GUI Programming

Retake Exam

Create a WPF application, which allows the user to assemble hiking trails from waypoints and save them. The application should use **data binding** for both data (listbox, textbox, etc) and user events (**commands**) and it should follow the general **MVVM** principles. All application logic should be handled by viewmodels (except converters), the code-behind files should not contain any custom code.

1. There should be two extra classes: **Waypoint**{ PointCode, X, Y, Height, Category } and **Hike** { DateOfHike, ListOfPoints, Difficulty }. **(1 points)**

   Create a TXT file with some Waypoints, the list of waypoints should be initialized from this file in the constructor of the main window. (X and Y are coordinates in an imaginary coordinate system, measured in km) **(2 points)**

2. The waypoints should be displayed in the first **ListBox** control, using an **ItemTemplate** that shows all data **side by side**. **(3 points)**

3. The second ListBox should display the list of waypoints selected for a hike, using an **ItemTemplate** that shows the waypoint's code and its coordinates. **(2 points)**

4. Use a dialog window to edit the first ListBox (add/edit waypoints)! If no items are selected in the ListBox, the window should open with empty values and should add a new waypoint to the list when the dialog is closed. **(3 points)**

   The PointCode can only accept capital letters and numbers, while the point's category should be a **drop-down** list with the values - [easy, medium, hard] **(2 points)**

   If an item is selected, that item's values should be bound to the controls in the window and the item should be updated when the window is closed. **(2 points)**

5. Now we can assemble the hikes: one hike is represented by the second ListBox control, there should be two Buttons that move items in any direction between the first and second ListBox controls. **(1 points)**

   There is a special rule here, which mandates that between two waypoints, whose category is hard, there should be at least three waypoints, which are either medium or easy. If the user selects a waypoint that does not follow this rule, the moving button should become disabled. **(1 points)**

6. There should be a Button that allows the user to save a hike into the third listbox. Next to the button there should be a textbox control for the date of the hike. **(1 points)**

   The button should be disabled as long as this textbox is empty. **(1 points)**

   Once the button is clicked, the new hike should appear in the third listbox and the second listbox's value should be cleared (All elements should be removed). **(1 points)**

The third ListBox should display the list of assembled hikes using another **ItemTemplate**, which shows the **DateOfHike** and the **ListOfPoints** converted into a string side by side. For this task, you'll need to create a converter (ListToStringConverter), which is capable of turning a list objects into a string. **(5 points)**

This **ItemTemplate** should also show the **Difficulty** of the hike using a **slider** control, which should be under the DateOfHike and ListOfWaypoints in the template. To calculate the difficulty of a hike, you'll need to find the waypoints with the smallest height and the tallest height, subtract the smallest from the tallest and multiply it by the number of waypoints. The max value of the slider should be a 1000. **(5 points)**