**LACERA**

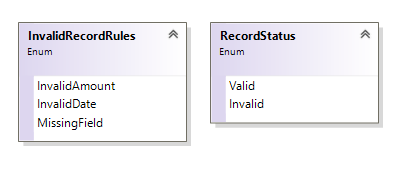
I created the solution with ***Microsoft Visual Studio 2013*** and***.Net framework 4.5.***

I made the following assumption when developing the solution

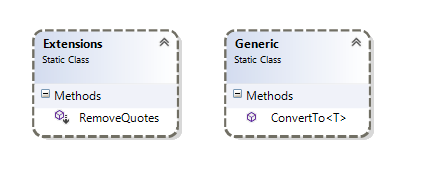
1. Since we have an InvalidAmount business rule, I applied the same for salary greater than 0. So when the salary is 0, the record is marked as INVALID

**Lacera.Enums**

I added two enums, one for the validation rules and the other for record status



**Lacera.Helper**

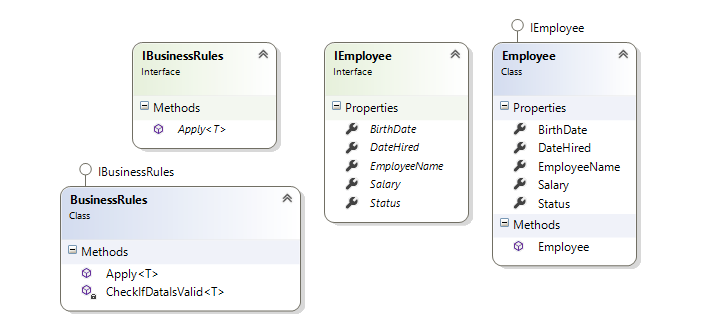
I created two static helper classes called Extensions and Generic

The Extension class has an extension to remove quotes from a given string and the generic class has a convert to method which could be used to convert an input to any given data type, if the input could not be casted to the requested data type, then the default value of that data type would be returned.

For example, If user requests an input of ”90000” to be converted to date time, it would return “1/1/0001” since the cast is not valid and the default value of date time is 1/1/0001

**Lacera.Objects**

The Object project contains all the required Models/Entities. I created a separate project, so that we could reuse the classes in different projects. I created two entities, one for Business Rules and another for Employee.

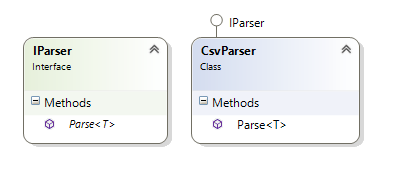


In order to reuse the class in any project, the Apply method takes in a generic type parameter so that it’s not tightly coupled to a single object/entity.

**Lacera.Parser**

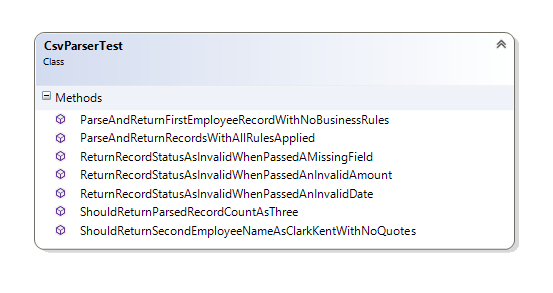
The Parser class is very generic and has been written with performance in mind

1. It can be used to parse CSV to any DTO/Entity class with very minimal effort.
2. A Stream reader is used instead of File.ReadAllLines so that when a large file needs to be processed we would not run into any memory issues.



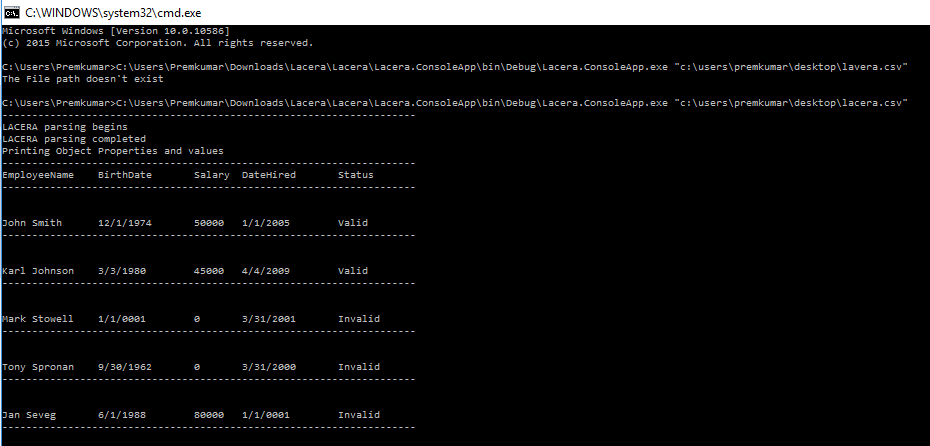
**Lacera.Parser.Test**

This project contains all the unit tests for the csvparser class.



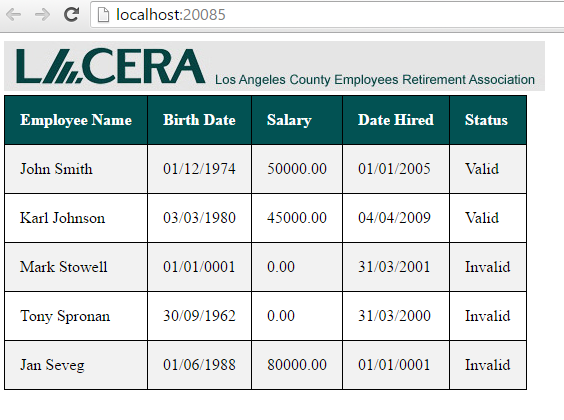
**Lacera.ConsoleApp**

This is the console app project which takes in the file path as an argument. It parses the file and writes the record to the console screen



**Lacera.Mvc**

This is the MVC project which takes in a hard coded file path. It parses the file and writes the record to the view. I apologize if it was not appropriate to add the LACERA logo to the MVC project.



**Improvements**

I made sure that the SRP (Single responsibility) and DRY principles are followed. I wanted the parser and business rules to be generic enough so that it could be re used in any project. I think I succeeded with implementing a generic class for parsing CSV, but with respect to business rules it is generic enough but I think it violates the OCP(Open closed principle) since any new rule addition would require the class to be modified

Autofac for dependency injection and a generic rule manager for business rules would be a great thing to have in the project.