ST2API – TP#3: Expand on TP#2

I hope you used S.O.L.I.D. principles, clean code, clean architecture… Now we're getting into the core of this class. You will have two weeks to expand and extend on your TP#2, with the deadline being Saturday 9th midnight. As per usual, I will be expecting a git link to a functional C# project. Questions are to be answer either in a *.pdf file on git, or in the readme.

Questions:

- In your point of view, what was complicated in the TP#2? What where your pain points? What could you do when expanding this TP to avoid these further?
- What are the S.O.L.I.D. principles? What are the "KISS rule" and the "boy scout rule"? What's clean code, clean architecture?
- How would you expand this code? What can you add to the software you build to make it more useful, reliable, relevant to the user? Find 5 ways to improve your previous TP, even if that's not doable.

Coding part:

- Give the user a choice of cities around the world to get information from a chosen list (at least 10 major cities)
    - Weather: temperature in Celsius, wind, sun rise, and sun set in UTC… the more the better. Only relevant data though!
    - Tomorrow's weather: Hourly information on a few key hours
    - Next 5 days forecast

- Build a small, usable UI
    - City name, chosen coordinates
    - Basic information (see before)
    - Must not burn my retina (aka be readable)

- Clean Archi
    - Separated class, one by file
    - Each class must have a single responsibility
    - Separated projects for each major component (UI, core, unit tests…)

- Unit tests
    - Using your preferred testing framework, have a few tests
        - MSTest, Nunit, MoQ are good exemples
    - Don't test the GUI, but test how your code reacts to data given to it.
        - Few relevant tests over 100% code coverage!

- Forbidden:
    - Wrappers in another language
    - Solutions that do everything for you
    - JObject, object, dynamic
    - Newton.Soft to serialize JSON is ok.

- Bonuses (in random order):
    - Weather icons (see on the website documentation)
    - Possibility to choose any point on the planet
    - Map to click on (aka google map)
    - Yearly/Monthly historical information on a coordinate (rain, temperature, sun)
    - France weather forecast with multiple towns at once (like on TV)
    - A standalone *.exe file
    - Any other idea will be marked as bonuses!

I will be available for any question / code demo to get feedback up until the deadline. I'll try to help you with either the code, the functionalities, the UI, the tests, or any other questions regarding the requirements.

Due to the heavy workload, I would advice against doing this at the very last minute…

Good luck !