Project Description

Create a fully functional simple web application that will be used to display the world countries, their capitals and the bordering countries.

1. Create a Visual Studio Web Application Solution – choose webforms + web api options. Add following pages and controls:
   * Default.aspx – main page that contains all controls. This page should be created by default at the time of the project creation.
   * Countries.ascx – user control to display list of countries
2. Implement Repository/Service pattern for data retrieval and manipulation.
   * Repository to performdata retrieval
     + URL for API is here <https://restcountries.com/v3.1/all> and for documentation is here <https://restcountries.com/#api-endpoints-v3>
   * Services to perform business logic data manipulation
     + Convert JSON data to your own model, retaining only minimum data necessary to satisfy the requirments (public api provides a lot more information than needed)
     + Filter data for a requested country: given country ID return country object that will contain list of neighbouring country names.
3. Add new web api controller that will be used to provide data to the webform with following methods:
   * /api/country GET : returns list of countries
   * /api/country/{id} GET : returns individual country object for a passed ID as int
4. Data should be cached, cached data should be used before attempting to call the public API, cached data should be refreshed periodically.
5. UI: application should allow user to select a country and upon selection display country’s capital, and bordering countries names. Use ajax (JQuery) to fetch country detailed information for a given country. Display selected country name and its neighbouring countries names in multiple langauges (german, spanish, french, japanese, italian – provided by the API). English is the default language. User should be able to select any other language, the user should always see the name of the country in English and one other langauge if selected.
6. Application should be localizable, i.e. can be adopted to be used in any new language – this is **not** to be confused with displaying the name of the country – in different langauges. For simplicity specify the lanaguage in the configuration file and provide translation for a couple of content strings.
7. Application should log errors, collect and log statistics which countries are being displayed, and public api calls (timestamp of the call, duration). The statistics should be stored on the hard disk. there should be only one stats file per day generated .
8. Application should email the critical errors as they happeed, and public api calls information daily. Email settings are to be defined in the configuration file.

NOTES:

* use only libraries provided by .NET framework (i.e no log4net).
* for simplicity keep everything in one project, but create subfolders for your services, repositories and any helpers
* the entire exercise should take less than 4 hours. If it takes you longer, you are over achieving.

**Deliverables**:

1. Design document for the application. Your document should have the following diagrams:  
   Component diagram to provide - overview of participating subsystems (Web Page, Cache, Service, Repository, External API, etc.). Provide UML action diagram for caching mechanism. Provide UML sequence diagram for the mechanism that returns country detailed information.
2. Visual Studio solution. This should include all files belonging to the solution!