Adaptation of .NET MVC Framework in Developing an Agriculture Sources Inventory System

Ray Adderley JM. Gining [1*], S.S.M. Fauzi [2], Muhammad Nabil Fikri Jamaluddin [3], Nur Syamimi Aima Bt Mohd Azmi[4], Muhamad Arif Hashim[5], Mohd Faris Mohd Fuzi[6], Alif Faisal Ibrahim[7]

[1] *Faculty of Computer Science and Mathematics, Universiti Teknologi MARA, Cawangan Perlis, Kampus Arau, Malaysia*

*Corresponding author: *ray_adderley@perlis.uitm.edu.my*

Received Date: 29 August 2018
Accepted Date: 18 October 2018

ABSTRACT

A viable inventory management system is crucial to an organization, especially those who deals with stock going in and out of their premise. The system should cover several aspects in managing the collection of the available stocks to ensure efficiency. An inventory management system comprises crucial functions that the goal is to produce accurate information – a minor mis-information could lead to a wrong decision making. Improper system design could lead to the problem; thus, the design of this inventory system is supported by .NET Model View Controller (MVC) Framework as its backbone. The .NET MVC Framework is one of the most widely used framework that is written in C# programming language. By utilizing the MVC Framework, its three main components; data model (Model), user interface (View), and business logic (Controller) can be managed separately without affecting the entire system. Thus, the risk of the system functions to produce inaccurate information is reduced significantly. Moreover, MVC also supports scalability, extend-ability and maintainability which is a very important aspect for an information system with growing needs of new functions.

Keywords: *inventory management system, .NET MVC Framework, model view controller*

INTRODUCTION

Inventory referred to a business process which mainly focused on stock and goods according to their business requirements (Nyemba & Mbohwa, 2017). In other words, the inventory can be simplified to unused resources, continuous development, extra resources and completed goods (Gitau, 2016). Those kinds of inventories needed to be in proper management as well as to maintain the steadiness of the organization with the current markets nowadays. According to Gitau et al. (2016), the activities related to the inventory must be stored in one place to ensure that the services and deliverable products met the user expectations. Meanwhile, inventory itself helped in reducing out of stock and the cost of inventory items purchased at the right times also will be cut off.

The objective of this paper is to develop an inventory system for agriculture sources and adopting the .NET MVC Framework as the backbone of the system. An inventory system development process has always been a challenge because the adopted technology requires to be addressed properly. An incorrect selection of a technology could lead to the information system to be un-maintainable, inefficient and inaccurate; thus, resulting to a problem of the needs to develop a brand-new system repeatedly – which is costly.

Hence, .NET MVC Framework is introduced in this project to support the system development process which could reduce the risks of facing said problems – specifically in maintaining the function integrity.

This first section of this paper introduces the motivation and objective of this paper. Followed by the second section which is the literature review focused on the discussion of the MVC Framework; its description, benefits, types and the selection of the framework. The third section discusses the system development process, detailing on how the MVC Framework is adopted, and finally, the result and discussion section which discusses the functionality evaluation method used to assess the effectiveness of the adopted MVC Framework.

LITERATURE REVIEW

Model View Controller (MVC) Framework

In the 1970s, Trygve Reenskaug is the first person who intended the MVC design pattern at the Xerox Parc (Pop & Altar, 2014). He considered the MVC as the connector between the human user's mental model and the digital model in the computer.

Basically, the framework itself consist of three components which are Model, View, and Controller. First, the Model described the business logic of a web application that related to the raw facts, validation, control and the knowledge base (Pop & Altar, 2014).

According to the Jess Chadwick and Todd Snyder (2012), the visual presentation of MVC framework has been handled by the View. They also stipulated that the Controller received the request from the view part and accessed the information from the model and passed back the result to the view.

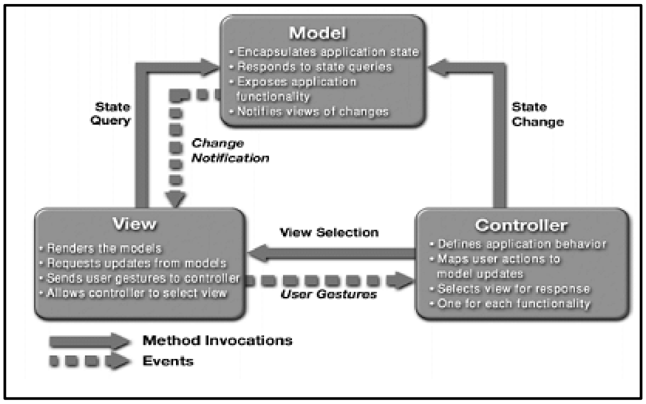

Model
• Encapsulates application state
• Responds to state queries
• Exposes application functionality
• Notifies views of changes
State Query
State Change
Change Notification
View
• Renders the models
• Requests updates from models
• Sends user gestures to controller
• Allows controller to select view
View Selection
User Gestures
Controller
• Defines application behavior
• Maps user actions to model updates
• Selects view for response
• One for each functionality
Method Invocations
Events

Figure 1: Model View Controller (MVC) Framework

(Source: Dhananjay Kumar, 2009)

MVC Benefits

There are several benefits of MVC model for building a web application. One of the benefits of using the MVC model itself is object-relational mapping system (ORM) designed to link with each of the models in the framework and helped to interact with the data especially related to the highest number of linkage between the models created (Pop & Altar, 2014). Focusing on the security, Pop and Altar (2014) show that the (ORM) system was a complex role-based security system assembled with CSRF prevention, where the developers are needed to give full attention against the attackers. According to the Point (2014), MVC platform is very efficient to handle while working with the large team of developers and for the web designer.

The benefit of MVC framework is it had its own specialty which inspired the programmers to develop web-based application suit to their company demands.

MVC Types

There are several frameworks available focusing on the development of web application in the market nowadays. JavaScript language is a cross-platform scripting language which provided the interactivity between HTML pages (Jiang Li et.al., 2010). For instance, AngularJS was one of the JavaScript frameworks. There were several characteristics in AngularJS such as two-way binding which enabled the code to disappear and focused mainly on the application, the browser is parsed into DOM while using HTML templates and others (Jain, Mangal & Mehta, 2014). Additional characteristics which included their preference were elements can be "decorated" and DOM can be manipulated in such an interesting way.

In addition, Symphony2, CakePHP, and Zend Framework are the examples of PHP frameworks which had their own strength in a web-based application. According to the (Prokofyeva & Boltunova, 2017), Symphony2 had a better choice in configurations of XML format and Symphony2 framework can be loaded the number of files around 15.86MB in a single request. Symphony also supported numerous database such as MySQL, SQLite, PostgreSQL etc. Scala is another programming language that had the functional extension of the object-oriented characteristics (Jiang Li et.al., 2010). They also stated that Scala had the features where it can be used all the available resources in Java including the libraries itself.

C# also is a modern and multi-paradigm language which was designed by Andres Hejlsberg (the inventor of Turbo Pascal) (Jiang Li et.al., 2010). In C# MVC framework, it can inherit a class from the parent class and the child class can inherit all the attributes from the parent class.

The selection of the framework used in designing the inventory system is based on the comparison of the programming language features practised in each of the framework. The features comparison is based on security, web application development and graphical user interface.

PHP and JavaScript are lacking in terms of security due to the type-checking during runtime (Jiang Li et.al., 2010). They also stated that C++ programming language considered not safe because C++ itself does not provide garbage collection and memory layout. C# and Java are the best selection because it already provided an exception handling mechanism. Thus, the C# programming language is preferable than the other programming languages.

Many types of programming language supported the web application development. In designing the web, the C# in ASP.NET, PHP and Java are the easiest ones because they supported several features and popular for last past years. On the side of PHP language, it was very convenient for the web development purposes but it less secured compared to others. Apart from that, OO-based abstraction also supported by Java and C# languages where they are strong enough rendering to the syntax and the extension libraries.

PHP also deliver the OO-based abstraction, but they were not fully supported by all the features in C# and Java-like object-oriented programming (OOP). In addition, Graphical User Interface (GUI) was offered in C# programming language where it had a pre-built component in Visual Studio IDE compared to PHP. PHP language only integrated with HTML codes in designing the web design (Jiang Li et.al., 2010). Hence, C# language stressed out its specialty according to the several features above and helped in developing a better and complete system – which results in the selection of .NET MVC Framework that is supported by C# programming language.

SYSTEM DEVELOPMENT

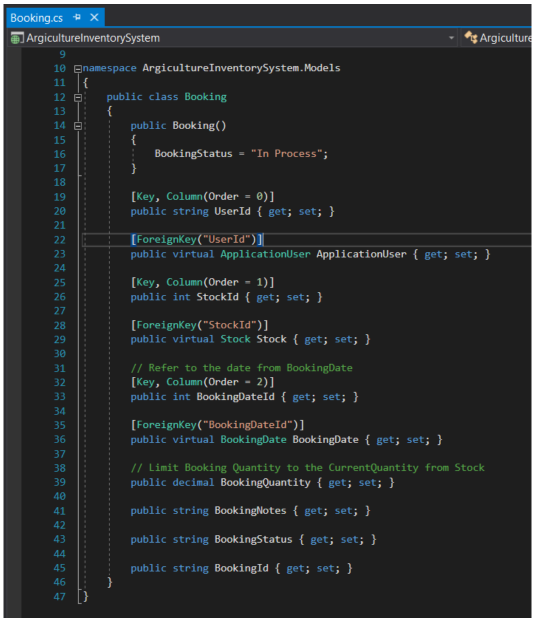
Developing the system by using the MVC .NET Framework shows the benefit of developing the components of the system separately; all the components could be worked on separately, it does not matter which comes first. In this section, the development process is described in three separate sub-section; 1) Database Design - Model 2) Interface Design - View, and 3) Business Logic - Controller.

Database Design – Model

The information displayed on the website primarily originates from the database. All the data have been captured, collected and stored in the database which is necessary for developing the system. In addition, the database technology is widely used in any kind of business application.

MySQL Server is used as the database technology in completing the development. The database file is created in a form of .mdf format in this MySQL server. The database content is generated by using Code-First features provided by the MVC .NET Framework.

The Model needed to be defined first by defining the tables and columns of the database. The design of the database is finalized through the Entity Relationship Diagram (ERD) and it is referred to create the Model. The ERD design is reflected in the Model where each of the Model represents the entity, and the attribute inside the model represents the attribute inside the created entity. After the creation of the Model, a migration feature - which is provided by the Code-First feature, allows the Model to be then converted into a table (Model) and column (attribute) inside the system database. Booking Model together with its attributes of the system is shown in Figure below:



```
namespace AgricultureInventorySystem.Models
{
    public class Booking
    {
        public Booking()
        {
            BookingStatus = "In Process";
        }

        [Key, Column(Order = 0)]
        public string UserId { get; set; }

        [ForeignKey("UserId")]
        public virtual ApplicationUser ApplicationUser { get; set; }

        [Key, Column(Order = 1)]
        public int StockId { get; set; }

        [ForeignKey("StockId")]
        public virtual Stock Stock { get; set; }

        // Refer to the date from BookingDate
        [Key, Column(Order = 2)]
        public int BookingDateId { get; set; }

        [ForeignKey("BookingDateId")]
        public virtual BookingDate BookingDate { get; set; }

        // Limit Booking Quantity to the CurrentQuantity from Stock
        public decimal BookingQuantity { get; set; }

        public string BookingNotes { get; set; }

        public string BookingStatus { get; set; }

        public string BookingId { get; set; }
    }
}
```

Figure 2: Booking Model with its Attributes

Interface Design – View

Essentially, the system has been developed by using Microsoft Visual 2015. It is one of the open source software where built with the snippet and syntax highlight to support the debug. Those are several features implemented in this software that facilitate the code writing effectively. With the support of the MVC framework, all the system interfaces are compiled in one folder called View. The View file consists of .cshtml files which is a file that could contain html, together with C# programming language, together with other programming language that supports interactive behaviour such as Javascript.

Besides, the interface of the system designed by selected from Bootswatch themes. Bootswatch is a collection of open source themes for bootstrap where the developers can choose any kind of templates according to their preferences. It such an accommodating tool that helps creates a simple yet interesting and attractive website.

In this project, Bootstrap Lumen is chosen to design the layout of the system. For instance, the main page of the system displayed several menus such as View All Bookings Menu, Manage Users & View Booking, Manage Stocks, Manage Department/Faculty, Generate Report and View Booking Status. Figure 3 illustrates the main page of the administrator of the system developed.

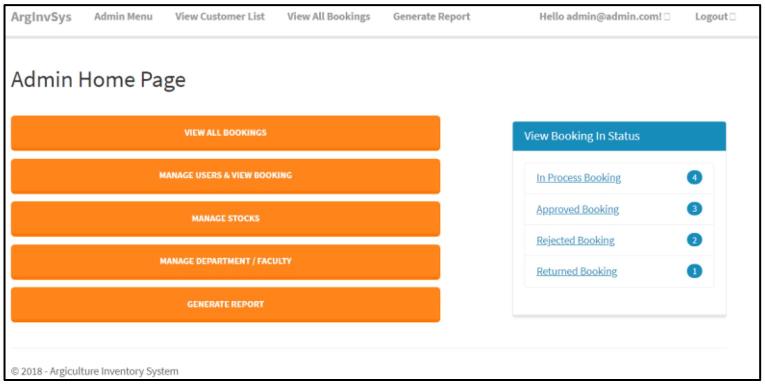
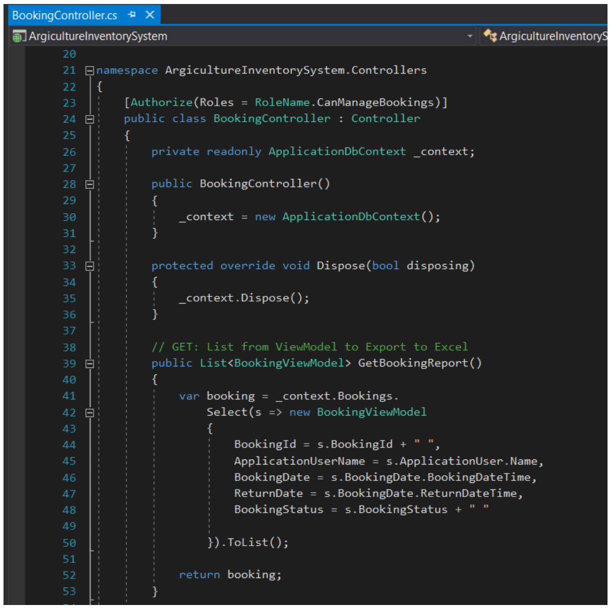


Figure 3: Main Page for Administrator

Business Logic - Controller

All the functionality available in this system such as create booking is implemented in the Controller of the MVC Framework. Each Model has its own Controller, means the Model could create its own functionality within their own boundary. As an example, the Booking Model had their own Controller which contain the function for Booking creation, update, viewing and delete action – which did not interfere with other Model such as Stock. Figure 4 illustrates the code snippet for the Booking Controller:



```
namespace ArgicultureInventorySystem.Controllers
{
    [Authorize(Roles = RoleName.CanManageBookings)]
    public class BookingController : Controller
    {
        private readonly ApplicationDbContext _context;

        public BookingController()
        {
            _context = new ApplicationDbContext();
        }

        protected override void Dispose(bool disposing)
        {
            _context.Dispose();
        }

        // GET: List from ViewModel to Export to Excel
        public List<BookingViewModel> GetBookingReport()
        {
            var booking = _context.Bookings.
                Select(s => new BookingViewModel
                {
                    BookingId = s.BookingId + " ",
                    ApplicationUserName = s.ApplicationUser.Name,
                    BookingDate = s.BookingDate.BookingDateTime,
                    ReturnDate = s.BookingDate.ReturnDateTime,
                    BookingStatus = s.BookingStatus + " "

                }).ToList();

            return booking;
        }
```

Figure 4: Booking Controller functions

All the entities that involved in the system has its own separate Model, View and Controller. A new entity could be added anytime in the development process which reflect the MVC framework extend-ability feature. It aids significantly in implementing a new function needed in the future.

RESULTS AND DISCUSSION

Test Script has been prepared as a checklist to test the system functionalities. The script consists of functional requirements and non-functional requirements. Functional requirements are highlighted on the functionality and the usability of the system. Meanwhile, the non-functional requirements are specified on the usability and the effectiveness of the system.

Both requirements have been evaluated to meet the user requirements. Furthermore, these functionalities are classified into three level of priorities; Mandatory (M) which had the highest priority, followed by Desirable (D) and Optional (O). The classifications are important in determining which functions needed to be implemented and assessed thoroughly. An example of Authentication functions with its description and priority are shown in the Table 1 below:

Table 1: Authentication Functional Requirement with Priority

No.	Requirement ID	Requirement Description	Priority
	IMS_01	Authentication System (Administrator, University Community)	
1.	IMS_01_01	New user can register as university community users by providing the information required.	M
2.	IMS_01_02	User must enter username and password to login.	M
3.	IMS_01_03	Entering wrong email format will show error message.	D

4.	IMS_01_04	Message will be prompted if username and password are not match.	D
5.	IMS_01_05	Admin user will be redirected to the admin page. The admin page shows menus consist of: view all bookings menu manage users& view booking manage stocks manage department/faculty generate report view booking status	M
6.	IMS_01_06	User will be redirected to their own list of booking.	M
7.	IMS_01_07	User can change the password/information by clicking on their username/email on the top right menu in the page.	O

The testing process includes the developer team to test all the sub-functions listed in sequence of Requirement ID (as shown in Table 1) to identify any errors occurred when specific function is used. The developer team-testing is delivered by testing the detailed function by entering varieties of possible inputs into the text field. These actions are important to ensure that the system delivered the correct outputs and the functionality of the system is well integrated.

The output of the testing is the Test Script checklist of the system functions which has been delivered successfully. The adaptation of the .NET MVC Framework has a noteworthy contribution to the success of the testing by providing its separation of components aspect. Most of the exceptions when using the functions are handled in the Controller component of the MVC Framework. All possible inputs are filtered through the Controller and the exception handle management become easier because of the separation of concern between the three components. Any additional error handler could be implemented easily without interfering with other components.

CONCLUSION

The use of correct technology or platform is crucial as the technology trends are changing in fast rate. An information system development process involved a lot of scalability as it needs to aid for a long-term information gathering. The MVC .NET Framework that is used in this project is a suitable platform as it is one of the current trend that is currently being used widely. The maintainability of the MVC .NET Framework is one of the highlighted characteristics. The framework itself could be referred as the documentation of the system and this allows future developers working on the same system to easily understand and continue evolving the system.

Functionality test has been conducted by involving the developer team to test all the functions and sub-functions in the system by using the Test Script checklist. The test has been done successfully, showing that the functions are well-integrated due to the support of .NET MVC Framework that are adopted in the system development process.

REFERENCES

Chadwick, J., Snyder, T.& Panda, H., (2012). Programming ASP.NET MVC 4 Jes. (R. Roumeliotis, Ed.) (First Edit). United States of America: O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

Gitau, R. W. (2016). Inventory Management Practices And Organizational Productivity In Parastatals In A Research Project Submitted In Partial Fulfillment Of The Requirements For The Award Of Degree Of Master Of Business Administration , School Of Business , University of Naroibi.

Jain, N., Mangal, P.,and Mehta, D., (2014). AngularJS : A Modern MVC Framework in JavaScript, 5(12).

Jiang Li, Sleiman Rabah, Mingzhi Liu, Y. L. (2010). Comparative Studies of 10 Programming Languages within 10 Diverse Criteria, 1–139.

Nyemba, W. R., & Mbohwa, C. (2017). Design of a 10-digit Inventory Codification System for a Tube and Pipe Manufacturing Company in Zimbabwe. Procedia Manufacturing, 8(October 2016), 503–510. https://doi.org/10.1016/j.promfg.2017.02.064

Pop, D. P., & Altar, A. (2014). Designing An MVC Model For Rapid Web Application Development. Procedia Engineering, 69, 1172–1179. https://doi.org/10.1016/J.Proeng.2014.03.106.

Prokofyeva, N., & Boltunova, V. (2017). Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems. Procedia - Procedia Computer Science, 104(December 2016), 51–56. https://doi.org/10.1016/j.procs.2017.01.059