but the users have requested support for more shapes: circles, squares, and rectangles.

Your task is to use object-oriented principles and test driven development to refactor the existing code and add new code to support these new shape additions. You are free to modify, rename, add, or remove any of the code in the solution to accomplish this objective. The goal is to solve the problem using elegant code and modern development techniques.

Acceptance Criteria

- The user should be able to add circles to the list, providing a name and radius of the circle
- The user should be able to add squares to the list, providing a name and side length
- The user should be able to add a rectangle to the list, providing a name, length, and width
- Total area should compute the total area of all shapes that have been added to the list
- The submissions service is unreliable and sometimes fails. The application should notify the user of whether submission was successful or not
- The application should remain responsive while transmitting total area to the submission service- it should not lock up
- Bonus Points: Enable the “Add” button to find any new shapes added to the code without having to modify the code for the “Add” button every time a shape is added. For example: I implement a rhombus and it is automatically supported by the “Add” button purely by implementing the code for the shape

Other Requirements

- All of your code (with the exception of XAML) should be covered with good unit tests that properly validate the expected behavior. There are some examples of good tests that use mocking in the solution
- Do not modify or change the SubmissionService dll