IDSA – Intelligent Data Stock Analyzer.

|  |  |  |
| --- | --- | --- |
| **Ver.** | **Description** | **Mod. Date** |
| 1.0 | Initial document. | 2013 – 05 - 31 |
| 1.01 | Critical update Task List. | 2013 – 06 - 24 |
| 1.02 | Changing developers tasks. Gather info about app architecture/structure. | 2013 – 10 - 11 |
| 1.0X | TODO: Detail and update actual app state in Task List. | DATE |
|  |  |  |

## Developers :

1. CezaryWronka – DataBase based on Entity Framework, task synchronization, chart’s, PAP parser, PDF parser, On-line Server Reports Crawler
2. Patryk Janota – csv reading, data presenting (main view), renta-wieczysta, Data Scanner, Add/Edit/Remove Reports in DB, Create Services,
3. Damian Malinowski - ????? web data gathering, analyzing..

## 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] [Kolejnoscmetod w klasach w sekcji Clean Code – **USE #REGIONMANDATORY**]

TO-TAKE(2) : \* Db Backup (csv, sql, other) \* we need some sort of save,load,back up system after new data occurs.

TO-TAKE(2) : \* Option to manual add data to our database, nice box view, data validation before inside, etc. \* This will be neceseary if we do not finish on time our pdf parser ☺

TO-TAKE(2) : \* Poprawienie osi wykresu (format w tysiacach lub tysiace oddzielone spacjami), Sortowanie od najwiekszego (rok&kwartal) do najmniejszego) lub wedle uznania., Poprawienie przedzialki na dolnej osi wykresu np. (co kwartal if < 10 jak wiecej > 10 co 2. – dobrac tak by widocznosc byla maksymalna.

TO-TAKE(2) : \*Browse through pap calendar. (<http://biznes.pap.pl/NSE/pl/reports/espi/term,0,0,0,1>)

\*Prepare class which will be able to load the fresh (now 2013 1Q result) for given company.obj (<http://biznes.pap.pl/NSE/pl/reports/espi/view/231746> ED INVEST)

Cezary(1) : \*Parsing PDF file, which will be provided for other external class(look task up)

[Estimated Time: **1 month to initial version**.

TASKS: Convert .pdf to .txt. Regex Try to take useful for parsing data to Reports section. ]

\*Parser PDF should be able to posses all the important data from the file, for reports record so polish names (Przyodnettonasprzedazy, PrzychodBrutto, EBIT, Przychod z dzialanoscioperacyjnej, przychodnetto, (danedotyczaceaktyw) , (danedotyczacepasyw), dane z segmentowdzialanosci (np. Eksport) bardzo przydatne do analizy.   
TO-TAKE(2) : \* Pozyskiwanie dannych dotyczacych surowcow (srebro zloto itp) , waluty podstawowe(eur/pln) (usd/pln) (yen/pln) (rubel/pln). Wprowadzanie ich do bazy ? przygotowanie obiektu, do przetrzymywania ? Pobierane dynamicznie (online) ?

\* Requirment interesują nas dane 3Miesięczne czyli kwartalne. Trzeba takie dane ‘wyprodukowac’ wyliczyc na bazie danych pojedynczych :)

TO-TAKE(2) : \* Prepare InternalViewTabbedProvider, przygotować logiczną architekturę dla wewnętrznego tab-panelu (zakladki) ktory bedzie obslugiwany jako panel wewnątrz SPLIT-CONTAINER- PANEL2., Kontener pierwszy zostanie bez zmian.

* Zbudowanie komunikacji miedzy kontenerem 1-szym InternalTabbedViewProviderem. Swobodona i elastyczna prawdopodobnie oparta na eventach.

Patryk(1): Addition on the csvEnum for BILANS AKTYWA PASYWA i CASH FLOW.

Patryk(1): Prepare Rzis and Bilansseperation

Patryk(2): Start scanner architecture module, and basic class implementation on the interface plugs ☺

Cezary(1): Przyspieszenie wczytywania danych do bazy do max 2 minut. Czyszczenie bazy max 30 sek.

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

Cezary(2): Clean -> ServiceLocator -> dodac funkcje usuwania ze slownika. Clean bedzieszybszy, alboprzerzucicsięna**PRISM Service Locator**

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): TranspozycjaDataGrid View....

## Key Money MAKER features.

GREEN – Implemented into application.

1. **RentaWieczysta** (Calculated)– ver 0
2. DCF (Discounted Cash Flow Value) Wycena DCF. (Most hardcore to do, many external parameters… - very close to TV although DCF is good for new dynamic good potential clear business model like BENEFIT SYSTEM.)
3. Volumen Signal’s( how to catch ? how to judge if is positive volume(positive information ) or negative(bad informations). The WAY TO CATCH?: take volume and price in small amount of time, when end price is lower than start price, then volumen is negative ☺)
4. WEB Scanner - 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)
5. Data Aquisitor– get the data from every possible source. PAP, Bankier.pl, Money.pl and other webpages.
6. **Chart Visualisation**– clean and good visualization of the result in the futer, process chart for some kind of external reports. Fast way of comparing more than two companys on the chart possibility to turning off and on the companies, and other LOOK AND FEEL features.
7. All in **scanner** –
   1. Terminal Value(Renta Wieczysta - wartosc teoretyczna spolki do generowania gotowki)
   2. Test TV versus Real Stock Price (Renta wieczysta kontra cenna gieldowa)
   3. Stable quarter growth on the last (4-8-12 quaters – parametrize)
   4. ROE / ROA best in the company enemys environment. (Parametrize)
   5. Other good quality financial parameters which determinize the good company value.

## 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**

## APP ARCHITECTURE/STRUCTURE.

[TODO: Put beautiful schema with IDSA structure here]

Application based on **Design Patterns**. Main pattern is **MVP**: Model-View-Presenter. Secondary patterns are: **Unit Of Work, Repository,** **Dependency Inversion(Service Locator), Singleton**.

When create application **TRYING** stick to the **SOLID**(Single responsibility principle, open/close principle, Liskov substitution principle, Interface segregation principle and Dependency Inversion principle) principles.

(source: <http://en.wikipedia.org/wiki/SOLID_(object-oriented_design)>

Using **Events & Delegates**. [TODO: need info ]

Implement Services: Cache Service, Calculation Service, Chart Service, Company Data Service, Display Format Service, Data Service.

To implement → Db Management Service, Company Matcher Service. More ???

To move → PAP Parser Service.

[TODO: please Patryk write sth about usefulness of services ;) ]

Using thirdparties: **Entity Framework, Ninject, NLog, Html Agility Pack, PRISM, Fast CSV Reader.**

**Entity Framework** – open source object-relational mapping (ORM) framework for the .NET Framework.

(homepage & source: <http://msdn.microsoft.com/en-us/data/ef(v=msdn.10).aspx> additional info: <http://msdn.microsoft.com/pl-pl/library/bb399572.aspx> <http://en.wikipedia.org/wiki/Entity_Framework>)

**Ninject** – open source dependency injector for .NET

(homepage: <http://www.ninject.org/>

info about dependency injection: <http://en.wikipedia.org/wiki/Dependency_injection>)

**NLog** – open source logging platform for .NET, Silverlight and Windows Phone with rich log routing and management capabilities. It makes it easy to produce and manage high-quality logs for your application regardless of its size or complexity.

NLog can process diagnostic messages emitted from any .NET language (such as C# or Visual Basic), augment them with [contextual information](https://github.com/nlog/NLog/wiki/Layout-Renderers) (such as date/time, severity, thread, process, environment enviroment), format them according to your preference and send them to one or more [targets](https://github.com/nlog/NLog/wiki/Targets) such as file or database.

(source: <https://github.com/NLog> homepage: [http://nlog-project.org](http://nlog-project.org/)).

**Html Agility Pack** – agile HTML parser that builds a read/write DOM and supports plain XPATH or XSLT (you actually don't HAVE to understand XPATH nor XSLT to use it, don't worry...). It is a .NET code library that allows you to parse "out of the web" HTML files.

(homepage: <http://htmlagilitypack.codeplex.com/>)

**PRISM** – provides guidance designed to help you more easily design and build rich, flexible, and easy-to-maintain Windows Presentation Foundation (WPF) desktop applications, Silverlight Rich Internet Applications (RIAs), and Windows Phone 7 applications.

Based on shell, Regions, Views.

(homepage: <http://compositewpf.codeplex.com/>)

**Fast CSV Reader** – oepn source the fastest CSV reader. The reader was processing about 30MB/sec, it performs about 15 times faster, than OLEDB and regex methods.

(homepage: <http://www.codeproject.com/Articles/9258/A-Fast-CSV-Reader>)

### 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,thatwewillgetintothefurther.Fornow,iseasytogetdataTablefromdatagrid,presenterofviewcanprovidethisaction, although wemustbeflexibleandbeabletovisualizethedatathatuserwanttosee.

             -> 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.

* + We need even more to work **on separate data.** (Czarek you now using the DataBase data not recalculated)

