IDSA – Inteligent Data Stock Analyzer.

## Developers :

1. CezaryWronka – data base behavior , task synchronization, chart’s
2. PiotrSacharuk – web data gathering, analyzing
3. PatrykJanota – csv reading, data presenting (main view), renta-wieczysta.

## Main Tasks (sorted with priority 1-5) TODO:

**All(1): CODE REFACTORING** - Spojnosc nazw: widok = to co widzimy bezposrednio w apps.[ Eventy – RaiseEventSTH] [Delegates - ViewName\_DelegateName] [Kolejnosc metod w klasach w sekcji Clean Code – **USE #REGION** **MANDATORY**]

Patryk(1): Obróbka danych przy DataGridzie(obliczenia)!!!!

Cezary(1): Reports (podzial w bazie na Bilans, RZiS, cala reszta)!!!!!

**Sth(1):** Implement using **Event Aggregator** in code

Cezary(2): Clean -> ServiceLocator -> dodac funkcje usuwania ze slownika. Clean bedzie szybszy, albo przerzucic się na **PRISM Service Locator**

Cezary(2): Spolki - nowe zakladki - new Tabbed View Provider(Internal) – zrobic tak, zeby nie bylo burdelu

Cezary(2): Chart Service Provider Interface(GetData(), Refresh(), Delegate do Company Changed?)

Cezary(2): Chart - Zaznaczasz kolumne, wyswietl wykres

Cezary(3): Dezaktywacja CsvUpload\_SaveDB

Cezary(3): Wykres funkcjonalnosc(wybierz max, ogranicz ilosc danych(raporty z 2 lat domyslnie), kolorystyka danych, ...)

Cezary(5): Add Companies/Reports w watkach(inny sposob blokowania aplikacji)

Cezary(5): Odpalenie bazy na innym watku.

Patryk(5): Filtr Box - Aktualizacja cofanie, jednorazowe wczytywanie z bazy(z poczatku, pozniej obiekty lokalnie - async)

Patryk(5): Transpozycja DataGrid View....

## Key Money MAKER features.

1. RentaWieczysta (Calculated)
2. RentaWieczysta (Stored ? where ? how to scan through ?)
3. Volumen Signal’s ( how to catch ? how to judge if is positive volume(positive information ) or negative(bad informations)
4. When quarter result comes out especially in hot season, there is a need for fast selection of companies that behave extra-ordinary. That with point (3) gives us possible investor’s money movment’s into this company, and other financial operators (for exampale TFI, OFE others)

## ADDITIONAL FEATURES.

1. Chart visualization on financial data when user select rows, columns etc.
2. Automatic download of data like SUROWCE , WALUTY etc.
3. **TODO: feel free to add more**

### Luźne przemyślenia:

Mamy 3 typy wyświetlania dla głównego View:

View = Reports DataGrid + chart

View = Company Info

View = Chart

## Clean Code:

Ordering of code in classes(ReSharper does this automatically;) ):

1. Static Variables
2. Static Methods
3. Public Variables
4. Protected Variables
5. Private Variables (3-5 Fields and Props)
6. Constructors (Ctors)
7. Public Methods (Event Handlers at top of public methods)
8. Protected Methods
9. Private Methods (Database Connection, etc.)

The following rules:

* static before anything
* variables before constructors before methods (i consider constructors to be in the category of methods)
* public before protected before private

The idea is that you **define the object** (the data), before the **behaviors** (methods). Statics need to be separated because they aren't really part of the object, nor it's behavior.

# Chart flow in IDSA (by Patric):

             I think more important is coneception about how to do it flexible, for all possible views, that we will get into the further. For now, is easy to get dataTable from datagrid, presenter of view can provide this action, although we must be flexible and be able to visualize the data that user want to see.

             -> user choose the data on the view (Ebit, sales) -> advanced(multiselection EBIT,SALES at once).

             -> user confirms he want the chart by (submenu (rightmouseclick)

             -> presenter get the event action (eventagregator from prism ?)

             -> presenter knows that the user select on view this data by the event,

             -> presenter lunch the ChartServiceProvider (DataInput.)

             -> chart service provider do all the data stuff

             -> presenter informs view that chart is ready to display

             -> view (shows) the chart.