**Developer Notes**

* Technologies used: Asp.Net MVC 5, jQuery, EF 6, Telerik kendo controls

**Project Structure**

* Damacon.StoreD.Console

To do various console tasks of the projects like generating updated resource file

* Damacon.StoreD.DAL

It’s the Data Access Layer to interact with database. Below is the logic in various folder/components of this project

* + Database –
    - EF - It has the EF edmx file. So, on updating db schema, we need to update edmx as well.
    - Migration - As a best practice, to keep track of all the db schema or data changes for new development or enhancement, create a sql script and put it into migration folder.
    - Partial – It is used to add something to the entities generated by EF. The namespace should be same as of the EF entities
  + i18n – It has the framework to support multiple languages in the application. Below are the steps to add new language resource in the application:
    - Add the language resource data in the database via sql script.

INSERT INTO [dbo].[LanguageResource] VALUES (‘<Key>','<English Text>','Italian Text',null, null)

* + - Run Damacon.StoreD.Console with debug parameter UpdateResources
    - It will generate the updated **Resources.cs** file in bin/debug folder of Damacon.StoreD.Console project
    - Replace the **Resources.cs** ini18n folder in the Damacon.StoreD.DAL project
  + Operations – It has all the DAL operations which actually interacts with DB
    - Contracts – has the interfaces for the operations. Like ILoanDAL.cs has all the loan related operations
    - Concrete – has the concrete implementation of the DAL operations
  + DALFactory.cs
    - Dal implementation has internal and just the interfaces are public. It’s a small factory to init the implementation by contract.
* Damacon.StoreD.Shared
  + Just a dummy project to store common things but not getting used right now
* Damacon.StoreD.WebApp

It’s has all the view logic of the project. It created in Asp.Net MVC and most the grids are kendo grids

* + Apart from regular Controller and View folder, it has few other folders as:
    - AboutFiles – It has the about files that shows under about menu item
    - Assets – 3rd party css and javascript libraries
    - Content – kendo libraries
    - Export – To generate pdf and excel reports
    - Helpers – Additional helper classes. One of the main class in this folder is ModelMapper.cs which use to convert EF model to UI Model and vice versa.
    - MarkupConverter – Convert about rtf files to HTML
    - Models – UI model that binds to views
    - Scripts – Javascripts

Branching Strategy

* You can see 2 branches as develop and master
* All the current development will go in develop branch and will be provided on test env for testing
* Once testing is done, we can move code to master branch via pull request.