# Financial Ratios Calculator

Design and implementation of a simple financial ratios calculator, where guest Users (without authentication) will be able to input chosen figures from a Balance Sheet and an Income Statement (// TODO: include import) and (if input data is valid) to receive (as an on-screen output (// TODO: include export)) a number of widely used financial ratios, as well as a Cash Flow Statement (based on the indirect approach) and implied Capital Expenditure figures (based on available information). Registered Users will be able to store all data and later (through authentication) have access to it. Access includes full CRUD on User’s data. The Calculator is implemented as server-side Web application using ASP .NET MVC 5.

**Data Layer:**

1. General
   1. Use Entity Framework as ORM and MS SQL Server Local DB as database storage engine;
   2. Implement Repository pattern;
2. Models (Code Fist approach will be used to access data from EF)
   1. User (ASP .NET Identity will be used to keep users and their encrypted passwords)
      1. Username;
      2. Password (to be stored encrypted in the DB);
      3. Email;
   2. Financial Statements (a model that includes the two basic statements – BS and PL)
      1. Author (currently logged user);
      2. Name (optional, used for indexing);
      3. CompanyName (optional, used for access to previous records);
      4. Comments (optional);
      5. EndDate (required);
      6. Period (required; in months);
      7. DateCreated (automatic);
      8. DateModified (automatic);
      9. DateDeleted (automatic; deletion will (for the time being) not be physical, the object will only be marked as deleted);
      10. BalanceSheet;
      11. ProfitAndLossStatement;
   3. Company
      1. Name;
   4. Balance Sheet
      1. Extended set of item as per common BS models, including control items for Total Assets/Liabilities, Total Cash & Cash Equivalents, Total Own Capital, etc., which will be read-only; only positive integers are allowed for the input items;
      2. IsValid (boolean property that validates the form based on common accounting principles);
   5. Profit and Loss Statement
      1. Extended set of item as per common PL models, including control items for EBITDA, EBIT, etc., which will be read-only; only positive integers are allowed for the input items;
      2. IsValid (boolean property that validates the form based on common accounting principles);
   6. Items
      1. CurrentPeriodValue;
      2. PreviousPeriodValue (will be validated against previous records of the same company, when available; only warnings of inconsistency will be displayed);

**Common features**

1. Layout – ASP .NET MVC Layout will be used;
2. In the navigation when an user is not logged in hide the search form;
3. When a user is logged in, show search form and display the 10 most recently modified FS’s;
4. When using the search form, the list will be updated onChange;
5. There will be no paging (for the time being);
6. User should be able to sort FS’s by DateCreated, DateModified, Name, CompanyName;
7. There should be a “Create new FS” button on top of the list;
8. Each entry will be displayed using partial HTML;
9. There should be 3 buttons for every entry – “View”, “Edit”, “Delete”;
10. Use unobtrusive client-side validation and server side validation, built in ASP .NET MVC;
11. In case of error (e.g. DB connection lost, incorrect request, etc.), an appropriate error message should be displayed;

**User Interface**

1. Public
   1. User interface should be created using Bootstrap;
   2. At the Home page should be displayed three buttons for the available choices: “Log in”, “Register” or “Use without logging”;
2. Logged User
   1. Display 10 last modified FS’s;
   2. Cache the last 10 modified FS’s for 1 hour;
   3. Above the list display a large button “Create new”;
   4. Display each of the 10 items in a separate box;
   5. For every item of the list there should be 3 small buttons in the right bottom corner of the box;