



CSE 499 Final Report

ASP.NET MVC Bootstrapper: Framework and a project template.

An enterprise framework which can reduce development time and balance between work and life.

Md. Alim Ul Karim
112 0821 042
Student of ECE Department,
North South University
Email: alim.karim.nsu@gmail.com

Supervised Under:
Dr. Nova Ahmed
Assistant Professor
Department of Electrical and
Computer Engineering
North South University
Bashundhara R/A, Dhaka,
Bangladesh.



Md. Alim Ul Karim
University Id: 112 0821 042,
Email : alim.karim.nsu@gmail.com
Department of Electrical and Computer Engineering,
North South University



Dr. Nova Ahmed
Asst. Professor,
Email : nova.ahmed@northsouth.edu
Department of Electrical and Computer Engineering,
North South University

LETTER OF DECLARATION

This paper is part of my Bachelor's Degree in Computer Science and Engineering.

In addition, I hereby, declare:

1. This paper has not been replicated using any other paper.
2. None should replicate or cannot replicate without referring to this paper.
3. All the major references used in the project are listed in the paper, in addition each of those are cited inside the code.
4. All original work has been done online and dated at bit.ly/bootstrap-commits (github opensource source control), which cannot be manipulated afterwards.
5. Beta release can be found at bit.ly/bootstrap-release

I here by aware of the fact that breach of these statements can lead to academic misconduct.

Author's Sign

Md. Alim UI Karim
112 0821 042
Student of ECE Department,
North South University

LETTER OF APPROVAL

The thesis titled "ASP.NET MVC Bootstraper" has been submitted by Md. Alim Ul Karim (ID: 1120 821 042) on _____ who is student of North South University and releated to Electrical and Computer Engineering (ECE) department. In addition, this paper has been accepted with satisfactory.

Dr. Nova Ahmed

Assistant Professor
Department of Electrical and Computer Engineering
North South University
Bashundhora R/A, Dhaka,
Bangladesh.

ACCKNOWLEDGEMENT

I would like to thank *Dr. Nova Ahmed, Assistant Professor at North South University* for giving me the honor to work on this project. Her glracious guidance and effort has been a true motivation for me. It was my immense pleasure to work on this project under her supervision. She inspired me in many ways. Finally, I would like to thank my parents for always been there for me.

TABLE OF CONTENTS

Founder	2
Letter of Declaration	3
Letter of Approval.....	4
Accknowledgement.....	5
TABLE OF CONTENTS	6
Glossary	9
Abstract	4
I. Introduction (Chapter 1)	8
II. Background (Chapter 2).....	11
III. Related Works (Chapter 3).....	14
IV. System Architecture (MVC) and Design (Chapter 4)	15
V. Existing System Design Process (Chapter 5).....	19
A. Code-first approach:	19
B. Database-first approach:	20
C. Model-first approach:.....	20
D. Decision for this framework:	20
VI. Problems of Existing Framework (Chapter 6)	21

A.	Inefficiency in built-in user registration process (ASP.NET [3] Identity)	21
B.	Out of the box inflexibilities	21
C.	Time zone detection.....	22
VII.	Project Description (Chapter 7).....	24
VIII.	Features (Chapter 8).....	25
A.	Reinventing the Wheel (Generic).....	26
B.	Time-zone fix, Country, Phone and Language	27
C.	UX and UI concepts	28
D.	Greater Degree of Security	29
E.	Efficient Scaffolding	30
F.	Hassle-free Uploading	30
IX.	Results and Verification (Chapter 9).....	31
X.	Future Work (Chapter 10).....	32
XI.	Additional Plugins and Resources Used (Chapter 11)	33
A.	Front-End	33
A.	Backend.....	33

<ul style="list-style-type: none"> References <ul style="list-style-type: none"> Antlr3.Runtime DevMVCComponent Elmah Elmah.Mvc EntityFramework EntityFramework.SqlServer MarkdownDeep Microsoft.AspNet.Identity.Core Microsoft.AspNet.Identity.EntityFramework Microsoft.AspNet.Identity.Owin Microsoft.CSharp Microsoft.Owin Microsoft.Owin.Host.SystemWeb Microsoft.Owin.Security Microsoft.Owin.Security.Cookies Microsoft.Owin.Security.Facebook Microsoft.Owin.Security.Google Microsoft.Owin.Security.MicrosoftAccount Microsoft.Owin.Security.OAuth Microsoft.Owin.Security.Twitter Microsoft.Web.Infrastructure Newtonsoft.Json Owin SendGrid.SmtpApi SendGridMail System System.ComponentModel.DataAnnotations 	<ul style="list-style-type: none"> System.Drawing System.EnterpriseServices System.Net.Http System.Net.Http.Formatting System.Net.Http.WebRequest System.Runtime.Serialization System.Security System.Web System.Web.Abstractions System.Web.ApplicationServices System.Web.DynamicData System.Web.Entity System.Web.Extensions System.Web.Helpers System.Web.Http System.Web.Http.WebHost System.Web.Mvc System.Web.Optimization System.Web.Razor System.Web.Routing System.Web.Services System.Web.WebPages System.Web.WebPages.Deployment System.Web.WebPages.Razor System.Xml System.Xml.Linq WebGrease 34
---	--	--------

XII.	Source Code	(Chapter 12)	35
XIII.	Conclusion	(Chapter 13)	36
XIV.	References	(Chapter 14)	37
XV.	Code Commits	(Chapter 15)	39
XVI.	Notable Code Examples	(Chapter 16)	41
A.	Timezone code examples		41
B.	Helper Methods for all Developers.....		47
C.	JavaScript plugins for all components.....		56

GLOSSARY

ASP ACTIVE SERVER PAGES;

.NET FRAMEWORK;

MVC MODEL VIEW CONTROLLER;

LARAVEL PHP FRAMEWORK;

ASP.NET MVC ACTIVE SERVER PAGES .NET MODEL VIEW CONTROLLER;

ASP.NET MVC BOOTSTRAPPER AN ENTERPRISE DEVELOPMENT FRAMEWORK FOR LARGE
SCALE USERS

BOOTSTRAPPER IS A CONCEPT TO GET THINGS STARTED QUICKLY.

COC CONVENTION OVER CONFIGURATION

CTP COMMUNITY TECHNOLOGY PREVIEW

CMS CONTENT MANAGEMENT SYSTEM

ROI RETURN ON INVESTMENT

ABSTRACT

ASP.NET MVC Bootstrapper or MVC Bootstrapper is a framework and a project template which is designed on top of another framework and project template to boost productivity for ASP.NET MVC developers. Anyone can write framework and customized methods, there is no challenge; the big challenge is to modify Visual Studio to increase productivity. In addition, productivity (doing things quick) and efficiency (developing applications for large scale users) always opposes each other, as a result it is a big challenge to make a product productive and at the same time optimize the code for large scale users (efficient).

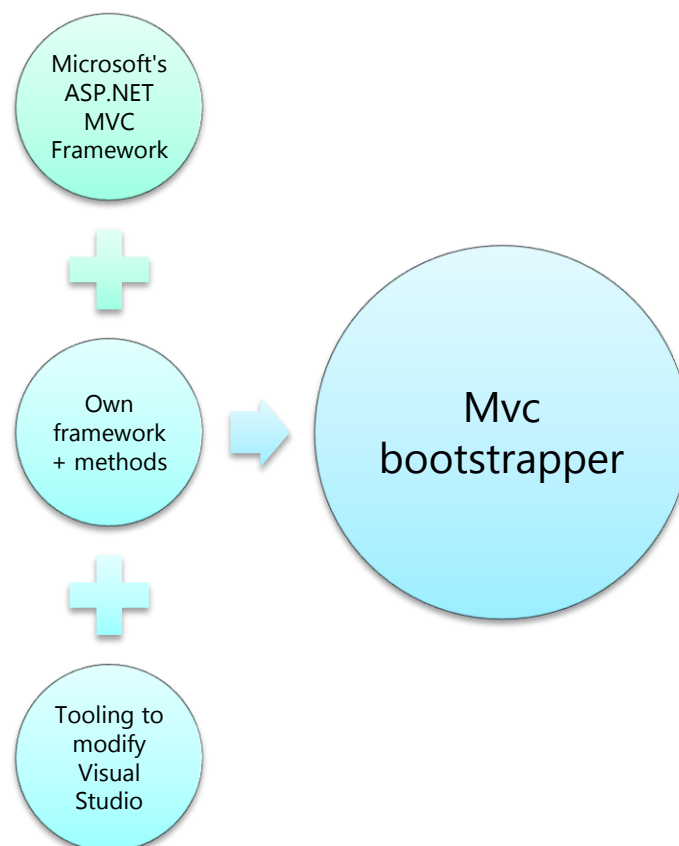


FIGURE 0.1: MVC BOOTSTRAPPER CONCEPT

In contrast, from many other products like Orchard [1], Wordpress [2], ASP.NET [3] etc... MVC Bootstrapper [4] is easy to use, integrate and modify. Mentioned content management system, blog and framework has a lot of cons which is greatly enhanced in this framework. A plugin integration, such as, a datepicker or combo takes quite a while to integrate inside development pages; however inside MVC Bootstrapper it only takes a line of code. All of these are done through javascript plugin which written by Md. Alim Ul Karim which also uses other javascript frameworks (which are cited in the reference section). Framework does not reinvent the wheel, whichever plugin (combo, datepicker, uploader) is available had been used it. Moreover, those existing plugins are also modified those nicely (to customized UI, UX) so that it can be easily integrated with only a line of code.

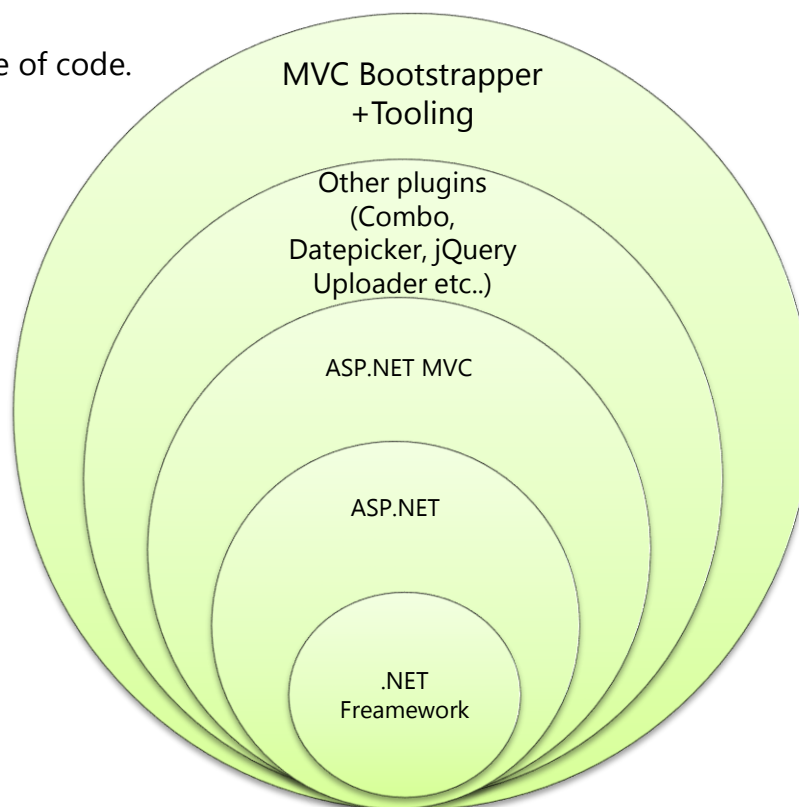


FIGURE 0.2: MVC BOOTSTRAPPER HIGH LEVEL OVERVIEW

Writing a framework on top of another framework is not a new concept, Microsoft has done it on their .NET Framework 3.0 [5] and 3.5 [5] release. .NET framework 3.0 are additional libraries and APIs on top of .NET framework 2.0 [5] and so does .NET framework 3.5 [5].

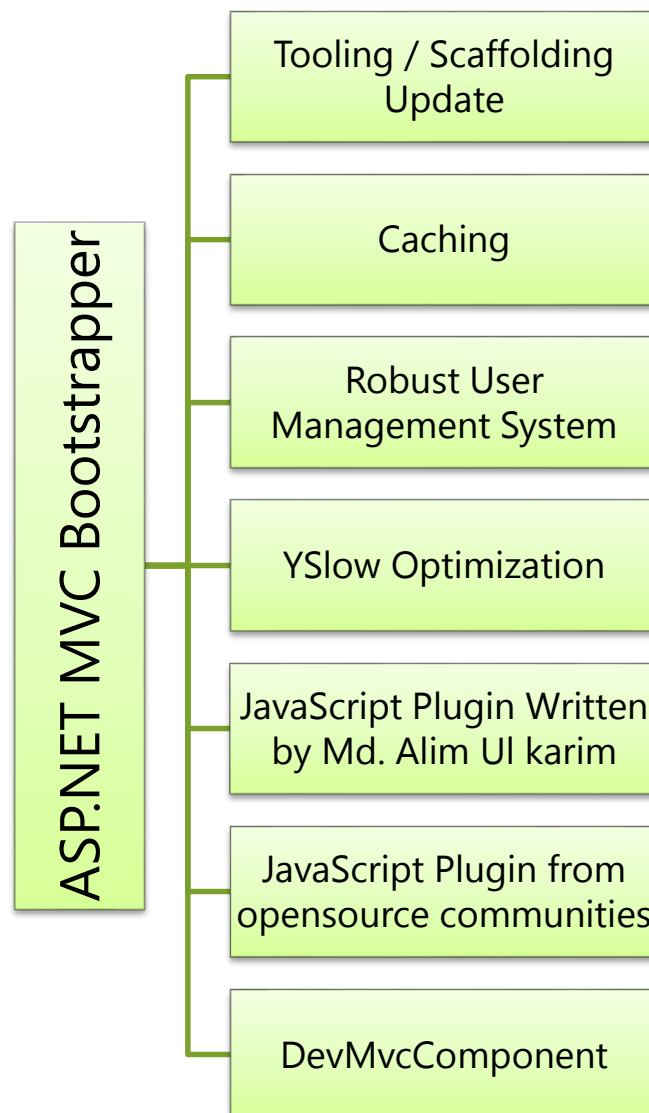


FIGURE 0.3: MVC BOOTSTRAPPER COMPONENTS AND FEATURES SUMMARY.

ASP.NET MVC Bootstrapper consist with a lot of additional and modified functionality on top of the exsisting framework. There is a test website [6] running and

all codes are open [\[7\]](#) with history and dates on github. These dates cannot be manipulated or altered. Consequently, work history and dates prove the existence of framework and test site.

I. INTRODUCTION (CHAPTER 1)

The ASP.NET MVC Bootstrapper is a framework that allows a developer to be productive in both the corporate and enterprise world *without having advanced knowledge of enterprise development and optimization for large scale users*. The framework itself is optimized and scalable for hundreds of thousands of users. To lessen the burden of developers, MVC Bootstrapper worked around key redundancies of the existing framework and offers an efficient and fully prepared web development platform.

ASP.NET MVC [8] has *clean separation with model, view and controller*, however, **problems** persist within the system are significant, for instance ASP.NET Identity [9] (which is the user-management framework out of the box) is highly inefficient, scaffolding (code generation tool) generates wet code, common features like time-zone fixation worldwide or country information missing, file upload is a big hassle and implementing takes hours, foreign key relations pull inefficient data from database, UI and UX design is moderate and this is where **ASP.NET MVC Bootstrapper** comes into play.

ASP.NET MVC Bootstrapper does not break existing system architecture (MVC [10] - model, view and controller), moreover it adds few extra convention along tons of features and plugins which takes months to write and implement. Visual Studio [11] and nugget [12] extension manager provides flexibility on adding plugins in a project,

however it does not do the implementation and ASP.NET MVC Bootstrapper plugins are implantation based. With this framework developers will write a single line of code and achieve tons of benefits. ASP.NET MVC Bootstrapper generates very efficient codes for large scale users and automatically caches data where necessary. However developer will have their control to change the cache facility. Finally, it puts UX and UI out of the box based on based the book “Don’t make me think” [13] .

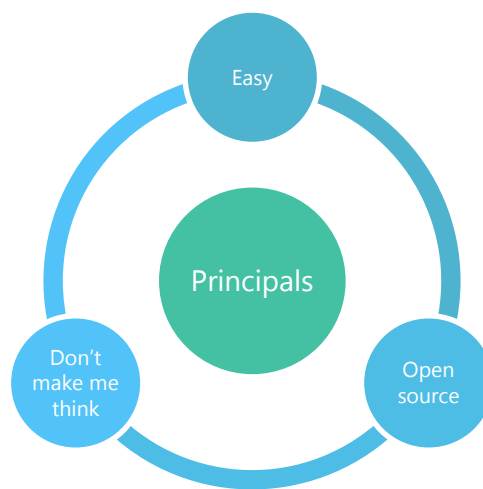


FIGURE 1.1: ASP.NET MVC BOOTSTRAPPER PRINCIPALS

ASP.NET MVC Bootstrapper build on top this three principals (easy to use, open source, finally don’t make me think).

1. **Easy:** Framework should be easy to understand and easy to modify.
2. **Open source:** Framework is open source at <https://goo.gl/xRDfoi> (github).
3. **Don’t make me think** [13]: It’s a famous book concept by Steve Krug, where UX design is explain very well, and the whole framework is

designed such a way so that developers don't have to think much,
instead out of the box system will do the heavy lifting for them.

II. BACKGROUND (CHAPTER 2)

ASP.NET MVC Bootstrapper is a fine tune framework, however there are few flaws which can be tiresome sometime. To extend the framework one has to know javascript, which is hard for many developers. As a result, it could be a blocked for many developers. Any framework needs learning (small or big) to implement, even though it is easy to use, one has to read manuals (or watch videos) to implement it. Framework has extensive features, and currently no other CMS or framework has it. One of the most notable one is **ip to country detect and then detect time zone**, which is influenced by google's registration page. Other features like automatic caching, optimized scaffolding [14] makes the framework unique and robust. Framework is also consist of user management feature with handling role management with privileged orders and so on.

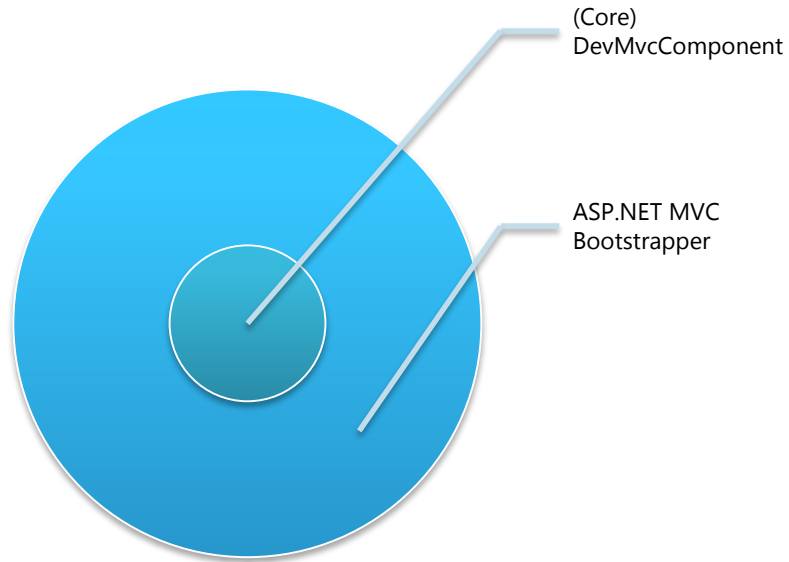


FIGURE 2.1: CORE OF ASP.NET MVC BOOTSTRAPPER IS DEVMVCCOMPONENT

DevMvcComponent [15] is the core of this framework which is developed by Md. Alim Ul Karim, which consist of several unique features like error logging, sending emails, paging data, caching, cookie management and so on. Figure 2.2 shows the features and components of DevMvcComponent [15].

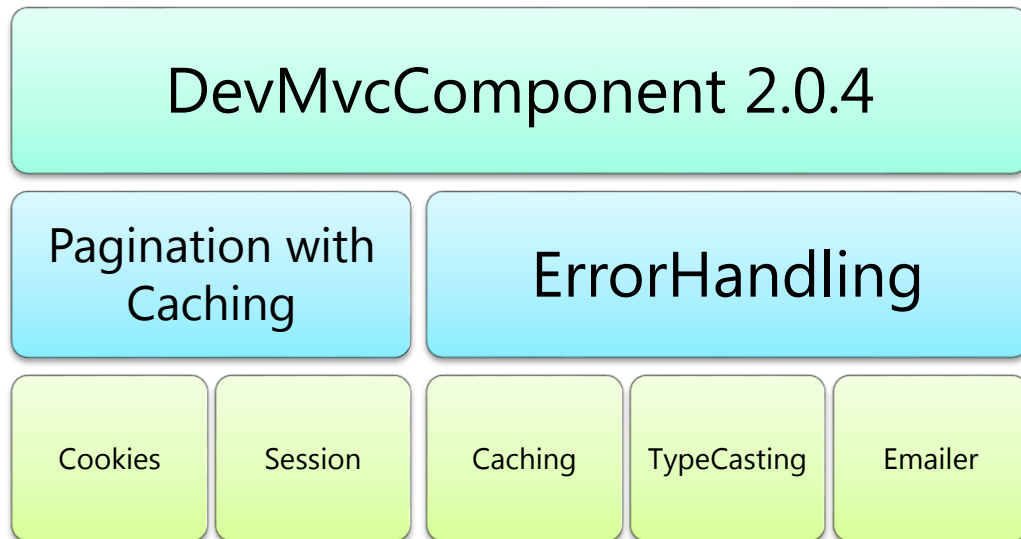


FIGURE 2.2: DEVMVCCOMPONENT 2.0.4 [15] COMPONENTS AND FEATURES.

Most notable features in DevMvcComponent are *ErrorHandling with email and Pagination with Caching*, there are several components which offers pagination, however none provides caching with it. There are also popular error handler ELMAH [16] to handle errors, however it does not provide custom error handling feature for specific developers and DevMvcComponent [15] is all about custom error handle for specific developers. Developers can give specific email address to send specific emails to specific developers or admins.

Altogether, ASP.NET MVC Bootstrapper is great if we ignore learning part and javascript skills require sets.

III. RELATED WORKS (CHAPTER 3)

At the end of 2005 Ruby on Rails [17] came up with great concepts, and very soon it became popular. After that Microsoft followed few ideas from Ruby on Rails [17] and implemented those inside ASP.NET MVC [8] framework. Since, ASP.NET MVC Bootstrapper is a framework on top of ASP.NET MVC [8], hence MVC Bootstrapper consists of those concepts from Ruby on Rails [17]. The concept of scaffolding [14] is came from Ruby on Rails [17].

Wordpress [2] is a very popular blogging system for developers and non-developers, with help of plugins any non-developer can use it as blogging, content management, video posting and many more. However, when it comes to making or developing plugins to custom made page or programmatic page, it takes a lot of time. In contrast, ASP.NET MVC Bootstrapper is developed for developers, so any place custom plugin implementations are very easy and thus takes few lines of code.

Laravel [18] is PHP framework which is popular among many developers, and has similar features like ASP.NET MVC [8], Ruby on Rails [17] and ASP.NET MVC Bootstrapper. Scaffolding [14] is the common feature which is available inside these frameworks and CMS. However, architecture and developing styles are different for each frameworks and CMS.

IV. SYSTEM ARCHITECTURE (MVC) AND DESIGN

(CHAPTER 4)

There was a time when developers introduced 3-tier or N-tier (*data-architecture, business architecture, presentation/client architecture*) system to achieve performance benefits and convention. However, in last few years cloud computing has resolve this backdated concept. Concept behind this 3-tier or N-tier largely involved 3 or N separate machines (*data-architecture, business architecture, presentation/client architecture*) so that multiple machines can handle and process different actions, hence performance increases. However, nowadays in cloud computing we can add multiple array processors just by few click and there is no need to use this convention.

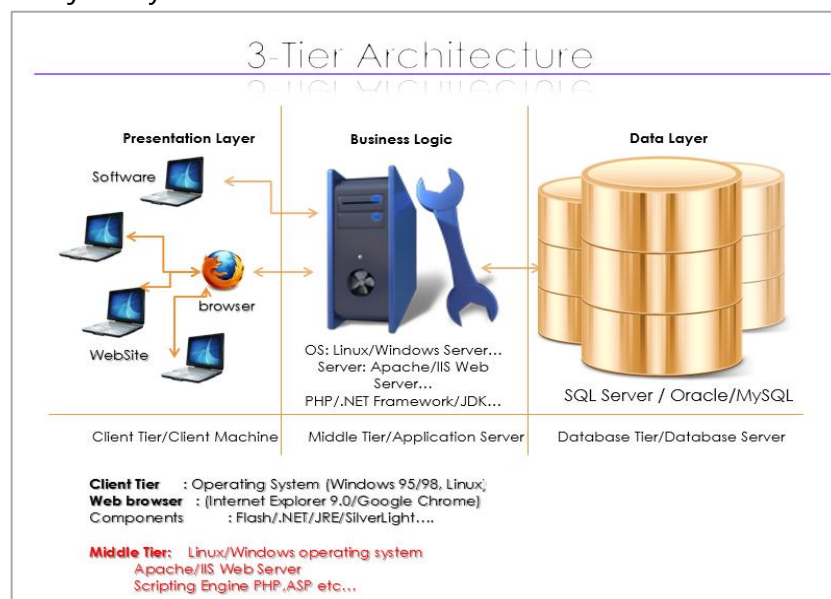


FIGURE 4.1: 3-TIER ARCHITECTURE (DRAWN BY ALIM UL KARIM, JUNE 2012)

Several noticeable problems exist in 3-tier or N-tier architecture, for instance machines are separate, as a result communication is hard, and it is even worse when developers have to write *every method as socket programming or remote accessible*

method. Every socket method had a huge over-head on data-transmission from one-architecture to another. Managing three or N projects and machines are also very hard and cumbersome for developers. Finally end-results are not that beneficial, however developers has been using it for over decades. For some context it worked and optimized performance, however in cloud context it is not necessary.

As a solution, this framework usages ASP.NET MVC, thus system architecture remains same. Stackoverflow.com is one of greatest enterprise web application where millions of users visit every day and make thousands of posts every day, it is amazing that very few people know that first version of stackoverflow was built with ASP.NET MVC 2, later it was upgraded to ASP.NET MVC 4.

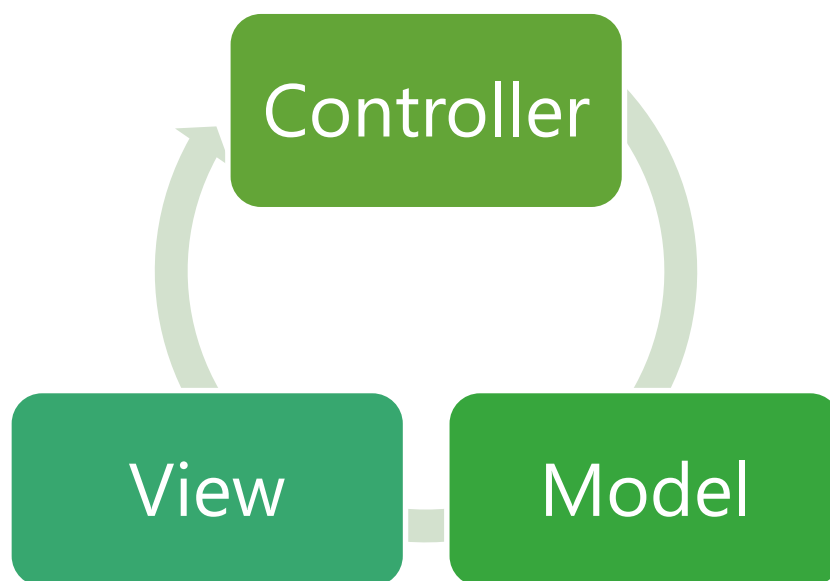


FIGURE 4.2: MODEL VIEW CONTROLLER ARCHITECTURE.

In MVC there are these three components (Model, View, and Controller) which imposes *convention over configuration (COC)*.

1. Model: represents data or holds data information.
2. View: represents the UI or interfaces for client use.
3. Controller: which glues other two and finally lead to an appropriate view.

Convention over configuration (COC) [19] is a software design paradigm which enhance decision making scenarios and keep things simple. If we make a configuration then we can change the convention, however by default convention will win. As a result, system architecture is very flexible. Moreover, there is a video from the founder to explain the basics of ASP.NET MVC [20] (<http://bit.ly/asp-net-mvc-basic> prequel to this framework).

In addition with the system architecture, in system design it contains user management, timezone, ip to contry and so on. Below a sample database diagrams is given in figure 4.3, the whole diagram will be available at bit.ly/mvc-boots-acc-dia .

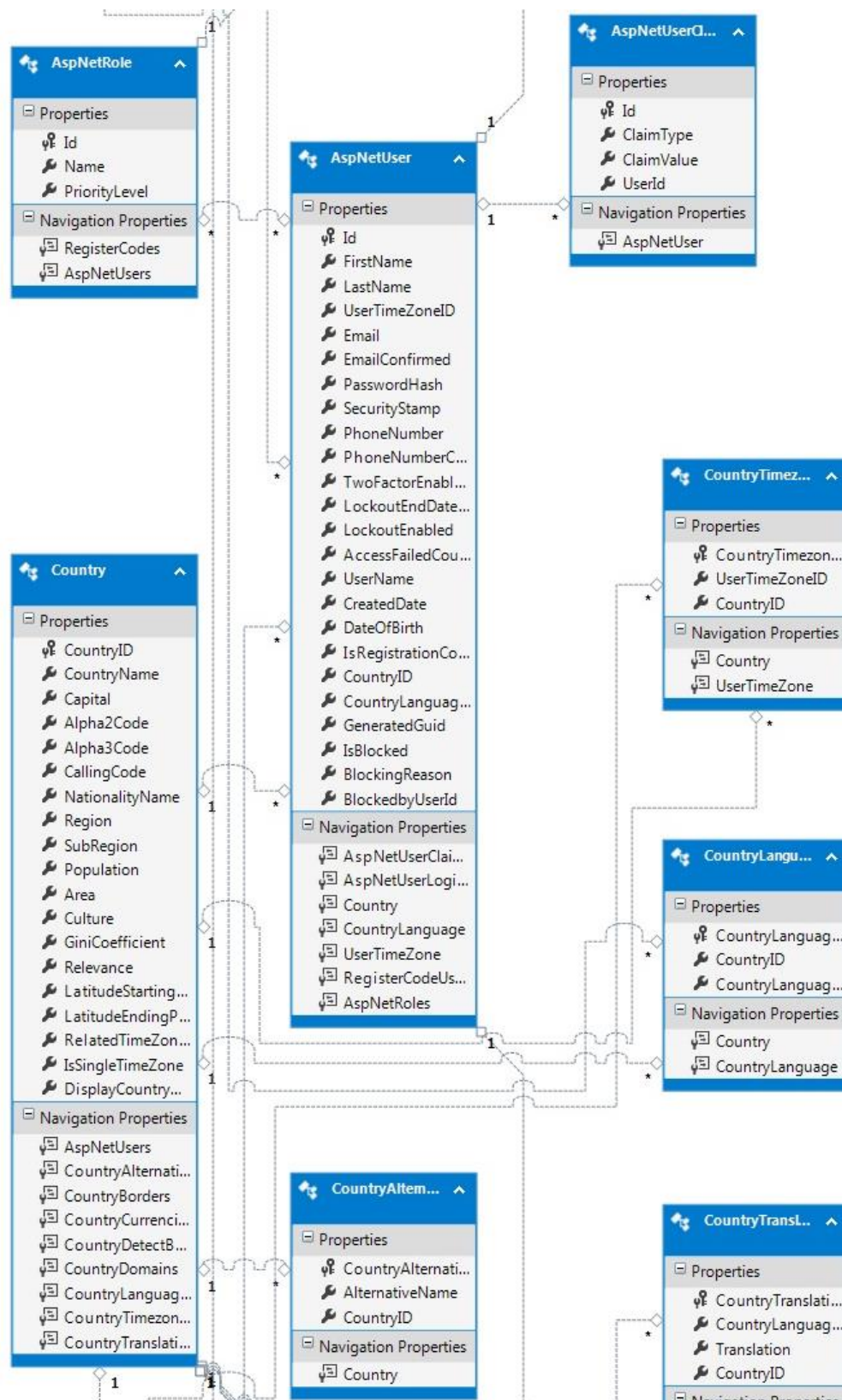


FIGURE 4.3: SYSTEM DESIGN PART - INTERNAL DATABASE (BIT.LY/MVC-BOOTS-ACC-DIA)

V. EXISTING SYSTEM DESIGN PROCESS (CHAPTER 5)

In ASP.NET MVC [8], there are three ways (*code-first approach*, *database-first approach*, and *finally model first approach*) to design the system.

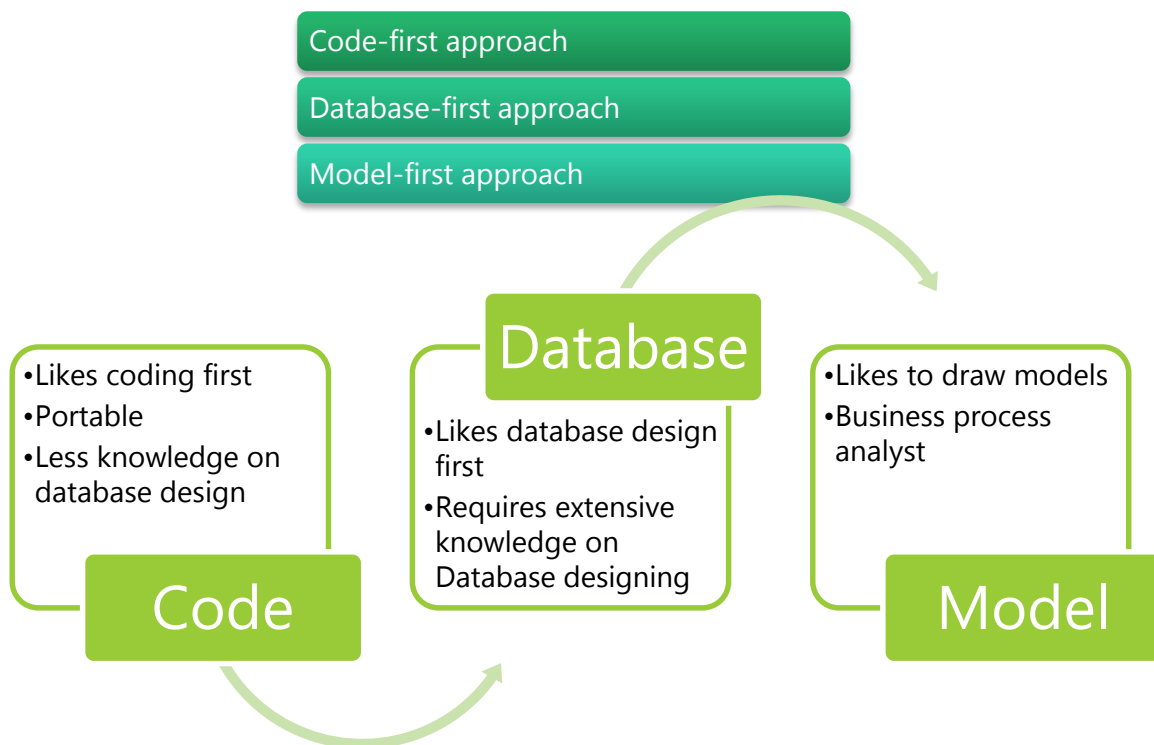


FIGURE 5.1: METHOD USAGES IT AND WHY USAGES IT (CODE-FIRST, DATABASE-FIRST, AND FINALLY MODEL FIRST)

Each one of these process has its pros and cons and based on that developer should pick what which is right for him or her.

A. **Code-first approach:**

Microsoft always shows code-first in their demos, as a result many developers adopting the *code-first approach* with little or no knowledge in database normalization. Thus, the lack of knowledge on normalization results

in inefficient design of the database. Plus every developer needs to write many lines of code to achieve the expected result, which does not come in handy. The best benefit of *code-first approach* is portability, one project code can be copied to another project with little refactoring.

B. Database-first approach:

People have misconception on this approach. This approach require building database first , however in MSDN it explains that if you have an existing database then you should use database first approach. However the main idea is quite different and mostly focuses on building database first. *Database-first approach* requires extensive knowledge in database design and end-results are efficient (if it has been done correctly) and requires less coding, finally enables rapid development. However, codes are not as portable as code-first.

C. Model-first approach:

Those who works in business domain will like this feature, it's mostly abstract from every other approach. Finally results are uncertain and very few users' usages this approach.

D. Decision for this framework:

ASP.NET MVC Bootstrapper usages both code-first and database-first approach to achieve the best of both worlds.

VI. PROBLEMS OF EXISTING FRAMEWORK (CHAPTER 6)

As a web development framework, the ASP.NET MVC [8] already fares better than many of its alternatives. With *clean separation of concerns*, *easy testability* and many other benefits ASP.NET MVC [8] brings to the table, even-though out of the box things are amazing; however, problems within the system are significant. Let us take look at few key points.

A. Inefficiency in built-in user registration process (ASP.NET [3] Identity)

While the ASP.NET MVC framework provides easy fixes for many problems, these *solutions can often be redundant*. For example, the platform offers a user management and registration system that can be better optimized. The inefficiency arising from the Microsoft-provided system has its roots in the database creation process, where the primary key for the user data is a string value. As a result, large scale user will decrease performance for sure and end result would be a large size database having small number of users.

B. Out of the box inflexibilities

ASP.NET MVC [8] has some great features (*scaffolding [14]*, *built-in user-management classes*, *ASP.NET identity [9]*, *user-role management [9]*), however these feature are not exactly optimized for real-world. As a result, great features become useless in enterprise world unless it has been tweaked, and tweaking

takes forever. For example, changing the user-registration string type to any number or any other types takes 500 lines+ coding. Scaffolding [14] generates things faster, however generated codes are not efficient. To modify scaffolding [14] developers have to modify T4 templates [21] which is very hard to modify and there is no support from Microsoft to edit T4 templates [21]. Third party (<http://www.devart.com/t4-editor>) resources are available to modify T4 templates [21], however those are requires payment to enable extended features. ***The most complicated part for T4 edit is that developer cannot find the error line number which may have bug and there is no right way to debug it.*** Hence, it takes months to modify simple functionality in T4 template [21].

C. Time zone detection

"A true creator is necessity, which is the mother of our invention." - Plato

In every popular framework or CMS or blogging system time zone is built-in. However, there is none as an out of the box solution. Moreover, it takes quite a lot of time to add this feature in one project. Hence, if we have several projects then 40% of the time might get wasted on this basic feature, which should be there in the first place. Time zone fixation requires worldwide country information, country code and relationships between country and time-zone. Unfortunately there is no website or service which serves both of those as a packages except <http://www.geonames.org>. Geonames.org serves unstructured

data for free users. They do have some process to put it in the mysql database. After putting it into the database, size becomes almost 4 gigabytes, database normalization has not been applied that is why it has gigantic size. Hence, it would be very inappropriate to push that 4 gigabytes of data with every project, as a result solution has to be found other than this one.

VII. PROJECT DESCRIPTION (CHAPTER 7)

"I begin with an idea and then it becomes something else."

- Pablo Picasso

ASP.NET MVC Bootstrapper was started without knowing to the founder. Solving similar problems over and over again leads to this amazing framework which will save a lot of time from every developer's workload by making smart use of her or his time. Quite a disproportionate chunk of time is spent in doing repetitive things: things that add little value but are indispensable to the system. It is rather ironic that the widely used platform needs to be prepared further to use it for every specific project. To eliminate such problems the ASP.NET MVC Bootstrapper *identifies the needs of the developers* that are not being fulfilled by the existing system, especially that of the developers with less depth of knowledge about the mechanism of ASP.NET MVC [8] framework who generally build websites out of the box using this framework.

VIII. FEATURES

(CHAPTER 8)

The purpose of the project is to make a developer's life easier. In trying to do so, the key problems of ASP.NET MVC [8] have been identified and dealt with. Comparing the modifiability or extendibility of the existing platforms generally used in web developing, ASP.NET MVC already fares better than many of its alternatives. Still, some *preset standardized templates are hard to modify* in ASP.NET MVC [8] that takes a lot of time of a developer. In order to increase the actual productivity of ASP.NET MVC [8] -based developers, this Bootstrapper sets up the platform where they will spend more time on new things rather than working on the repetitive occupational hazards they need to go through.

Features of the ASP.NET MVC Bootstrapper are as follows:

Feature	ASP.NET MVC [8]	ASP.NET MVC Bootstrapper
<i>ASP.NET Identity(user-management)</i>	(non-optimized)	(optimized)
<i>Identity(additional role-management)</i>	(doesn't exist)	(exist)
<i>Caching</i>	(exist but not applied)	(applied out of the box)
<i>Output-caching(HTML caching in server)</i>	(exist but not applied)	(applied out of the box)
<i>Page generation with icons and standard twitter bootstrap theme</i>	(partially applied)	(fully applied)
<i>YSlow performance</i>	(non-optimized)	(Optimized up to 95%)
<i>Worldwide country listing</i>	(doesn't exist)	(exist)
<i>Worldwide phone codes based on country</i>	(doesn't exist)	(exist)

<i>Time zone listing with Country relations</i>	(doesn't exist)	(exist)
<i>Worldwide language information</i>	(doesn't exist)	(exist)
<i>Based on database relationship specific component display (UX concepts)</i>	(doesn't exist)	(exist)
<i>Configurable Components</i>	(doesn't exist)	(exist)
<i>Efficient scaffolding : T4 Generation consist of pagination , searching and exporting to excel, json etc</i>	(doesn't exist)	(exist)
<i>Centralized configuration system</i>	(doesn't exist)	(exist)
<i>Ajax based uploading plugin</i>	(doesn't exist)	(exist)
<i>Ajax based hack-proof on keypress validator</i>	(doesn't exist)	(exist)
<i>Dynamic drop down component</i>	(doesn't exist)	(exist)
<i>Dependable dropdown component</i>	(doesn't exist)	(exist)
<i>Register user based on employee code</i>	(doesn't exist)	(exist)
<i>Hack-protections</i>	(exist but not configured)	(configured out of the box, please check the list)
<i>ELMAH (configured, authorized)</i>	(doesn't exist)	(exist)

FIGURE 8.1: FEATURES OF ASP.NET MVC BOOTSTRAPPER.

A. Reinventing the Wheel (Generic)

Much of the effort developers put in building up a website is spent on taming the system. This includes customizing the UX-poor framework to *suit the developer's needs* before actually starting the development process.

As an example, let us look at *designing a page for registration* of new users. Although a similar pattern for this particular action is followed in every website with little to no variety, developers need to develop these systems from scratch. Worse, even if a template for such function is not provided, it is often not optimized and greatly inefficient. In addition, modifying such preset

template is often really difficult and takes a lot of time and effort to make some small changes. However, ASP.NET MVC bootstrapper provides easier way to modify these components.

Moreover, consider *the uploading page* mechanism remains the same for any types of uploading that takes place in websites. Still, unfortunately, there remains no convenient way to efficiently implement a page for users to upload contents that, with easy modifications, suits the need of every developer. Likewise, inefficiency is found in user and role management system.

By default ASP.NET *Identity* [9] *database implementation is not optimized*, for example string is used as the primary key. To make the user experience more favorable and to equip the developer with the tool to do so, our Bootstrapper uses long data type as the primary key.

This also *saves the hassle of the repeated construction* of the similar functionalities and enables the developer to focus on more important things. Thus the concept of reinventing a wheel in its truest sense is made possible through this Bootstrapper.

B. Time-zone fix, Country, Phone and Language

ASP.NET MVC Bootstrapper provides time-zone fix modules as well as worldwide country information in convenient and optimized fashion. All a

developers needs is execute the zipped (400KB) SQL to get everything from out of the box.

C. UX and UI concepts

Framework components were designed and suited based on the book “Don’t make me think”. Which is a very popular book and explained many design and UX concepts to reduce confusions for worldwide clients and users.

The screenshot shows a web application interface for a registration form. At the top, there is a navigation bar with a home icon, the text 'We Review App', and links for 'Admin' and 'Contact Us'. Below the navigation bar, the title 'Register' is displayed, followed by the subtitle 'Create a new account for We Review App'. The form itself consists of several input fields, each with a label and a red asterisk indicating it is required. The fields are: 'User name*' with a user icon and the value 'newuser'; 'First Name*' with the value 'New'; 'Last Name*' with the value 'User'; 'Email Address*' with the value '@ newuser@gmail.com'; 'Password*' with masked characters '*****'; 'Confirm password*' with masked characters '*****'; 'Country*' with a dropdown menu showing a flag and the text '(+880) BANGLADESH'; 'Timezone*' with a dropdown menu showing '(UTC+06:00) ASTANA'; 'Language*' with a dropdown menu showing 'BENGALI - বাংলা'; and 'Phone*' with a phone icon and the value '880880100000000'. At the bottom of the form is a green button with a user icon and the text 'REGISTER'.

FIGURE 8.2: REGISTRATION UI AND UX EXAMPLE

For example, page with numerous fields to be filled by the user. This often drives away potential registered members. The Bootstrapper provides an easily customizable registration template to make use of as many fields necessary without much effort. To keep the potential member on the page without repelling him with an annoying registration form, this template makes a few fields appear gradually upon filling of the previous fields. This retains the user and lets him know how much is left, in a very user-friendly manner. In a way, this Bootstrapper *enhances the user experience (UX)* by equipping the developers with this interactive registration tool.

D. Greater Degree of Security

The ASP.NET MVC Bootstrapper has *several layers of protection* to ensure the security of the data and the system. This extension ensures an applied protection from multiple types of hacking, and although the system cannot claimed to be hack-proof, it can be asserted with reasonable confidence that this platform would withstand many strong attacks and mishaps and protect the data from manipulation. The modes of protection against many of the popular forms of hacking are not previously provided as a default in ASP.NET MVCs. Most importantly, the Bootstrapper it provides *reasonably strong protection against many known attacks*.

E. Efficient Scaffolding

Scaffolding greatly helps in organizing the system and can be really useful to save time. However, the scaffolding templates provided by Microsoft are very inefficient. This Bootstrapper provides *a flexible scaffolding system* with much more optimization that adds to the efficiency of a developer.

Scaffolding greatly helps in organizing the system and can be really useful to save time. However, the scaffolding templates provided by Microsoft are very inefficient. This Bootstrapper provides a flexible scaffolding system with much more optimization that adds to the efficiency of a developer.

F. Hassle-free Uploading

The ASP.NET MVC Bootstrapper allows a *smart uploading structure* that is at the same time makes efficient use of the time of a developer and also improves the user experience greatly. Unlike the generic uploading template that redirects the user multiple times to different pages, this template makes it possible for the user to upload all the necessary files (e.g. the original file, the thumbnail and cover images etc.) in a single page, thereby reducing the hassle and increasing the attractiveness of the website.

IX. RESULTS AND VERIFICATION (CHAPTER 9)

Full-fledged website's (<http://wereviewapp.com> [6]) prototype has been created within one week, which proves its efficiency and productivity in real world development scenario. However, later it was modified a lot and still it is not a finished product. I am regularly updating what is necessary to be added in the system for all developers around the globe.

X. **FUTURE WORK** **(CHAPTER 10)**

As developers prefer to make AJAX-based systems, the MVC Bootstrapper has plans to add AJAX compatibility in the near future. Full-fledged AJAX support will be made available soon in addition to further efficiencies in the code. The objective is to make an efficient and user-friendly Bootstrapper with the end-user in mind.

XI. ADDITIONAL PLUGINS AND RESOURCES USED

(CHAPTER 11)

Number of open source plugin and resources has been used to make this project successfully, however plugins are also modified to fit with this project and act as a component.

A. Front-End

1. Twitter Bootstrap [22] : <http://getbootstrap.com>
2. Bootstrap table [23]: <https://github.com/wenzhixin/bootstrap-table>
3. jQuery [24] : <https://jquery.com>
4. jQuery validation: <http://jqueryvalidation.org>
5. ASP.NET unobtrusive validation : <http://bit.ly/1AAifw9>
6. jQuery upload : <https://blueimp.github.io/jQuery-File-Upload>
7. jQuery upload modified by Alim UI Karim : <http://bit.ly/1AAhNxM>
8. Star rating plugin : <http://plugins.krajee.com/star-rating>
9. Underscore js : <http://underscorejs.org>

A. Backend

1. DevMVCComponent [15] (Developed by Alim UI Karim): It is the core component of the framework, which deals with error reporting, database pagination, caching and sessions handling and many more things.

2. ASP.NET [3] : Framework by Microsoft to build web applications drag-drop abilities.
3. ASP.NET MVC [8] : Framework designed by Microsoft leveraging ASP.NET, using MVC architecture.

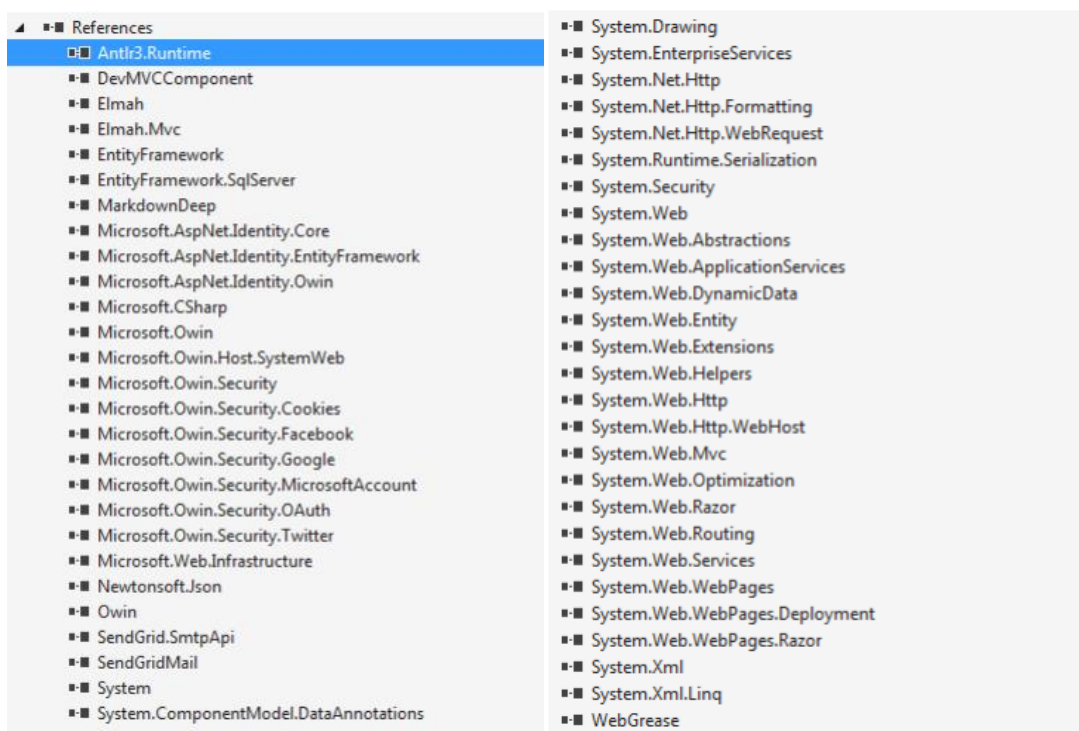


FIGURE 5: BACKEND ADDITIONAL PACKAGES TO RUN ASP.NET BOOTSTRAPPER

There are also several packages are list in the github packages config bit.ly/boot-proj-config .

XII. SOURCE CODE (CHAPTER 12)

The community-oriented Bootstrapper source code is available on GitHub (<http://j.mp/asp-net-boot>). This framework is freely available online and updated frequently to add more features and squash bugs. There are several steps should be follow when installing this framework. There is an installation video [25] , which can help developers quickly install this framework right away (<http://bit.ly/how-asp-boot>).

XIII. **CONCLUSION** **(CHAPTER 13)**

An average developer to this date has been a geek toiling away on their keyboards to make things possible. Their life, with all the opportunities to do amazingly diverse and powerful things, has still been miserable. Rather than being brilliant strategists who see through the maze of complex things, they are reduced to some coding soldiers who are required to spend a ridiculous chunk of their time in tedious adjustments.

The ASP.NET MVC Bootstrapper is a small step to put an end such repetition of mind numbing monotony. It is to aid the developers to bring about efficiency, in their websites and their lives. It is to make a developer's workday shorter and smarter.

XIV. REFERENCES

(CHAPTER 14)

- [1] "Orchard Project - Wikipedia, the free encyclopedia," Microsoft Developers, 1 July 2014. [Online]. Available: http://en.wikipedia.org/wiki/Orchard_Project.
- [2] "WordPress - Wikipedia, the free encyclopedia," WordPress Foundation, 27 April 2015. [Online]. Available: <http://en.wikipedia.org/wiki/WordPress>.
- [3] Microsoft, "ASP.NET - Wikipedia, the free encyclopedia," 2014.
- [4] aukgit, "aukgit/ASP.NET-MVC-Bootstrapper," [Online]. Available: <https://github.com/aukgit/ASP.NET-MVC-Bootstrapper>.
- [5] Wikipedia, ".NET Framework - Wikipedia, the free encyclopedia," Microsoft, [Online]. Available: https://en.wikipedia.org/wiki/.NET_Framework.
- [6] "We Review App : App Review Website," [Online]. Available: <http://wereviewapp.com/>.
- [7] "Commits · aukgit/WereViewProject," [Online]. Available: <http://bit.ly/1Rv6akH>.
- [8] Microsoft, "ASP.NET MVC Framework - Wikipedia, the free encyclopedia," 12 November 2014. [Online]. Available: http://en.wikipedia.org/wiki/ASP.NET_MVC_Framework.
- [9] R. A. T. D. Pranav R. and J. G., "Introduction to ASP.NET Identity | The ASP.NET Site," Microsoft, 17 October 2013. [Online]. Available: <http://www.asp.net/identity/overview/getting-started/introduction-to-aspnet-identity>.
- [10] Wikipedia, "Model–view–controller - Wikipedia, the free encyclopedia," [Online]. Available: <http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>.
- [11] Wikipedia, "Microsoft Visual Studio - Wikipedia, the free encyclopedia," Microsoft, 12 November 2014. [Online]. Available: http://en.wikipedia.org/wiki/Microsoft_Visual_Studio.
- [12] Wikipedia, "NuGet - Wikipedia, the free encyclopedia," Microsoft / Outercurve Foundation, 30 March 2015. [Online]. Available: <http://en.wikipedia.org/wiki/NuGet>.
- [13] S. Krug, "Don't Make Me Think - Wikipedia, the free encyclopedia," 2013. [Online]. Available: http://en.wikipedia.org/wiki/Don%27t_Make_Me_Think.
- [14] T. FitzMacken, "ASP.NET Scaffolding in Visual Studio 2013 | The ASP.NET Site," Microsoft, 9 April 2014. [Online]. Available: <http://www.asp.net/visual-studio/overview/2013/aspnet-scaffolding-overview>.
- [15] M. A. U. Karim, "aukgit/DevMvcComponent · GitHub," August 2015. [Online]. Available: <https://github.com/aukgit/DevMVCComponent>.
- [16] A. A., "elmah - Error Logging Modules and Handlers for ASP.NET - Google Project Hosting," Google, [Online]. Available: <https://code.google.com/p/elmah/>.
- [17] "Ruby on Rails," [Online]. Available: <http://rubyonrails.org/>.
- [18] "Laravel - Wikipedia, the free encyclopedia," Taylor Otwell, 4 April 2015. [Online]. Available: <http://en.wikipedia.org/wiki/Laravel>.
- [19] "Convention over configuration - Wikipedia, the free encyclopedia," [Online]. Available: http://en.wikipedia.org/wiki/Convention_over_configuration.

- [20] A. U. Karim, "ASP NET MVC Basics 101," 3 May 2015. [Online]. Available: <http://bit.ly/asp-net-mvc-basic>.
- [21] Wikipedia, "Text Template Transformation Toolkit - Wikipedia, the free encyclopedia," Microsoft, 2005. [Online]. Available: http://en.wikipedia.org/wiki/Text_Template_Transformation_Toolkit.
- [22] Wikipedia, "Bootstrap (front-end framework) - Wikipedia, the free encyclopedia," 16 March 2015. [Online]. Available: http://en.wikipedia.org/wiki/Bootstrap_%28front-end_framework%29.
- [23] "wenzhixin/bootstrap-table · GitHub," [Online]. Available: <https://github.com/wenzhixin/bootstrap-table>.
- [24] J. Resig, "jQuery - Wikipedia, the free encyclopedia," jQuery Team, 28 April 2015. [Online]. Available: <http://en.wikipedia.org/wiki/JQuery>.
- [25] A. U. Karim, "ASP NET Bootstrapper : How to Install it (beta 0.2)," Developers Organism, 4 Apr 2015. [Online]. Available: <http://bit.ly/how-asp-boot>.

XV. CODE COMMITS (CHAPTER 15)

GitHub This repository Search Explore Features Enterprise Pricing Sign up Sign in

aaukgit / ASP.NET-MVC-Bootstrapper Watch 2 Star 2 Fork 0

A framework and project template that can change developers life.

259 commits 4 branches 2 releases 1 contributor

Branch: master ASP.NET-MVC-Bootstrapper / +

Fix less and login page

aaukgit authored 17 days ago latest commit 4e05521297

DevBootstrapper	Fix less and login page	17 days ago
.gitattributes	Added .gitattributes & .gitignore files	7 months ago
.gitignore	Git ignore	a month ago
ASP.MVC DevBootstrapper 2.0.zip	Project Template 2.0 added	6 months ago
DevBootstrapper.zip	beta 0.2 Project Template is also added	6 months ago
Presentation.pptx	presentation added	6 months ago
Uploading.png	proj	6 months ago
changing2.txt	editor templates final changed	3 months ago
replacing.txt	editor templates final changed	3 months ago

Code

Issues 0

Pull requests 0

Pulse

Graphs

HTTPS clone URL

https://github.com/aaukgit/ASP.NET-MVC-Bootstrapper

You can clone with HTTPS or Subversion.

Clone in Desktop

Download ZIP

FIGURE 15.1: PROJECT COMMITS IN GITHUB

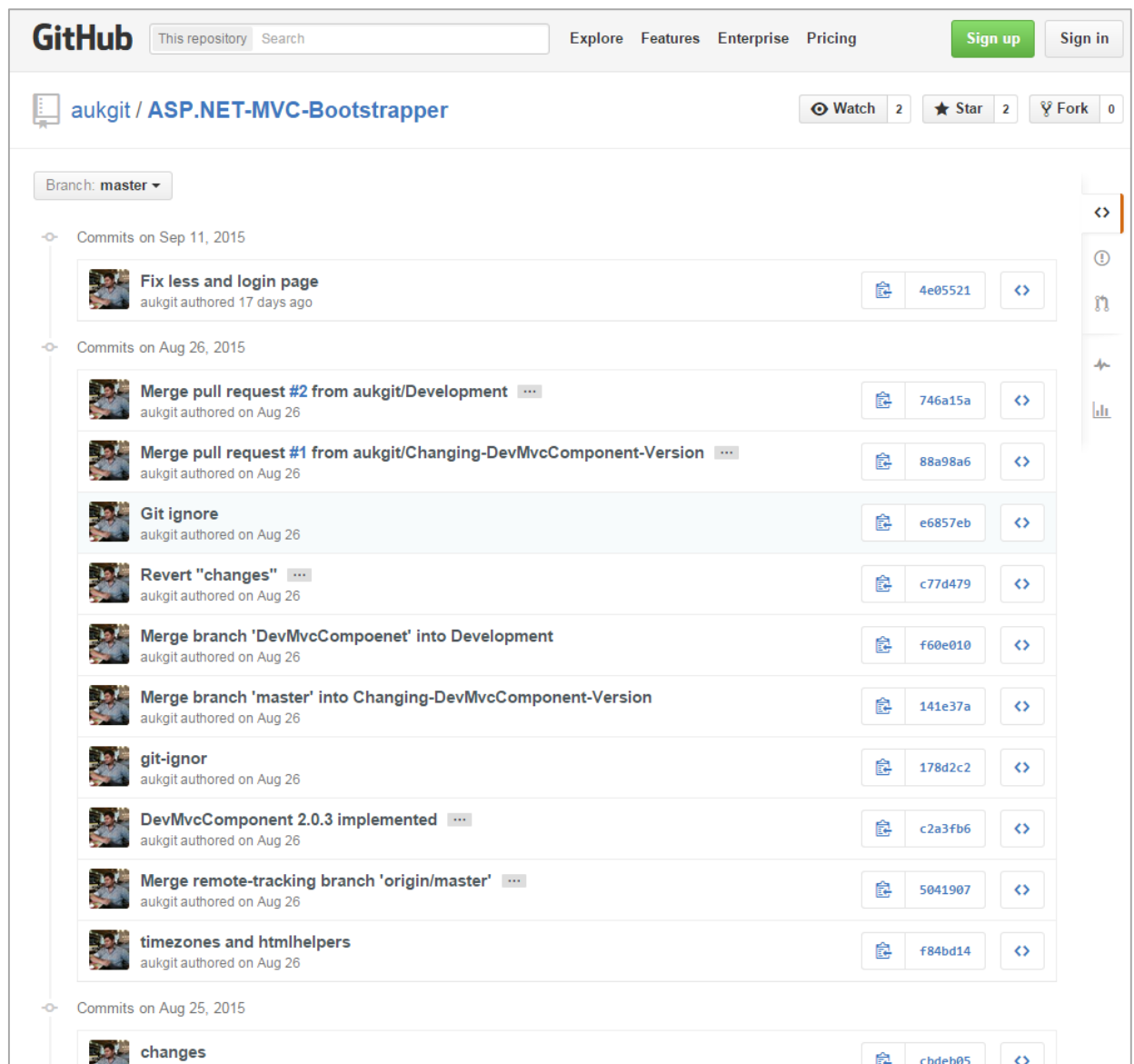


FIGURE 15.2: PROJECT COMMITS IN GITHUB WITH DATES ([BIT.LY/1YIq00C](https://bit.ly/1YIq00C))

XVI. NOTABLE CODE EXAMPLES (CHAPTER 16)

A. Timezone code examples

```

public class Zone {

    #region Fields

    private static string _defaultTimeFormat = "hh:mm:ss tt";
    private static string _defaultDateFormat = "dd-MMM-yy";
    private static string _defaultDateTimeFormat = "dd-MMM-yy hh:mm:ss tt";

    #endregion

    #region Properties

    /// <summary>
    ///     hh:mm:ss tt
    /// </summary>
    public static string TimeFormat {
        get { return _defaultTimeFormat; }
        set { _defaultTimeFormat = value; }
    }

    /// <summary>
    ///     dd-MMM-yy
    /// </summary>
    public static string DateFormat {
        get { return _defaultDateFormat; }
        set { _defaultDateFormat = value; }
    }

    /// <summary>
    ///     dd-MMM-yy
    /// </summary>
    public static string DateTimeFormat {
        get { return _defaultDateTimeFormat; }
        set { _defaultDateTimeFormat = value; }
    }

    private static readonly ReadOnlyCollection<TimeZoneInfo> SystemTimeZones =
    TimeZoneInfo.GetSystemTimeZones();
    private static List<UserTimeZone> _dbTimeZones;

    #endregion

    #region Constructor

    public Zone() {
    }

    public Zone(string timeFormat, string dateFormat = null, string dateTimeFormat
    = null) {
        _defaultTimeFormat = timeFormat;
        if (dateFormat != null) {
            _defaultDateFormat = dateFormat;
        }
        if (dateTimeFormat != null) {
            _defaultDateTimeFormat = dateTimeFormat;
        }
    }

    #endregion
}

```

```

#region Application Startup function for database

public static void LoadTimeZonesIntoMemory() {
    _dbTimeZones = CachedQueriedData.GetTimezones();
}

#endregion

/// <summary>
///     Flush cache information about user time-zone.
/// </summary>
/// <param name="log"></param>
public static void RemoveTimeZoneCache(string log) {
    if (log == null) {
        return;
    }
    AppConfig.Caches.Remove(CookiesNames.ZoneInfo + log);
}

#region Dynamic Timing
/// <summary>
///     Returns a dynamic string value using time and other logics.
/// </summary>
/// <returns>Always get a unique string using date.</returns>
public static string GetTimeDynamic() {
    var dynamic = DateTime.Now.Millisecond + DateTime.Now.Second +
DateTime.Now.Minute +
        DateTime.Now.Millisecond;

    return DateTime.Now.ToShortTimeString() + dynamic + (dynamic ^ dynamic);
}

#endregion

#region Get Zone from Cache

/// <summary>
///     Get UserTimeZone from database using caching if possible.
/// </summary>
/// <param name="zone">Pass TimeZoneInfo to get the usertimezone from
database.</param>
/// <returns>Returns timezone from cache if possible if not found anywhere
returns null.</returns>
public static UserTimeZone Get(TimeZoneInfo zone) {
    var id = "timezone-id:" + zone.Id;
    var userTimeZone = (UserTimeZone)AppConfig.Caches.Get(id);
    if (userTimeZone == null) {
        userTimeZone = _dbTimeZones.FirstOrDefault(n => n.InfoID == zone.Id);
        AppConfig.Caches.Set(id, userTimeZone);
    }
    return userTimeZone;
}

/// <summary>
///     Get timezone by userid.
/// </summary>
/// <param name="userId"></param>
/// <returns>Returns timezone from cache if possible if not found anywhere
returns null.</returns>
public static TimeZoneSet Get(long userId) {
    TimeZoneSet timeZoneInfo = null;
    var idString = "-id:" + userId;
    timeZoneInfo = GetSavedTimeZone(idString);
    if (timeZoneInfo != null) {
        //got time zone from cache.
        return timeZoneInfo;
    }
    //if cache time zone not exist.
    var user = UserManager.GetUser(userId);
    if (user != null) {
        var timeZoneDb = _dbTimeZones.FirstOrDefault(n => n.UserTimeZoneID ==
user.UserTimeZoneID);

```



```

        if (timezoneDb != null) {
            timeZoneInfo = new TimeZoneSet();
            timeZoneInfo.UserTimeZone = timezoneDb;
            timeZoneInfo.TimeZoneInfo = SystemTimeZones.FirstOrDefault(n =>
n.Id == timezoneDb.InfoID);
        }
        if (timeZoneInfo != null && timeZoneInfo.TimeZoneInfo != null) {
            // Save the time zone to the cache.
            SaveTimeZone(timeZoneInfo, idString);
            return timeZoneInfo;
        }
        return null;
    }
    /// <summary>
    ///     Optimized fist check on cache then database.
    ///     Get current logged time zone from database or from cache.
    /// </summary>
    /// <returns>Returns time zone of the user.</returns>
    public static TimeZoneSet Get() {
        if (!HttpContext.Current.User.Identity.IsAuthenticated) {
            return null;
        }
        var log = HttpContext.Current.User.Identity.Name;
        return Get(log);
    }

    /// <summary>
    ///     Optimized fist check on cache then database.
    ///     Get time zone from database base on user name.
    /// </summary>
    /// <param name="username"></param>
    /// <returns>Returns time zone of the user.</returns>
    public static TimeZoneSet Get(string username) {
        TimeZoneSet timeZoneInfo = null;
        timeZoneInfo = GetSavedTimeZone(username);
        if (timeZoneInfo != null) {
            //got time zone from cache.
            return timeZoneInfo;
        }
        //if cache time zone not exist.
        var user = UserManager.GetUser(username);
        if (user != null) {
            var timezoneDb = _dbTimeZones.FirstOrDefault(n => n.UserTimeZoneID ==
user.UserTimeZoneID);
            if (timezoneDb != null) {
                timeZoneInfo = new TimeZoneSet();
                timeZoneInfo.UserTimeZone = timezoneDb;
                timeZoneInfo.TimeZoneInfo = SystemTimeZones.FirstOrDefault(n =>
n.Id == timezoneDb.InfoID);
            }
            if (timeZoneInfo != null && timeZoneInfo.TimeZoneInfo != null) {
                // Save the time zone to the cache.
                SaveTimeZone(timeZoneInfo, username);
                return timeZoneInfo;
            }
        }
        return null;
    }

    /// <summary>
    ///     Get time zone from save cache.
    /// </summary>
    /// <param name="log"></param>
    /// <returns></returns>
    private static TimeZoneSet GetSavedTimeZone(string log) {
        //save to cookie
        if (!String.IsNullOrEmpty(log)) {
            var cZone = (TimeZoneSet)AppConfig.Caches.Get(CookiesNames.ZoneInfo +
log);

            return cZone; //fast
        }
    }

```

```

        return null;
    }

#endregion

#region Save Zone in Cache

    /// <summary>
    ///     Saved for current logged user.
    /// </summary>
    /// <param name="timeZoneInfo"></param>
    private static void SaveTimeZone(TimeZoneSet timeZoneInfo) {
        if (!HttpContext.Current.User.Identity.IsAuthenticated) {
            return;
        }
        var log = HttpContext.Current.User.Identity.Name;
        SaveTimeZone(timeZoneInfo, log);
    }

    private static void SaveTimeZone(TimeZoneSet timeZoneInfo, string log) {
        if (log == null || timeZoneInfo == null) {
            return;
        }
        //save to cache
        AppConfig.Caches.Set(CookiesNames.ZoneInfo + log, timeZoneInfo);
    }

#endregion

#region Get times format based on zone
    /// <summary>
    ///
    /// </summary>
    /// <param name="userId"></param>
    /// <param name="dt"></param>
    /// <param name="format"></param>
    /// <param name="addTimeZoneString">Add timezone string with Date. Eg. 26-Aug-
    2015 (GMT -07:00)</param>
    /// <returns></returns>
    public static string GetDateTime(
        long userId,
        DateTime? dt,
        string format = null,
        bool addTimeZoneString = true) {
        if (format == null) {
            format = DateTimeFormat;
        }
        var zone = Get(userId);
        return GetDateTime(zone, dt, format, addTimeZoneString);
    }

    /// <summary>
    ///     Get date to print as string.
    ///     Time zone by user logged in.
    ///     It will get the logged user and then get the time-zone and then print.
    /// </summary>
    /// <param name="userId"></param>
    /// <param name="dt"></param>
    /// <param name="format">if format null then default format.</param>
    /// <param name="addTimeZoneString">Add timezone string with Date. Eg. 26-Aug-
    2015 (GMT -07:00)</param>
    /// <returns>Returns nice string format based on logged user's selected time
    zone.</returns>
    public static string GetTime(
        long userId,
        DateTime? dt,
        string format = null,
        bool addTimeZoneString = true) {
        if (format == null) {
            format = TimeFormat;
        }
        var zone = Get(userId);
        return GetDateTime(zone, dt, format);
    }

```

```

    }
    /// <summary>
    ///     Get date to print as string.
    ///     Time zone by user logged in.
    ///     It will get the logged user and then get the time-zone and then print.
    /// </summary>
    /// <param name="dt"></param>
    /// <param name="format">if format null then default format.</param>
    /// <param name="addTimeZoneString">Add timezone string with Date. Eg. 26-Aug-
2015 (GMT -07:00)</param>
    /// <returns>Returns nice string format based on logged user's selected time
zone.</returns>
    public static string GetTime(
        DateTime? dt,
        string format = null,
        bool addTimeZoneString = true) {
        if (format == null) {
            format = TimeFormat;
        }
        return GetDateTime(dt, format, addTimeZoneString);
    }

    /// <summary>
    ///     No time zone required.
    /// </summary>
    /// <param name="dt"></param>
    /// <param name="format">if format null then default format.</param>
    /// <param name="addTimeZoneString">Add timezone string with Date. Eg. 26-Aug-
2015 (GMT -07:00)</param>
    /// <returns>Returns nice string format based on logged user's selected time
zone.</returns>
    public static string GetDate(
        DateTime? dt,
        string format = null,
        bool addTimeZoneString = true) {
        if (format == null) {
            format = DateFormat;
        }
        return GetDateTime(dt, format, addTimeZoneString);
    }

    /// <summary>
    ///     Get date to print as string.
    ///     Time zone by user logged in.
    ///     It will get the logged user and then get the time-zone and then print.
    /// </summary>
    /// <param name="dt">Returns "" if null</param>
    /// <param name="format">if format null then default format.</param>
    /// <param name="addTimeZoneString">Add timezone string with Date. Eg. 26-Aug-
2015 (GMT -07:00)</param>
    /// <returns>Returns nice string format based on logged user's selected time
zone. If no logged user then default datetime.</returns>
    public static string GetDateTime(
        DateTime? dt,
        string format = null,
        bool addTimeZoneString = true) {
        if (format == null) {
            format = DateTimeFormat;
        }
        var timeZone = Get();
        return GetDateTime(timeZone, dt, format, addTimeZoneString);
    }
}

#endregion

#region Based on timezone

    /// <summary>
    ///     Get date to print as string.
    ///     Time zone by user logged in.
    ///     It will get the logged user and then get the time-zone and then print.
    /// </summary>
    /// <param name="timeZone"></param>

```

```

    /// <param name="dt"></param>
    /// <param name="format">if format null then default format.</param>
    /// <param name="addTimeZoneString">Add timezone string with Date. Eg. 26-Aug-
2015 (GMT -07:00)</param>
    /// <returns>Returns nice string format based on logged user's selected time
zone.</returns>
    public static string GetTime(
        TimeZoneSet timeZone,
        DateTime? dt,
        string format = null,
        bool addTimeZoneString = true) {
        if (format == null) {
            format = TimeFormat;
        }
        return GetDateTime(timeZone, dt, format, addTimeZoneString);
    }

    /// <summary>
    ///     No need to convert dates based on time zones.
    /// </summary>
    /// <param name="timeZone"></param>
    /// <param name="dt"></param>
    /// <param name="format">if format null then default format.</param>
    /// <param name="addTimeZoneString">Add timezone string with Date. Eg. 26-Aug-
2015 (GMT -07:00)</param>
    /// <returns>Returns nice string format based on logged user's selected time
zone.</returns>
    public static string GetDate(
        TimeZoneSet timeZone,
        DateTime? dt,
        string format = null,
        bool addTimeZoneString = true) {
        if (format == null) {
            format = DateFormat;
        }
        return GetDateTime(timeZone, dt, format, addTimeZoneString);
    }

    /// <summary>
    ///     Get date time to print as string.
    ///     Time zone by user logged in.
    ///     It will get the logged user and then get the time-zone and then print.
    /// </summary>
    /// <param name="format">if format null then default format.</param>
    /// <param name="addTimeZoneString">Add timezone string with Date. Eg. 26-Aug-
2015 (GMT -07:00)</param>
    /// <returns>Returns nice string format based on logged user's selected time
zone.</returns>
    public static string GetCurrentDateTime(string format = null, bool
addTimeZoneString = true) {
        return GetDateTime(DateTime.Now, format, addTimeZoneString);
    }

    /// <summary>
    ///     Get date time to print as string.
    ///     Time zone by user logged in.
    ///     It will get the logged user and then get the time-zone and then print.
    /// </summary>
    /// <param name="format">if format null then default format.</param>
    /// <param name="addTimeZoneString">Add timezone string with Date. Eg. 26-Aug-
2015 (GMT -07:00)</param>
    /// <returns>Returns nice string format based on logged user's selected time
zone.</returns>
    public static string GetCurrentDate(
        string format = null,
        bool addTimeZoneString = true) {
        return GetDate(DateTime.Now, format, addTimeZoneString);
    }

    /// <summary>
    ///     Get date to print as string.
    ///     Time zone by user logged in.

```

```

    /// It will get the logged user and then get the time-zone and then print.
    /// </summary>
    /// <param name="timeZone"></param>
    /// <param name="dt"></param>
    /// <param name="format">if format null then default format.</param>
    /// <param name="addTimeZoneString">Add timezone string with Date. Eg. 26-Aug-
2015 (GMT -07:00)</param>
    /// <returns>Returns nice string format based on logged user's selected time
zone.</returns>
    public static string GetDateTime(
        TimeZoneSet timeZone,
        DateTime? dt,
        string format = null,
        bool addTimeZoneString = true) {
        if (dt == null) {
            return "";
        }
        var dt2 = (DateTime)dt;

        if (format == null) {
            format = DateTimeFormat;
        }
        if (timeZone == null || !timeZone.IsTimeZoneInfoExist()) {
            return dt2.ToString(format);
        }
        var currentZone = timeZone.TimeZoneInfo;
        //time zone found.
        var newDate = TimeZoneInfo.ConvertTime(dt2, currentZone);
        string additionalString = "";
        if (addTimeZoneString) {
            var userZone = timeZone.UserTimezone;
            additionalString = "(" + userZone.TimePartOnly + ")";
        }
        return newDate.ToString(format) + additionalString;
    }

    #endregion
}

```

FIGURE 16.1: TIMEZONE MANAGEMENT CODE

B. Helper Methods for all Developers

```

public static class HtmlHelpers {
    private const string Selected = "selected='selected'";
    public static int TruncateLength = AppConfig.TruncateLength;

    #region Icons generate : badge

    public static HtmlString GetBadge(this HtmlHelper helper, long number) {
        var markup = string.Format(@"<span class='badge'>{0}</span>", number);

        return new HtmlString(markup);
    }

    #endregion

    #region Generate Navigation

    public static HtmlString GetMenu(this HtmlHelper helper, string menuName, bool
isDependOnUserLogState = false) {
        var cacheName = menuName + "-menu-";
        if (isDependOnUserLogState && UserManager.IsAuthenticated()) {
            cacheName += UserManager.GetCurrentUserName();
        }
        var cache = (string)AppConfig.Caches.Get(cacheName);

        if (cache != null && !string.IsNullOrWhiteSpace(cache)) {

```

```

        return new HtmlString(cache);
    }

    using (var menuGenerator = new GenerateMenu()) {
        var menuItems = menuGenerator.GetMenuItem(menuName);

        if (menuItems != null && menuItems.NavigationItems != null) {
            var items = menuItems.NavigationItems.ToList();
            var menuListItems =
menuGenerator.GenerateRecursiveMenuItems(items);
            // keeping cache
            AppConfig.Caches.Set(cacheName, menuListItems);
            return new HtmlString(menuListItems);
        }
    }
    return new HtmlString("");
}

#endregion

#region List, Item Generate / Route Generates

public static HtmlString RoutelistItemGenerate(this HtmlHelper helper, string
area, string display,
string controller, string currentController) {
    var addClass = " class='active' ";
    if (controller != currentController)
        addClass = "";
    var markup = string.Format("<li{0}><a href='{1}'>{2}</a></li>", addClass,
"/" + area + "/" + controller,
display);
    //return new HtmlString(markup);
    return new HtmlString(markup);
}

#endregion

#region Confirming Buttons
/// <summary>
/// Confirms before submit.
/// </summary>
/// <param name="helper"></param>
/// <param name="buttonName"></param>
/// <param name="alertMessage"></param>
/// <returns></returns>
public static HtmlString ConfirmingSubmitButton(this HtmlHelper helper, string
buttonName = "Save",
string alertMessage = "Are you sure about this action?") {
    var sendbtn = string.Format(
        "<input type='submit' value='{0}' onClick='return
confirm('{1}');' />",
        buttonName, alertMessage);
    return new HtmlString(sendbtn);
}

#endregion

#region jQueryMobile Options
/// <summary>
/// JQueryMobile BackButton
/// </summary>
/// <param name="helper"></param>
/// <param name="buttonName"></param>
/// <param name="icon"></param>
/// <returns></returns>
public static HtmlString BackButton(this HtmlHelper helper, string buttonName =
"Back", bool isMini = false,
string icon = "arrow-l") {
    var mini = (isMini)
        ? "data-mini='true'"
        : "";

```

```

        var backbtn = "<a href='#' data-role='button' class = 'back-button' data-
rel='back' data_icon='" + icon +
        "" " + mini + ">" + buttonName + "</a>";
        return new HtmlString(backbtn);
    }

#endregion

#region Drop Downs: General, Country

#region Country
/// <summary>
/// </summary>
/// <param name="countries"></param>
/// <param name="classes">use spaces to describe the classes</param>
/// <param name="otherAttributes"></param>
/// <returns></returns>
public static string DropDownCountry(List<Country> countries, string classes =
"",
    string otherAttributes = "", string contentAddedString = "") {
    var countryOptionsGenerate = "<select class='form-control selectpicker " +
classes +
                                " country-combo' data-live-search='true'
name='CountryID' " + otherAttributes +
                                " title='Country' data-style='btn-success
flag-combo fc-af'>";
    var sb = new StringBuilder(countryOptionsGenerate, countries.Count * 7);
    foreach (var country in countries) {
        sb.Append(string.Format("<option class='flag-country-combo flag {0}'
title='| {1}' value='{2}'>",
                                country.Alpha2Code.ToLower(), country.DisplayCountryName,
country.CountryID));
        sb.Append(contentAddedString);
        //sb.Append();
        sb.Append(country.DisplayCountryName);
        sb.Append("</option>").AppendLine();
    }
    sb.AppendLine("</select>");
    return sb.ToString();
}
/// <summary>
/// </summary>
/// <param name="helper"></param>
/// <param name="countries"></param>
/// <param name="classes">use spaces to describe the classes</param>
/// <param name="otherAttributes"></param>
/// <returns></returns>
public static HtmlString DropDownCountry(this HtmlHelper helper, List<Country>
countries, string classes = "",
    string otherAttributes = "", string contentAddedString = "") {
    string strHtml = DropDownCountry(countries, classes, otherAttributes,
contentAddedString);
    return new HtmlString(strHtml);
}

#endregion

#region General DropDowns

public static HtmlString DropDowns(this HtmlHelper helper, string valueField,
string textField,
    string htmlName = null, string displayName = null, string modelValue =
null, string isRequired = "*",
    string classes = null, string tooltipValue = null, string otherAttributes =
"", string tableName = null,
    ConnectionStrings.ConnectionStringType connectionType =
ConnectionStrings.ConnectionStringType.DefaultConnection) {
    var divElement = @"<div class='form-group {0}-main'>
                        <div class='controls'>
                            <label class='col-md-2 control-label'
for='{0}'>{1}<span class='red '>{2}</span></label>
                            <div class='col-md-10 {0}-combo-div'>
                                {3}

```

```

<a href='#' data-toggle='tooltip' data-original-
title='{4}' title='{4}' class='tooltip-show'><span class='glyphicon glyphicon-question-sign
font-size-22 glyphicon-top-fix almost-white'></span></a>
</div>
</div>
</div>";

// 0- name
// 1- displayName
// 2 - isRequired
// 3 - select element
// 4 - tooltipValue
if (tableName == null) {
    tableName = valueField.Replace("ID", "");
}
if (modelValue == null) {
    modelValue = "";
}
if (classes == null) {
    classes = "btn btn-success";
}
if (displayName == null) {
    displayName = textField;
}
if (toolTipValue == null) {
    toolTipValue = "Please select " + displayName;
}
if (htmlName == null) {
    htmlName = valueField;
}
var selected = "";
var countryOptionsGenerate = "<select class='form-control selectpicker " +
classes +
" ' data-live-search='true' name='" + htmlName
+ " ' title='Choose...' data-style='" + classes +
">";
var dt = CachedQueriedData.GetTable(tableName, connectionType, new[] {
valueField, textField });
if (dt != null && dt.Rows.Count > 0) {
    var sb = new StringBuilder(countryOptionsGenerate, dt.Rows.Count + 10);
    DataRow row;
    for (var i = 0; i < dt.Rows.Count; i++) {
        row = dt.Rows[i];
        if (row[valueField].Equals(modelValue)) {
            selected = Selected;
        }
        sb.Append(string.Format("<option value='{0}' {1} {2}>{2}</option>",
row[valueField], selected,
row[textField]));
    }
    sb.AppendLine("</select>");
    var complete = string.Format(divElement, htmlName, displayName,
isRequired, sb, tooltipValue);

    return new HtmlString(complete);
}
return new HtmlString("");
}

#endregion

#endregion

#region Truncates

public static string Truncate(this HtmlHelper helper, string input, int?
length, bool isShowEllipseDot = true) {
    if (string.IsNullOrEmpty(input))
        return "";
    if (length == null) {
        length = TruncateLength;
    }
    if (input.Length <= length) {

```



```

        return input;
    }
    if (isShowEllipseDot) {
        return input.Substring(0, (int)length) + "...";
    }
    return input.Substring(0, (int)length);
}

public static string Truncate(this HtmlHelper helper, string input, int
starting, int length) {
    if (string.IsNullOrEmpty(input))
        return "";
    if (length == -1) {
        length = input.Length;
    }
    if (input.Length <= length) {
        length = input.Length;
    }
    length = length - starting;
    if (input.Length < starting) {
        return "";
    }
    return input.Substring(starting, length);
}

public static bool IsTruncateNeeded(this HtmlHelper helper, string input, int
mid) {
    if (string.IsNullOrEmpty(input))
        return false;
    if (input.Length > mid) {
        return false;
    }
    return true;
}

#endregion

#region Link Generates

public static HtmlString ContactFormActionLink(this HtmlHelper helper, string
linkName, string title,
string addClass = "") {
    var markup = string.Format(MailHtml.ContactUsLink, title, linkName,
addClass, AppVar.Url);
    return new HtmlString(markup);
}

public static HtmlString GetCurrentPageUrlAnchorTag(this HtmlHelper helper,
string linkName, string title, bool h1 = true,
string addClass = "") {
    //var area =
HttpContext.Current.Request.RequestContext.RouteData.DataTokens["area"];
    //var controller =
HttpContext.Current.Request.RequestContext.RouteData.Values["controller"];
    //var action =
HttpContext.Current.Request.RequestContext.RouteData.Values["action"];

    var markup = "";

    //if (area != null) {
    //    markup = string.Format("<a href='{0}/{1}/{2}' class='{3}'
title='{4}'>{5}</a>", area, controller, action, addClass, title, linkName);
    //} else {
    //    markup = string.Format("<a href='{0}/{1}' class='{2}'
title='{3}'>{4}</a>", controller, action, addClass, title, linkName);
    //}
    var uri = HttpContext.Current.Request.RawUrl;
    uri = AppVar.Url + uri;
    markup = string.Format("<a href='{0}' class='{1}' title='{2}'>{3}</a>",
uri, addClass, title, linkName);
    if (h1) {
        markup = string.Format("<h1 title='{0}'>{1}</h1>", title, markup);
    }
}

```

```

    }
    return new HtmlString(markup);
}

/// <summary>
/// Returns url without the host name.
/// Slash is included
/// </summary>
/// <param name="helper"></param>
/// <returns>Returns url without the host name.</returns>
public static string GetCurrentPageUrl(this HtmlHelper helper) {
    return HttpContext.Current.Request.RawUrl;
}

/// <summary>
/// Returns url whole page url with the host name.
/// </summary>
/// <param name="helper"></param>
/// <returns>Returns url whole page url with the host name. </returns>
public static string GetCurrentUrlWithHostName(this HtmlHelper helper) {
    return AppVar.Url + HttpContext.Current.Request.RawUrl;
}

/// <summary>
/// </summary>
/// <param name="helper"></param>
/// <param name="linkName">null gives the number on the display</param>
/// <param name="title"></param>
/// <param name="number"></param>
/// <param name="h1"></param>
/// <param name="addClass"></param>
/// <returns></returns>
public static HtmlString PhoneNumberLink(this HtmlHelper helper, long
phonenummer, string title, bool h1 = false,
    string addClass = "") {
    var phone = "+" + phonenummer;

    var markup = string.Format("<a href='tel:{0}' class='{1}'
title='{2}'>{3}</a>", phone, addClass, title,
    phone);

    if (h1) {
        markup = string.Format("<h1 title='{0}'>{1}</h1>", title, markup);
    }
    return new HtmlString(markup);
}

/// <summary>
/// </summary>
/// <param name="helper"></param>
/// <param name="linkName">null gives the number on the display</param>
/// <param name="title"></param>
/// <param name="number"></param>
/// <param name="h1"></param>
/// <param name="addClass"></param>
/// <returns></returns>
public static HtmlString PhoneNumberLink(this HtmlHelper helper, string
phonenummer, string title,
    bool h1 = false, string addClass = "") {
    var phone = "+" + phonenummer;

    var markup = string.Format("<a href='tel:{0}' class='{1}'
title='{2}'>{3}</a>", phone, addClass, title,
    phone);

    if (h1) {
        markup = string.Format("<h1 title='{0}'>{1}</h1>", title, markup);
    }
    return new HtmlString(markup);
}

public static HtmlString EmailLink(this HtmlHelper helper, string email, string
title, bool h1 = false,
    string addClass = "") {

```

```

        var markup = string.Format("<a href='mailto:{0}' class='{1}' title='{2}'>{3}</a>", email, addClass, title, email);

        if (h1) {
            markup = string.Format("<h1 title='{0}'><strong title='" + title + "'>{1}</strong></h1>", title, markup);
        }
        return new HtmlString(markup);
    }

    #endregion

    #region Generate Publisher, Ideas, Tags

    #endregion

    #region Image Generates

    /// <summary>
    /// </summary>
    /// <param name="helper"></param>
    /// <param name="additionalRootPath"></param>
    /// <param name="file"></param>
    /// <param name="isPrivate"></param>
    /// <param name="asTemp"></param>
    /// <param name="tempString"></param>
    /// <param name="rootPath"></param>
    /// <returns></returns>
    public static string UploadedImageSrc(this HtmlHelper helper, string
additionalRootPath, IUploadableFile file,
    bool isPrivate = false, bool asTemp = false, string tempString = "_temp",
    string rootPath = "~/Uploads/Images/") {
        if (isPrivate) {
            rootPath += "Private/";
        }
        rootPath += additionalRootPath;
        if (!asTemp) {
            tempString = "";
        }
        var fileName = file.FileUploadId + "-" + file.Sequence + tempString + "." +
file.Extension;

        var path = string.Format("{0}{1}", rootPath, fileName);
        return AppVar.Url + VirtualPathUtility.ToAbsolute(path);
        //return (markup);
    }

    public static string GetOrganizeName(IUploadableFile file, bool
includeExtention = false, bool asTemp = false,
    string tempString = "_temp") {
        var ext = "";
        if (!asTemp) {
            tempString = "";
        }
        if (includeExtention) {
            ext = "." + file.Extension;
        }
        return file.FileUploadId + "-" + file.Sequence + tempString + ext;
    }

    /// <summary>
    /// </summary>
    /// <param name="helper"></param>
    /// <param name="src">use absolute http url for image src.</param>
    /// <param name="alt"></param>
    /// <returns></returns>
    public static HtmlString ImageFromAbsolutePath(this HtmlHelper helper, string
src, string alt) {
        var markup = string.Format("<img src='{0}' alt='{1}'/>", src, alt);
        return new HtmlString(markup);
        //return (markup);
    }
}

```

```

    /// <summary>
    /// </summary>
    /// <param name="helper"></param>
    /// <param name="src">relative url.</param>
    /// <param name="alt"></param>
    /// <returns></returns>
    public static HtmlString Image(this HtmlHelper helper, string src, string alt)
    {
        var markup = string.Format("<img src='{0}' alt='{1}'/>",
VirtualPathUtility.ToAbsolute(src), alt);
        return new HtmlString(markup);
        //return (markup);
    }

    /// <summary>
    /// </summary>
    /// <param name="helper"></param>
    /// <param name="folder"></param>
    /// <param name="src">relative from folder</param>
    /// <param name="ext"></param>
    /// <param name="alt"></param>
    /// <returns></returns>
    public static HtmlString Image(this HtmlHelper helper, string folder, string
src, string ext, string alt) {
        var markup = string.Format("<img src='{0}{1}.{2}' alt='{3}'/>",
VirtualPathUtility.ToAbsolute(folder), src,
        ext, alt);
        //return new HtmlString(markup);
        return new HtmlString(markup);
    }

    #endregion

    #region Date and Time Display

    private static string GetDefaultTimeZoneFormat(DateTimeFormatType type =
DateTimeFormatType.Date, string customFormat = null) {
        string format;
        if (!string.IsNullOrEmpty(customFormat)) {
            return customFormat;
        }
        switch (type) {
            case DateTimeFormatType.Date:
                format = "dd-MMM-yyyy";
                break;
            case DateTimeFormatType.DateTimeSimple:
                format = "dd-MMM-yyyy hh:mm:ss tt";
                break;
            case DateTimeFormatType.DateTimeFull:
                format = "MMMM dd, yyyy hh:mm:ss tt";
                break;
            case DateTimeFormatType.DateTimeShort:
                format = "d-MMM-yy hh:mm:ss tt";
                break;
            case DateTimeFormatType.Time:
                format = "hh:mm:ss tt";
                break;
            default:
                format = "dd-MMM-yyyy";
                break;
        }

        return format;
    }

    /// <summary>
    /// Returns a date-time using time-zone
    /// </summary>
    /// <param name="helper"></param>
    /// <param name="timeZone">Timezone set</param>
    /// <param name="dt"></param>

```

```

    /// <param name="formatType">
    /// switch (type) {
    ///     case DateTimeFormatType.Date:
    ///         format = "dd-MMM-yyyy";
    ///         break;
    ///     case DateTimeFormatType.DateTimeSimple:
    ///         format = "dd-MMM-yyyy hh:mm:ss tt";
    ///         break;
    ///     case DateTimeFormatType.DateTimeFull:
    ///         format = "MMMM dd, yyyy hh:mm:ss tt";
    ///         break;
    ///     case DateTimeFormatType.DateTimeShort:
    ///         format = "d-MMM-yy hh:mm:ss tt";
    ///         break;
    ///     case DateTimeFormatType.Time:
    ///         format = "hh:mm:ss tt";
    ///         break;
    ///     default:
    ///         format = "dd-MMM-yyyy";
    ///         break;
    /// }
    /// </param>
    /// <param name="customFormat">If anything passed then this format will be
used.</param>
    /// <param name="addTimeZoneString">Add timezone string with Date. Eg. 26-Aug-
2015 (GMT -07:00)</param>
    /// <returns>Returns a data-time using given format and timezone</returns>
    public static string DisplayDateTime(
        this HtmlHelper helper,
        TimeZoneSet timeZone,
        DateTime? dt = null,
        DateTimeFormatType formatType = DateTimeFormatType.DateTimeCustom,
        string customFormat = null,
        bool addTimeZoneString = false) {
        var format = GetDefaultTimeZoneFormat(formatType, customFormat);
        return Zone.GetDateTime(timeZone, dt, format, addTimeZoneString);
    }

    /// <summary>
    /// Returns a date-time using time-zone
    /// </summary>
    /// <param name="helper"></param>
    /// <param name="userId">User id</param>
    /// <param name="dt"></param>
    /// <param name="formatType">
    /// switch (type) {
    ///     case DateTimeFormatType.Date:
    ///         format = "dd-MMM-yyyy";
    ///         break;
    ///     case DateTimeFormatType.DateTimeSimple:
    ///         format = "dd-MMM-yyyy hh:mm:ss tt";
    ///         break;
    ///     case DateTimeFormatType.DateTimeFull:
    ///         format = "MMMM dd, yyyy hh:mm:ss tt";
    ///         break;
    ///     case DateTimeFormatType.DateTimeShort:
    ///         format = "d-MMM-yy hh:mm:ss tt";
    ///         break;
    ///     case DateTimeFormatType.Time:
    ///         format = "hh:mm:ss tt";
    ///         break;
    ///     default:
    ///         format = "dd-MMM-yyyy";
    ///         break;
    /// }
    /// </param>
    /// <param name="customFormat">If anything passed then this format will be
used.</param>
    /// <param name="addTimeZoneString">Add timezone string with Date. Eg. 26-Aug-
2015 (GMT -07:00)</param>
    /// <returns>Returns a data-time using given format and timezone</returns>
    public static string DisplayDateTime(
        this HtmlHelper helper,

```

```

        long userId,
        DateTime? dt = null,
        DateTimeFormatType formatType = DateTimeFormatType.DateTimeCustom,
        string customFormat = null,
        bool addTimeZoneString = false) {
    var format = GetDefaultTimeZoneFormat(formatType, customFormat);
    return Zone.GetDateTime(userId, dt, format, addTimeZoneString);
}

/// <summary>
/// Returns a date-time using time-zone
/// </summary>
/// <param name="helper"></param>
/// <param name="timeZoneId">TimezoneId from UserTimezone id.</param>
/// <param name="dt"></param>
/// <param name="formatType">
/// switch (type) {
///     case DateTimeFormatType.Date:
///         format = "dd-MMM-yyyy";
///         break;
///     case DateTimeFormatType.DateTimeSimple:
///         format = "dd-MMM-yyyy hh:mm:ss tt";
///         break;
///     case DateTimeFormatType.DateTimeFull:
///         format = "MMMM dd, yyyy hh:mm:ss tt";
///         break;
///     case DateTimeFormatType.DateTimeShort:
///         format = "d-MMM-yy hh:mm:ss tt";
///         break;
///     case DateTimeFormatType.Time:
///         format = "hh:mm:ss tt";
///         break;
///     default:
///         format = "dd-MMM-yyyy";
///         break;
/// }
/// </param>
/// <param name="customFormat">If anything passed then this format will be
used.</param>
/// <param name="addTimeZoneString">Add timezone string with Date. Eg. 26-Aug-
2015 (GMT -07:00)</param>
/// <returns>Returns a data-time using given format and timezone</returns>
public static string DisplayDateTime(
    this HtmlHelper helper,
    DateTime? dt = null,
    DateTimeFormatType formatType = DateTimeFormatType.DateTimeCustom,
    string customFormat = null,
    bool addTimeZoneString = false) {
    var timeZoneSet = Zone.Get();
    var format = GetDefaultTimeZoneFormat(formatType, customFormat);
    return Zone.GetDateTime(timeZoneSet, dt, format, addTimeZoneString);
}

#endregion
}

```

FIGURE 16.2: HELPER METHODS EXAMPLES

C. JavaScript plugins for all components

```

/// <reference path="../jQuery/jquery-2.1.3.js" />
/// <reference path="../jQuery/jquery-2.1.3.intellisense.js" />
/// <reference path="../jQuery/jquery.number.js" />
/// <reference path="../jQuery/jquery.unobtrusive-ajax.js" />
/// <reference path="../jQuery/jquery.validate-vsdoc.js" />
/// <reference path="../jQuery/jquery.validate.js" />
/// <reference path="../jQuery/jquery.validate.unobtrusive.js" />

```

```

/// <reference path="../../jQuery/moment.js" />
/// <reference path="../../jQuery/underscore.js" />
/// <reference path="../../jQuery/validation.js" />
/// <reference path="../../jQuery/Upload/jquery.fileupload.js" />
/// <reference path="../../Bootstrap/bootstrap.js" />
/// <reference path="../../Bootstrap/bootstrap-select.js" />
/// <reference path="../../Bootstrap/bootstrap-table-export.js" />
/// <reference path="../../Bootstrap/bootstrap-table-filter.js" />
/// <reference path="../../Bootstrap/bootstrap-datetimepicker.js" />
/// <reference path="../../Bootstrap/bootstrap-datepicker.js" />
/// <reference path="../../Bootstrap/common-tasks-run-every-page.js" />
/// <reference path="../../Bootstrap/modernizr-2.8.3.js" />
/// <reference path="../../Bootstrap/respond.js" />
/// <reference path="../../Bootstrap/star-rating.js" />

/*
 * Version 2.2
 * Written by Alim Ul Karim
 * mailto:alim.karim.nsu@gmail.com
 */

$.fn.extend({
    // jQuery element get all classes
    getAllClasses: function () {
        if (this.length === 1) {
            return this.attr("class").split(/\s+/);
        }
        return null;
    }
});

$.devOrg = {

    // get all the classes from an jQuery element
    getAllClasses: function ($jQueryHtmlElement) {
        "use strict";
        return $jQueryHtmlElement.getAllClasses();
    },

    // allClassesArray = ['a','b','c'] , exceptClassesArray=['b','c'], results=['a']
    getClassesExcept: function (allClassesArray, exceptClassesArray) {
        "use strict";
        if (allClassesArray === null || allClassesArray === undefined) {
            return [];
        }

        if (exceptClassesArray === null || exceptClassesArray === undefined) {
            return allClassesArray;
        }
        var len = allClassesArray.length;
        var results = [];
        for (var i = 0; i < len; i++) {
            if (exceptClassesArray.indexOf(allClassesArray[i]) === -1) {
                // not found
                results.push(allClassesArray[i]);
            }
        }
        return results;
    },
    // all Selectors are jQuery Selector Text only.
    // selectpicker will be called inside function, no need to call outside.
    countryFlagRefresh: function (countrySelector, dropdownItemsSelector,
        dropdownBtnSelector) {
        "use strict";
        var countryBox = $(countrySelector).selectpicker(); // only select a select
        element then apply the custom bootstrap selector
        var dropdownItems = $(dropdownItemsSelector); // getting generated dropdown
        items from the custom bootstrap selector
        var dropdownBtn = $(dropdownBtnSelector); // generated new button from the
        selectpicker option
    }
};

```

```

var skippingClassesAnchor = ["flag-country-combo", "flag"];
var skippingClassesForBtn = ["btn", "dropdown-toggle", "selectpicker", "btn-
success", "flag-combo"];

// console.log(dropDownItems.length);
countryBox.change(function (e) {
    var listItem = dropDownItems.find("li.selected");
    var anchorItem = listItem.find("a");
    var listOfAllAnchorClasses = anchorItem.getAllClasses();
    var listOfAllClassesdropDownBtn = dropDownBtn.getAllClasses();
    var flagClass = $.devOrg.getClassesExcept(listOfAllAnchorClasses,
skippingClassesAnchor);
    var btnFlagClass = $.devOrg.getClassesExcept(listOfAllClassesdropDownBtn,
skippingClassesForBtn);
    for (var i = 0; i < btnFlagClass.length; i++) {
        dropDownBtn.removeClass(btnFlagClass[i]);
    }
    dropDownBtn.addClass("fc-" + flagClass[0]);
});
},
// countryFlagRefresh must be called first or selectpicker must be called first
// all Selectors are jQuery Selector Text only.
countryRelatedToPhone: function (countrySelector, dropDownItemsSelector,
dropDownBtnSelector, phoneNumberInputSelector) {
    "use strict";
    var countryBox = $(countrySelector);
    var dropDownItems = $(dropDownItemsSelector);
    //var dropDownBtn = $(dropDownBtnSelector);
    var phoneNumberBox = $(phoneNumberInputSelector);
    var previousCallingCode = "";

    function selectChangeState() {
        // console.log("executed");
        var listItem = dropDownItems.find("li.selected");
        var spanText = listItem.find("a > span").text().toString();
        var newCallingCode = $.devOrg.subStringMod(spanText, "(", ")");
        var getWrittenPhoneNumber = phoneNumberBox.val();
        // console.log(listItem);
        newCallingCode = $.devOrg.replaceStartsWith(newCallingCode, "+", "");
        if ((!_.isEmpty(getWrittenPhoneNumber) && !_.isEmpty(previousCallingCode))
        && $.devOrg.isStartsWith(getWrittenPhoneNumber, previousCallingCode)) {
            getWrittenPhoneNumber =
$.devOrg.replaceStartsWith(getWrittenPhoneNumber, previousCallingCode, newCallingCode);
        } else {
            getWrittenPhoneNumber = newCallingCode + getWrittenPhoneNumber;
        }
        previousCallingCode = newCallingCode;
        phoneNumberBox.val(getWrittenPhoneNumber);
    }

    countryBox.ready(selectChangeState).change(selectChangeState);
    //phoneNumberBox.keyup(selectChangeState);
    // $("#selectID option")[index].selected = true;
},

subStringMod: function (givenString, startSequence, endingSequence) {
    "use strict";
    if (_.isString(givenString)) {
        var index1 = givenString.indexOf(startSequence);
        if (index1 > -1) {
            var index2 = givenString.indexOf(endingSequence);
            if (index2 > -1) {
                // exist
                return givenString.substr(index1 + 1, index2 - index1 - 1);
            }
        }
    }
    return null;
},
// parentjQueryCombo = passjQueryElement , mainDivContainerSelector = ".something-
main", innerDivSelectorForPlacingCombo= ".something-combo-div"
// it would be better to execute parentjQueryCombo as selectpicker or have a
selectpicker class.

```



```

// No combo will appear , even the main div will disappear if no item is received
from the link.
// json sender should sends as id and text only.
smartDependableCombo: function (parentjQuerySelector, mainDivContainerSelector,
innerDivSelectorForPlacingCombo, urlToGetJson, placeComboName, placedComboId,
placedComboClass, placedComboAdditionalClassesWithItems,
placedComboAdditionalHtmlWithEachItem) {
    "use strict";
    var $parentjQueryCombo = $(parentjQuerySelector);
    if (_.isEmpty($parentjQueryCombo)) {
        console.error.log("error raised from developers organism component's
smartDependableCombo that no parent is detected.");
        return; // nothing exist in parent.
    }
    var $mainDiv = $(mainDivContainerSelector);
    var $innerDiv = $mainDiv.find(innerDivSelectorForPlacingCombo);

    function hideDiv() {
        if ($mainDiv.length > 0) {
            $mainDiv.hide();
        } else {
            console.error.log("devOrg->smartDependableCombo: main div not found for
'" + mainDivContainerSelector + "'");
        }
    }

    hideDiv();

    function showDiv() {
        // remove select if exist.
        var options = $innerDiv.find("select, div.bootstrap-select");
        if (options.length > 0) {
            options.remove();
        }
        $mainDiv.show("slow");
    }

    function createCombo(response) {
        if (!_.isEmpty(placedComboId)) {
            placedComboId = " id='" + placedComboId + "' ";
        } else {
            placedComboId = "";
        }
        if (_.isEmpty(placedComboClass)) {
            placedComboClass = "";
        }

        if (_.isEmpty(placeComboName)) {
            placeComboName = "";
        } else {
            placeComboName = " name='" + placeComboName + "' ";
        }

        $innerDiv.prepend("<select " + placeComboName + " class='devOrgSmartCombo
form-control " + placedComboClass + " selectpicker'" + placedComboId + " data-style='" +
placedComboClass + "' data-live-search='true'></select>");
        var $combo = $innerDiv.find("select");
        $.devOrg.appendedComboElement($combo, response,
placedComboAdditionalHtmlWithEachItem, placedComboAdditionalClassesWithItems);
        $combo.selectpicker();
    }

    $parentjQueryCombo.change(function () {
        "use strict";
        var $parentComboValue = $parentjQueryCombo.val();
        var actualUrl = urlToGetJson + "/" + $parentComboValue;
        $.ajax({
            type: "POST",
            dataType: "JSON",
            url: actualUrl,
            success: function (response) {
                if (response.length === 0) {

```

```

        hideDiv();
        return;
    }
    $innerDiv = $(mainDivContainerSelector + " " +
innerDivSelectorForPlacingCombo);
    // items exist.
    showDiv(); //remove inner options if exist any
    createCombo(response); // create if necessary and then append
options to it.
    },
    error: function (xhr, status, error) {
        hideDiv();
    }
    });
});
},
// listOfItems = expected a json item with id and text property
// extraHtmlWithEachElement : represents like below
// <option .. > extraHtmlWithEachElement Item </option>
appenedComboElement: function ($combo, listOfItems, extraHtmlWithEachElement,
itemClasses) {
    /// <summary>
    ///
    /// </summary>
    /// <param name="$combo"></param>
    /// <param name="listOfItems"></param>
    /// <param name="extraHtmlWithEachElement"></param>
    /// <param name="itemClasses"></param>
    "use strict";
    // followed by the best practice : http://
allthingscraig.com/blog/2012/09/28/best-practice-appending-items-to-the-dom-using-jquery/
    if (_.isEmpty(itemClasses)) {
        itemClasses = "";
    }
    if (_.isEmpty(extraHtmlWithEachElement)) {
        extraHtmlWithEachElement = "";
    }
    if (listOfItems.length > 0) {
        var length = listOfItems.length;
        var options = "";
        var selected = " Selected='selected' ";
        var optionStarting = "<option class='devorgCombo-item " + itemClasses +
"";
        var optionEnding = "</option>";
        for (var i = 0; i < length; i++) {
            if (i === 0) {
                selected = "";
            }
            options += optionStarting + selected + "value='" + listOfItems[i].id +
">" + extraHtmlWithEachElement + listOfItems[i].text + optionEnding;
        }
        $combo.append(options);
    }
},
bootstrapComboSelectbyFindingValue: function (comboSelector, searchForvalue) {
    "use strict";
    $(comboSelector).selectpicker("val", searchForvalue).trigger("change");
},
bootstrapComboSelectIndex: function (comboSelector, index) {
    "use strict";
    var combo = $(comboSelector + ">option");
    if (combo.length > 0 && index <= (combo.length - 1)) {
        var itemFound = $(combo[index]);
        var value = itemFound.val();
        $.devOrg.bootstrapComboSelectbyFindingValue(comboSelector, value);
    }
},
// givenString "Example ( Hello )"
// startsWith= "Example" ; returns true.
isStartsWith: function (givenString, startsWith) {
    "use strict";

```

```

        if (_.isString(givenString)) {
            var substringOfGiventext = givenString.substr(0, startsWith.length);
            if (substringOfGiventext === startsWith) {
                return true;
            }
        }
        return false;
    },

    replaceStartsWith: function (givenString, findStartsWith, replaceString) {
        "use strict";
        if (_.isString(givenString) && !_.isEmpty(findStartsWith)) {
            var substringOfGiventext = givenString.substr(0, findStartsWith.length);
            if (substringOfGiventext === findStartsWith) {
                var nextStringIndex = findStartsWith.length;
                var otherHalfText = givenString.substr(nextStringIndex,
givenString.length - nextStringIndex);
                return replaceString + otherHalfText;
            }
        }
        return givenString;
    },

    // jquery formSelector, submitAtLast:true/false
    enterToNextTextBox: function (formSelector, submitAtLast) {
        "use strict";
        $(formSelector + " input:text:first").focus();
        var binders = formSelector + " input[type='text']:visible," +
            formSelector + " input[type='password']:visible," +
            formSelector + " input[type='numeric']:visible," +
            formSelector + " input[type='email']:visible," +
            //formSelector + " textarea:visible," +
            formSelector + " button.selectpicker[type='button']:visible," +
            formSelector + " select:visible";
        $(document).on("keypress", binders, function (e) {
            // var codeAbove = d.keyCode || d.which;
            // console.log("above code :" + codeAbove);
            var code = e.keyCode || e.which;
            // console.log("inside code :" + code);
            if (code === 13) { // Enter key
                e.preventDefault(); // Skip default behavior of the enter key
                var n = $(binders).length;
                var nextIndex = $(binders).index(this) + 1;
                if (nextIndex < n) {
                    $(binders)[nextIndex].focus();
                } else {
                    $(binders)[nextIndex - 1].blur();
                    if (submitAtLast === true) {
                        $(formSelector).submit();
                    }
                }
            }
        });
    },

    validateTextInputBasedOnRegex: function (jQuerySelectorforTextBox, stringRegex,
msgOnInvalidPattern) {
        "use strict";
        /// <summary>
        ///     Validate text input while typing with ASP.NET jquery validation.
        ///     Only the attributes with the text. No event is bound.
        /// </summary>
        /// <param name="jQuerySelectorforTextBox">string: jQuery Selector</param>
        /// <param name="stringRegex">string: Regular expression to validate the
textInput</param>
        $(jQuerySelectorforTextBox).attr("data-val-regex-pattern", stringRegex);
        if (!_.isEmpty(msgOnInvalidPattern)) {
            $(jQuerySelectorforTextBox).attr("data-val-regex", msgOnInvalidPattern);
        }
    },

    reSetupjQueryValidate: function (jQueryFormSelector) {

```

```

        "use strict";
        /// <summary>
        ///     call after setting new reg ex via validateTextInputBasedOnRegEx
        /// </summary>
        /// <param name="jQueryFormSelector"></param>
        var $form = $(jQueryFormSelector)
            .removeData("validator") /* added by the raw jquery.validate plugin */
            .removeData("unobtrusiveValidation");
        /* added by the jquery unobtrusive plugin */
        $.validator.unobtrusive.parse($form);
    },

    validateTextInputsBasedOnHiddenSpansGiven: function (formSelector) {
        "use strict";
        /// <summary>
        ///     inComplete
        ///     There should be span for each of inputs that needs to be modified or
changed.
        ///     hidden span with data-name="same as input"
        ///     data-display, data-reg, data-reg-fail-msg, data-min,data-max,data-
range-failed-msg
        ///     data-placeholder, data-class
        /// </summary>
        /// <param name="formSelector">jQuery form selector</param>
        var $form = $(formSelector);
        var dataRegSpanAttr = "data-reg";
        var dataRegMsgSpanAttr = "data-reg-fail-msg";
        var dataMinSpanAttr = "data-min";
        var dataMaxSpanAttr = "data-max";
        var dataRangeMsgSpanAttr = "data-min";
        var dataDisplaySpanAttr = "data-range-failed-msg";
        var dataMinSpanAttr = "data-min";

        if ($form.length > 0) {
            var binders = "input[type='text'], " +
                "input[type='password'], " +
                "input[type='email'], " +
                "input[type='numeric'], " +
                "select:visible";
            var $inputs = $form.find(binders);
            for (var i = 0; i < $inputs.length; i++) {
                var $input = $($inputs[i]);
                var $nameOftheInput = $input.attr("name");
                // now find the span based on the name
                var $span = $form.find("span[data-name='" + $nameOftheInput +
                "']:hidden");

                if ($span.length > 0) {
                    // set the attr to the input.

                    // setting reg ex
                }
            }
        }
    },

    validateInputFromServer: function (jQuerytextBoxSelector, validationUrl,
internalValidatorSpanClassName, isAlwaysFocusUntilValid, isDisable, minChars,
isSubmitTheWholeForm, onInvalidStringStatementInCrossMark,
onValidStringStatementInCheckMark, $formGiven, maxTryLimit) {
        /// <summary>
        ///     Made validation easy on the fly with a server response.
        /// </summary>
        /// <param name="jQuerytextBoxSelector">string: jQuery Selector</param>
        /// <param name="validationURL">string: Url to validate</param>
        /// <param name="internalValidatorSpanClassName">Propertyname (refer to the
class of validation span) Indicating class.</param>
        /// <param name="isAlwaysFocusUntilValid">Boolean: Keep the focus until it's
valid. By default: false</param>
        /// <param name="isDisable">Boolean: Disable the textbox after being validated.
By default : false.</param>
        /// <param name="minChars">number: min chars to send the request</param>

```

```

    /// <param name="isSubmitTheWholeForm">Boolean:Rather than submitting the small
form submit the whole related closet form.</param>
    /// <param name="onInvalidStringStatementInCrossMark">invalid statement show on
the cross mark.</param>
    /// <param name="onValidStringStatementInCheckMark">
    ///     valid statement show on the check mark. By default: fieldDisplayname +
    ///     is available and valid
    /// </param>

    "use strict";

    //if (_.isEmpty(isSubmitTheWholeForm)) {
    //    isSubmitTheWholeForm = false;
    //}

    //if (_.isEmpty(isAlwaysFocusUntilValid)) {
    //    isAlwaysFocusUntilValid = false;
    //}

    //if (_.isEmpty(isDisable)) {
    //    isDisable = false;
    //}

    var sentRequestCount = 0;
    var $userTextbox = $(jQuerytextBoxSelector);
    if ($userTextbox.length > 0) {
        $userTextbox.removeAttr("isDisable");

        if (!isSubmitTheWholeForm) {
            $userTextbox.keyup(function () {
                $("#validation #id").val($userTextbox.val());
                // console.log(user);
            }).focus(function () {
                $("#validation #id").val($userTextbox.val());
                // console.log(user);
            });
        }

        $userTextbox.blur(function () {
            "use strict";
            if (!isSubmitTheWholeForm) {
                $("#validation #id").val($userTextbox.val());
            }
            var $passingText = $userTextbox.val();
            if (_.isEmpty($passingText) || $passingText.length < minChars) {
                // if empty text then don't send.
                return;
            }
            if (_.isEmpty(onValidStringStatementInCheckMark)) {
                onValidStringStatementInCheckMark = "is available and valid.";
            }
            if (_.isEmpty(onInvalidStringStatementInCrossMark)) {
                onInvalidStringStatementInCrossMark = "is not valid or already
exist. Your input can't contain ( [ ] ' , * & ? \ " ) or space or any other special
character for this data-type.";
            }

            // Validation should be a formData underlying the original form with
            // only antiforgery token and a hidden id field
            // whatever is typed in that selected text box will be pushed into
            // this formData
            var formData;
            if ($formGiven === null || $formGiven === undefined ||
$formGiven.length == 0) {
                if (!isSubmitTheWholeForm) {
                    formData = $("#validation").serialize();
                } else {
                    formData = $userTextbox.closest("form").serializeArray();
                }
            } else {
                formData = $formGiven.serializeArray();
            }
        }
    }

```

```

        //console.log(formData);

        var validatorName = "span.CustomValidation." +
internalValidatorSpanClassName;
        var token = $("input[name=__RequestVerificationToken]").val();
        var processingState1 = "glyphicon-refresh";
        var processingState2 = "glyphicon-refresh-animate";
        var isHideClass = "hide";
        var colorGreen = "green";
        var colorRed = "red";
        var correctState = "glyphicon-ok";
        var incorrectState = "glyphicon-remove";
        var $validatorBox = $(validatorName);
        var displayName = $validatorBox.attr("data-display");
        var correctStateTitle = displayName + " " +
onValidStringStatementInCheckMark;
        var invalidAttrName = "data-invalid";
        var incorrectStateTitle = displayName + " " +
onInvalidStringStatementInCrossMark;
        var tooltipName = "a.CustomValidation." +
internalValidatorSpanClassName + ".tooltip-show";
        var $tooltipBox = $(tooltipName);

        // console.log($("#validation #id").val());
        $validatorBox.removeClass(incorrectState).removeClass(correctState);

        // if no processing state then add it
        if (!$validatorBox.hasClass(processingState1)) {
            $validatorBox.addClass(processingState1);
        }

        if (!$validatorBox.hasClass(processingState2)) {
            $validatorBox.addClass(processingState2);
        }
        if ($validatorBox.hasClass(isHideClass)) {
            $validatorBox.removeClass(isHideClass);
        }
        $tooltipBox.attr("data-original-title", "Validating " + displayName)
            .attr("title", "Validating " + displayName);
        // confirming processing state.
        if (maxTryLimit !== null && maxTryLimit !== undefined &&
sentRequestCount > maxTryLimit) {
            return;
        }
        $.ajax({
            type: "POST",
            dataType: "JSON",
            url: validationUrl,
            data: formData,
            success: function (response) {
                sentRequestCount = sentRequestCount + 1;
                // Remove the processing state
                if ($validatorBox.hasClass(processingState1)) {
                    $validatorBox.removeClass(processingState1);
                }

                if ($validatorBox.hasClass(processingState2)) {
                    $validatorBox.removeClass(processingState2);
                }
                if ($validatorBox.hasClass(isHideClass)) {
                    $validatorBox.removeClass(isHideClass);
                }
                // Remove the processing state
                if (response == true) {
                    if ($validatorBox.hasClass(incorrectState)) {
                        $validatorBox.removeClass(incorrectState);
                    }
                    if ($validatorBox.hasClass(colorRed)) {
                        $validatorBox.removeClass(colorRed);
                    }
                    $validatorBox.addClass(colorGreen)
                        .addClass(correctState)

```

```

        .attr("title", correctStateTitle);

        $tooltipBox.attr("title", correctStateTitle)
            .attr("data-original-title", correctStateTitle);
        if (isDisable) {
            $userTextbox.prop("isDisable", true);
        }

        $userTextbox.addClass("bold")
            .addClass("green")
            .next()
            .focus();

        $userTextbox.removeAttr(invalidAttrName);
    } else {
        if ($validatorBox.hasClass(colorGreen)) {
            $validatorBox.removeClass(colorGreen);
        }
        if ($validatorBox.hasClass(correctState)) {
            $validatorBox.removeClass(correctState);
        }
        $userTextbox.prop("isDisable", false)
            .addClass("bold")
            .addClass("red");

        $validatorBox.addClass(colorRed)
            .addClass(incorrectState)
            .attr("title", incorrectStateTitle);

        $tooltipBox.attr("title", incorrectStateTitle)
            .attr("data-original-title", incorrectStateTitle);
        if (isAlwaysFocusUntilValid === true) {
            $userTextbox.focus();
        }
        $userTextbox.attr(invalidAttrName, "true");
    }
    $(".tooltip-show").tooltip();
},
error: function (xhr, status, error) {
    // Remove the processing state
    if ($validatorBox.hasClass(processingState1)) {
        $validatorBox.removeClass(processingState1);
    }

    if ($validatorBox.hasClass(processingState2)) {
        $validatorBox.removeClass(processingState2);
    }
    if ($validatorBox.hasClass(isHideClass)) {
        $validatorBox.removeClass(isHideClass);
    }
    // Remove the processing state
    if ($validatorBox.hasClass(correctState)) {
        $validatorBox.removeClass(correctState);
    }

    if ($validatorBox.hasClass(colorGreen)) {
        $validatorBox.removeClass(colorGreen);
    }
    $userTextbox.prop("isDisable", false)
        .addClass("bold")
        .addClass("red");

    $validatorBox.addClass(colorRed)
        .addClass(incorrectState)
        .attr("title", error);

    $tooltipBox.attr("title", status)
        .attr("data-original-title", error);

    $(".tooltip-show").tooltip();
    $userTextbox.attr(invalidAttrName, "true");
}
}

```

```

    }); // ajax end
  });
}; // if else end
},

fillRegisterFieldsOnDemo: function () {
  /// <summary>
  /// Test Function
  /// </summary>
  var i = 0;
  var controls = $(".form-group");
  var $fields = controls.find("input[type=text]");
  $.each($fields, function () {
    this.value = 111111111111;
  });

  $fields = controls.find("input[type=password]");
  $.each($fields, function () {
    this.value = "asdf1234@";
  });

  $fields = controls.find("input[type=number]");
  $.each($fields, function () {
    this.value = i++;
  });

  $fields = controls.find("textarea");
  $.each($fields, function () {
    this.value = "111111111111";
  });

  $fields = controls.find("input[type=email]");
  $.each($fields, function () {
    this.value = "auk.junk@live.com";
  });

  $fields = controls.find("input[type=checkbox]");
  $.each($fields, function () {
    this.prop("checked", true);
  });
},
//'.make-it-tab'
bootstrapTabsMordernize: function (tabSelector) {
  /// <summary>
  /// give jQuery selector to add tab functionality
  /// </summary>
  /// <param name="tabSelector"></param>
  "use strict";
  var bootstrapTabs = $(tabSelector);
  if (bootstrapTabs.length > 0) {
    var tabHidden = $(".tab-content input[type='hidden'][name='tab']");

    if (tabHidden.length > 0) {
      var tabHiddenValue = tabHidden.val();
      if (!_.isEmpty(tabHiddenValue)) {
        //tab name exist
        bootstrapTabs.find("li>a[href='" + tabHiddenValue +
        ""']").tab("show");
      } else {
        //no tab name exist.. select default one.
        bootstrapTabs.find("li > a:first").tab("show");
      }
    }

    bootstrapTabs.click(function (e) {
      //e.preventDefault();
      e.preventDefault();
      $(this).tab("show");
    });
  }
};

```



```

(e) {
    $("ul" + tabSelector + ".nav-tabs > li > a").on("shown.bs.tab", function
        var valueOfActive = $(e.target).attr("href");
        // = $(tabSelector + " li.active>a").attr('href');
        tabHidden.val(valueOfActive);
        //window.location.hash = id;
    });
}
},
ratingMordernize: function () {

    var ratingItems = $(".rating-5");

    if (ratingItems.length > 0) {
        ratingItems.rating({
            showClear: false
        });
    }
    ratingItems = $(".rating-10");

    if (ratingItems.length > 0) {
        ratingItems.rating({
            showClear: false,
            starCaptionClasses: {
                0.5: "label label-danger",
                1: "label label-danger",
                1.5: "label label-danger",
                2: "label label-danger",
                2.5: "label label-danger",
                3: "label label-warning",
                3.5: "label label-warning",
                4: "label label-warning",
                4.5: "label label-warning",
                5: "label label-warning",
                5.5: "label label-info",
                6: "label label-info",
                6.5: "label label-info",
                7: "label label-info",
                7.5: "label label-primary",
                8: "label label-primary",
                8.5: "label label-success",
                9: "label label-success",
                9.5: "label label-success",
                10: "label label-success"
            }
        });
    }
},

uxFriendlySlide: function (jQueryformSelector, keepOthersVisible, dontSubmit) {
    /// <summary>
    ///     hides except for the first div with value 0. Add attributes to divs
    ///     [data-dev-slide='number-zero-based'][data-dev-visited='false'] and
    ///     encapsulate inputs. Each click clicked on submit it will verify the
    inputs if verified next hide ones will be shown
    ///     it will be continuous process until hit the last.
    ///     Always use lower case false
    ///     [data-dev-slide='number-zero-based'][data-dev-visited='false']
    /// </summary>
    /// <param name="jQueryformSelector">jQuery selector for the form</param>
    /// <param name="keepOthersVisible">Should add new hide ones or previous ones
    hides and load new ones(divs)</param>
    /// <param name="dontSubmit">When none left , do we submit? True: don't
    submit</param>
    "use strict";

    var slideObjects = $(jQueryformSelector + " [data-dev-slide][data-dev-
    visited='false']");
    var executedOnce = false;
    var binders = "input[type='text']:visible," +

```

```

        "input[type='password']:visible," +
        "input[type='email']:visible," +
        "input[type='numeric']:visible," +
        "select:visible";
    var order = 0;
    var totalSliderLength = slideObjects.length;
    var previousSlideNumber = 0;
    if (totalSliderLength > 0) {
        // exist slides.
        slideObjects.hide();
        previousSlideNumber = order;
        slideObjects.filter("[data-dev-slide='" + (order++) + "'][data-dev-
visited='false']").show();
        $(jQueryformSelector).submit(function (e) {
            e.preventDefault();

            var nextOne = slideObjects.filter("[data-dev-slide='" + order +
            "'][data-dev-visited='false']");
            // if (nextOne.length == 0) {
            //     for (order += 1; nextOne.length == 0 && totalSliderLength >=
order; order++) {
                //         nextOne = slideObjects.filter("[data-dev-slide='" + order +
                "'][data-dev-visited='false']");
                //     }
                // }
                var previousOne;
                var inputBoxes;
                if (nextOne.length > 0) {
                    previousOne = slideObjects.filter("[data-dev-slide='" + (order - 1)
+ "']"); // console.log(previousOne);
                    inputBoxes = previousOne.find("input, textarea"); // still exist ,
prevent submission
                    if (inputBoxes.length > 0 && $.devOrg.checkValidInputs(inputBoxes))
                    {
                        if (!keepOthersVisible) {
                            previousOne.hide("slow");
                        }
                        //console.log(inputBoxes);
                        //console.log(binders);
                        if (!nextOne.prop("data-dev-visited")) {
                            nextOne.attr("data-dev-visited", "true");
                            nextOne.show("slow");
                            //console.log(nextOne);
                            order++;
                        }
                    } else {
                        //console.log("no inboxes");
                    }
                } else {
                    // nothing left.
                    // still check the validation.
                    previousOne = slideObjects.filter("[data-dev-slide='" + (order - 1)
+ "']"); // console.log(previousOne);
                    inputBoxes = previousOne.find("input");
                    if (inputBoxes.length > 0 && $.devOrg.checkValidInputs(inputBoxes))
                    {
                        if (!dontSubmit) {
                            this.submit();
                        }
                    }
                }
            }

        });

        // var notVisited = slideObjects.filter("[data-dev-visited='false']");
    },
    // Send inputs array, if any of those false , returns false.
    checkValidInputs: function (jBinders) {
        "use strict";

```

```

var $currentInput = null;
var length = jBinders.length;
var label = "<label class='label label-danger small-font-size'>Please rate
first.</label>";
if (length > 0) {
    for (var i = 0; i < length; i++) {
        $currentInput = $(jBinders[i]);

        if ($currentInput.hasClass("common-rating")) {
            var $ratingContainer = $currentInput.closest("div.rating-
container");
            var $wholeContainer = $ratingContainer.closest("div.star-rating");

            if ($currentInput.val() === "0") {
                $ratingContainer.css({
                    'text-shadow': "2px 2px red"
                });
                if (!$wholeContainer.attr("data-warned")) {
                    $wholeContainer.append(label);
                    $wholeContainer.attr("data-warned", "true");
                }
                return false;
            } else {
                $ratingContainer.css({
                    'text-shadow': "none"
                });

                if ($wholeContainer.attr("data-warned")) {
                    $wholeContainer.find("label").remove();
                    $wholeContainer.attr("data-warned", "false");
                }
            }
        }
        if (!$currentInput.valid()) {
            return false;
        }
    }
}
return true;
}
};

```

FIGURE 16.3: JAVASCRIPT COMPONENTS CODES