International Journal of Engineering Research And Management (IJERM)
ISSN: 2349- 2058, Volume-05, Issue-01, January 2018

BWE+: The Design and Implementation of a B2B Platform for Connecting Angolan and Chinese Companies

Edgar Lukemba Fernandes Agostinho, Dr. An Ning, Jamal Jason Alwin Solomon

Abstract— **Managing a B2B e-commerce platform can be a difficult task for programmers in the long run, due to the complexity of required features. There are quite a few 'solutions' that are available, however, many limit the ability to extend and customize, especially the UI. By using ASP.NET MVC, .NET Entity Framework & ASP.NET MVC Identity, creating a full-feature B2B platform that allows for easy customization of the UI is simplified greatly. These three technologies make it easier to develop and maintain large-scale web applications easier by decreasing dependencies between the application layers. Improving testability by supporting test driven development, also improves development efficiency and software reusability.**

The application functionality is separated into three components, the presentation tier (UI), the business tier (business logic), and the data tier (database tier).

The system is built to take advantages of the "one belt one road initiative" created by the Chinese government. The system will function as a window to strengthen and create new business opportunities between companies of both countries. Satisfy a variety of market demands and enhance cooperation.

Index Terms— **B2B e-Commerce, e-Commerce Website Platform, ASP.NET MVC.**

I. INTRODUCTION

E-commerce is a modern way of conducting business, by buying, selling goods and services over the internet. Providing to the end user the ability to compare prices and products before purchase.

As of 2011, Angola is China's second biggest trading partner in Africa [1]. Since then many projects have been launched between the two countries, and more Chinese companies have established a presence in the Angolan market. But these transactions are still bellow desired results, due to the lack of a platform dedicated for conducting business between both countries enterprises making it easier to communicate and located them.

BWE+ is a B2B e-commerce platform with the main purpose to host business transactions between Angolan and Chinese companies, transactions such as buying, selling products or raw materials and services. To enhance cooperation, boost economy growth and satisfy market demand we design and develop the system based on Asp.Net MVC technology. Allowing reduction of consumers' search cost, speed up sale transactions and efficient monitoring of every basic operations. With BWE+ not only will improve Chinese business in Angola but also consequently will allow an availability of Angolan products in the Chinese market for purchase.

Edgar Lukemba Fernandes Agostinho, College of Computer Science, Hefei University of Technology, Hefei, China, (+86) 188-5697-8489, (e-mail: e_2203@hotmail.com).

Dr. An Ning, His Department Name, College of Computer Science, Hefei University of Technology, Hefei, China, (+86) 180-1995-6086, (e-mail: ning.g.an@qq.com).

Jamal Jason Alwin Solomon, College of Mathematics & Computer Science, Anhui Normal University, Wuhu, China, (+86) 155-5530-9250, (e-mail: jamalsolomon@hotmail.com).

II. SYSTEM ARCHITECTURE

The System, built combining the Asp.Net technology and MVC model, is based on a three-tier architecture (the presentation tier, the business tier, and the data tier) to best process clients request and other system operations.

Asp.Net gives a great support to build websites and web applications using HTML, CSS, and JavaScript. It gives a sophisticated, pattern-based form of building dynamic web applications that allows for a clean separation of concerns and gives the developer full control over markup and agile development.

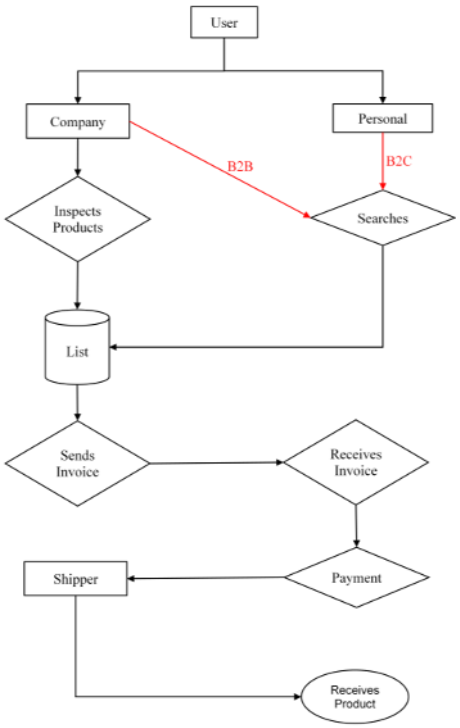

Figure 1 - The System Chart

To prevent unauthorized requests, ASP.NET MVC uses Authentication filters, that run prior to authorization filters in the ASP.NET MVC pipeline and allows developers to specify authentication logic per-action, or to all controllers.

The .Net Entity Framework works as an object-relational mapper that enables developers to work with relational data using domain-specific objects, eliminating the need for most of the data-access code that developers usually need to write.

By combining the features of ASP.NET MVC with jQuery (JavaScript Library) and Bootstrap (a famous CSS Framework) the system will comprise of three main layers. The presentation layer which operates on the client side will be responsible for the user interface, collecting client's requests and sending them to the web server. The web server layer, which contain both the ASP.NET MVC and the Entity Framework, will process request from the user performing business logic operations and interact with the data layer. The data layer is where the system data will be stored permanently and is made up of a database accessed via a MySQL database server.

Figure 1 shows the system chart displaying the process of the basic daily operations of the system.

A. *The System Software Model: ASP.NET MVC*

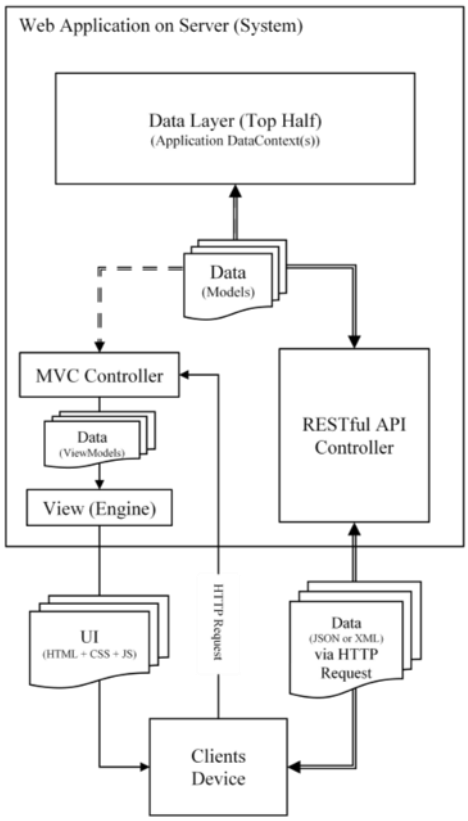

Figure 2 Flow of Data between the User's Device and the System's Components

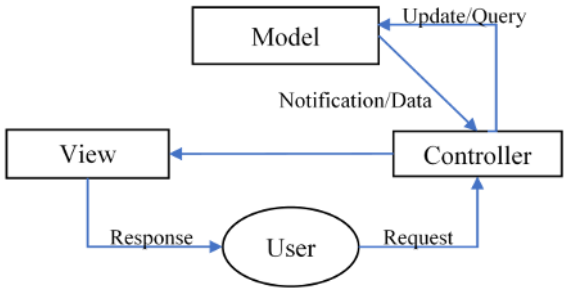

Figure 3 Interaction between a User and a MVC System

1) *The MVC Model*

Introduced into Smalltalk-76 (an object-oriented, dynamically typed, reflexive programming language) by Trygve Reenskaug in the 1970s, MVC is a software architectural pattern for implementing user interfaces by dividing a given application into three interconnected parts. These are:

- The *model*—stores data that is retrieved according to commands from the controller and displayed in the view.
- The *view*—generates new output to the user based on changes in the model.
- The *controller*—sends commands to the model to update the model's state and commands to its associated view to change the view's presentation of the model.

Using this method, we can separate the way the data is stored within the system, from the way the information is presented to the user; allowing us to reuse code and develop the system efficiently. In figure 3, an example of how a user and the model, view and controller of a web-based MVC application would interact.

For this reason, even though it was traditionally used for desktop graphical user interfaces (GUIs), this architecture has become popular for designing web applications and even mobile, desktop and other clients [2].

Popular programming languages like Java, C#, Ruby, PHP and others have popular MVC frameworks that are currently being used in web application development straight out of the box.

Some web MVC frameworks follow a 'thin client' approach that places almost the entire model, view and controller logic on the server (e.g. Django, Rails and ASP.NET MVC). The client sends either hyperlink requests or form submissions to the controller and then receives a complete and updated web page (or other document) from the view; the model exists entirely on the server [3].

Other frameworks such as AngularJS, EmberJS, JavaScript MVC and Backbone allow the MVC components to execute partly on the client-side.

2) *ASP.NET*

ASP.NET is a unified Web development model that includes the services necessary to build enterprise-class Web applications with a minimum of coding, and is part of the .NET Framework thus, when coding ASP.NET applications one have access to classes in the .NET Framework. Applications can be coded in any language compatible with the common language runtime (CLR), including Microsoft

Visual Basic and C#, thereby enabling the development of applications that benefit from the common language runtime, type safety, inheritance, and so on [4].

It was first released in January 2002 with version 1.0 of the .NET Framework, and is the successor to Microsoft's Active Server Pages (ASP) technology. The ASP.NET SOAP (originally Simple Object Access Protocol) extension framework allows ASP.NET components to process SOAP messages. ASP.NET's successor is ASP.NET Core.

3) ASP.NET MVC: Integration of MVC into ASP.NET

ASP.NET MVC is a web application framework that Microsoft developed and released initially December 2007. It is based on ASP.NET and implements the MVC pattern, allowing developers to build a web application as a composition of three roles (which are defined as 3 logic layers of a web application by the MVC model): Model (the business layer), View (the presentation layer) and Controller (the input control layer). ASP.NET MVC is open-source (with the exception of the proprietary ASP.NET Web Forms Component), and its source code was released initially under the Microsoft Public License in April 2009 [5].

ASP.NET MVC, Razor and Web API was later released under the Apache License 2.0 in 2012 [6] and these components have been merged into MVC 6.

In figure 2, how data would flow between the client user's device and the system, is displayed. In ASP.NET the models are passed between the DataContexts (which provides an interface between the system and the database server) and the controllers. The ViewModels represent the data the Controllers would pass to the Views, there are different from the Models they would be associated with as they would contain some (but not all) of the information that is stored within the model.

B. The Data Model

Our system uses MySQL server to host the database, which will be encrypted and user-password protected by the server. Direct access to and manipulation of the database will not occur and steps will be taken to prevent SQL injection attacks on the database.

To allow the system to effectively communicate with the database server and be able to perform create, read, update and delete (CRUD) operations as required by the business logic, we use Microsoft's Entity Framework.

Entity Framework, initially released in august 2008, is an open source [7] Object-relational mapping (ORM) Framework for ADO.Net, and was originally a part of the .Net Framework. It is a set of technologies in ADO.Net that support the development of data-oriented software applications, enabling developers to work with data in the form of domain-specific objects and properties without having to worry about the underlying tables and columns in the database where the data is stored.

III. Development and Implementation of the System

A. Client Layer Design and Development

Using a combination of JavaScript, (responsive) CSS frameworks; namely jQuery and Bootstrap; we will create an interactive responsive website that would be optimized to combine the best design without sacrificing performance. The website will utilize HTML5 and CSS3 at the basic level and the scripts will interact with the system's RESTful API. A RESTful API (also known as a RESTful web service) is an application program interface, based on representational state transfer (REST) technology, which (in our case) uses HTTP requests to pass data between the client layers of the system. This feature allows for mobile apps (android, iOS, windows, etc.) to be easily integrated into the system without (directly) affecting the development of the website itself.

B. Web Layer Design and Development

The web layer is where the system 'lives', and it consists of the website application running on an IIS server (which will, in turn, be running on a Windows server machine). Here, the following components of the system are dealt with:

- Routing and SEO: In ASP.NET, routing is handled by configuring the routes that would be used within a file called RouteConfig.cs. In this file we define a static function that will define the routes used by application as is necessary as shown below

```
public static void
RegisterRoutes(RouteCollection routes)
{
    //use to ignore a specific URL route

routes.IgnoreRoute("{resource}.axd/{*pathInf
o}");

    //uses to map a specified URL route and set
its default values
    routes.MapRoute(
     name: "Default",
     url:"{controller}/{action}/{id}",
     defaults: new
      {
         controller = "Home",
         action = "Index",
         id = UrlParameter.Optional
      }
    );
}
```

For the API, we define a separate function (found in another file known as WebApiConfig.cs) which will register the API configuration within the application and define the routes that will be used within that section (see below).

```
public static void Register(HttpConfiguration
config)
{
    // Web API configuration and services

    // Web API routes
    config.MapHttpAttributeRoutes();


    config.Routes.MapHttpRoute(
      name: "DefaultApi",
      routeTemplate: "api/{controller}/{id}",
      defaults: new
      {
        id = RouteParameter.Optional
      }
    );
}
```

To call these functions when the application starts we add the following lines within a function called Application_Start (found in the Global.asax.cs file in the website project's root folder).

```
//call function for API configuration
GlobalConfiguration.Configure(WebApiConfi
g.Register);

//call function for configuring and mapping
routes
RouteConfig.RegisterRoutes(RouteTable.Rou
tes);
```

By using well-defined routes within the application, the system will be SEO friendly.

- Request Handling: The request handling will be done by the controllers of the web application. In ASP.NET MVC there are two types of controllers, the (standard) MVC Controllers and the API Controllers. The MVC Controllers handle the basic webpage and section requests (e.g. the login page, user profile page, contact page) while the API Controllers handle the API requests entirely.

```
public class HomeController : Controller
{
  private AppDataContext dataContext = new
AppDataContext();

  public ActionResult Index()
  {
    //business logic goes here
    return View();
  }

  public ActionResult About()
  {
    //business logic goes here
     return View();
  }

  public ActionResult Contact()
  {
    //business logic goes here
    return View();
  }
}
```

An MVC Controller (as shown above), mainly has a list of functions that returns Views, while API Controller (as seen below) returns a HTTP response, sometimes with data either in JSON or XML format as requested.

```
public class PhoneApiController : ApiController
{
  [HttpPost]
  public HttpResponseMessage
RegisterApp(RegisterAppInfo appInfo)
  {
    //business logic goes here
    return
Request.CreateResponse(HttpStatusCode.OK, new
    {
      Success = true,
      AppId = 1000,
      IdAlias = "QAUBC9",
      appInfo = appInfo
    });
  }
}
```

C. *Business Logic Layer Design and Development*

In an application or program that follows the MVC pattern, the controller contains all the business logic for that particular application/program. In ASP.NET MVC, the Controller and ApiController are special inheritable classes that contains a set of methods (and in some cases, properties) which would contain the business logic. The methods within these two types of controllers would contain all the business logic used within the application itself which would be executed whenever a request is made that requires the use of that particular method.

The following diagram (Figure 4) shows how a request made from a JavaScript function to the PhoneApiController (see Request Handling in Web Layer Design and Development), would trigger the function RegisterApp within the controller.

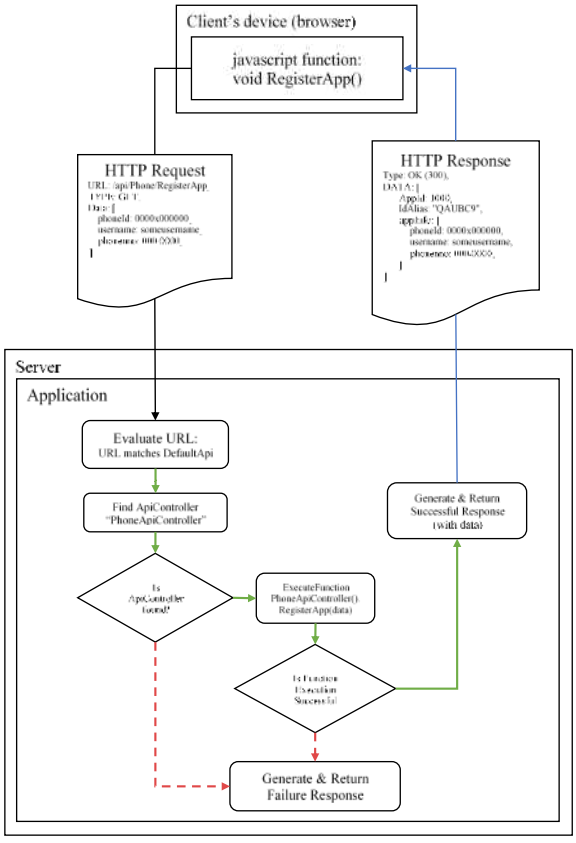

Figure 4 Example of Routing and Request Handling

D. Data Layer Design and Development

In addition to configuring the database server and the database itself (including creating a user to access the database, setting a database name, etc.) some additional configuration of the system was required before it could access (and interact with) the database successfully.

In our test environment we had to install 'MySQL for Visual Studio' and 'MySQL Connector/.Net' (in that exact order) [8]; then create a new project and configure it to connect to the database. By following an article located on Microsoft's technical documentation website [9], we were able to complete the configuration successfully. The basic steps that we followed (in chronological order) are as follows:

1) Created a new ASP.NET Web Application project in visual studio. This project was created using a template for an ASP.NET MVC website with 'Individual User Accounts' authentication and included additional references for Web API.
2) Using the NuGet Package Manager in Visual Studio we updated Entity Framework to the latest stable version (6.2.0) and installed the 'MySql.Data.Entity' version 6.9.10 package. We didn't use the latest version (6.10.4) because it has been reported to give problems (which we encountered when we tried to use it ourselves).
3) We updated the system's configurations which were stored in the web.config file in the root folder of the system. This includes replacing the original database connection(s) with our own that points to the MySQL Database that we created and configured previously and changing the database provider used.
4) To facilitate the changes, we added a new class to the model that defined a custom HistoryContext, DbConfiguration and Entity Framework initializer for the system to use in its database contexts
5) Finally we built and ran the system, and created a new dummy user account, double checking the MySQL server's database (using MySQL Workbench) to verify that the necessary database tables were created and that the user account information was successfully added.

With this the connection to the database was established and we moved on to coding the models that we would use to store data in the database to complete the data layer of the software.

IV. Conclusion

In today's programming world, some of the good programming practices include avoiding rigidity, fragility, immobility, or redundancy in the code, The Asp.Net MVC bring to us a pattern that divides the chain of responsibilities into Model, View, and Controller. This solution becomes more efficient when targeting large scale business applications. When the application is big and we expect changes to be a regular visitor to our development area, therefore the necessity to separate the whole application into components and modules and create a chain responsibilities with those components (every component have single responsibility) and since all those components are self-sustaining then we can test them without depending on any other components, thus we divide then refactor and eventually conquer.

The framework significantly reduces the amount of code required for building large and complex applications, thus making it faster. The .NET performance is increased by just-in-time compilation, smart caching technologies, and native optimization. Microsoft ensured high reliability and security due to built-in Windows authentication and per-application configuration. Another benefit is that the framework is regularly upgraded to meet the most up-to-date technology demands. It allows all processes to be smartly separated without losing the effectiveness. It enables development teams to apply agile development methods, for example, giving engineers an opportunity to develop thousands of lines of code, while designers are able to work on the interface at the same time, without disturbing each other.

Using the Asp.Net MVC solution for BWE+ platform will enable us to have full control over the components of the system as the business grows, making it easier to add more features if necessary or according to new demands and better manage the website in the long-term.

Acknowledgment

Edgar Lukemba Fernandes Agostinho expresses his deepest gratitude to his supervisor professor An Ning for his support and mentorship throughout this project. Edgar Lukemba Fernandes Agostinho thanks Mr. Biàn Yáng from the Anhui Construction Industry group (Angolan Office), for taking the time to answer the interview questions, and for giving his feedback on the BWE+ platform. Edgar Lukemba Fernandes Agostinho also thanks Jamal Jason Alwin Solomon for his support and contributions on writing this paper.

Jamal Jason Alwin Solomon thanks Edgar Lukemba Fernandes Agostinho for the opportunity to contribute to the writing of this paper.

References

[1] C. Briefing, "The China-Angola Partnership: A Case Study of China's Oil Relations in Africa," 25 May 2011. [Online]. Available: http://www.china-briefing.com/news/2011/05/25/the-china-angola-partnership-a-case-study-of-chinas-oil-relationships-with-african-nations.html.

[2] I. Davis, "What Are The Benefits of MVC?" 9 December 2008. [Online]. Available: http://blog.iandavis.com/2008/12/what-are-the-benefits-of-mvc/.

[3] A. Leff and J. T. Rayfield, "Web-Application Development Using the Model/View/Controller Design Pattern," in IEEE Enterprise Distributed Object Computing Conference, 2001.

[4] Microsoft, "ASP.NET Overview," 28 November 2017. [Online]. Available: https://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx.

[5] S. Guthrie, "ASP.NET MVC 1.0," 2 April 2009. [Online]. Available: https://weblogs.asp.net/scottgu/asp-net-mvc-1-0.

[6] S. Guthrie, "ASP.NET MVC, Web API, Razor and Open Source," 28 March 2012. [Online]. Available: https://weblogs.asp.net/scottgu/asp-net-mvc-web-api-razor-and-open-source.

[7] Microsoft, "Entity Framework," 12 December 2012. [Online]. Available: https://docs.microsoft.com/en-us/aspnet/entity-framework.

[8] Oracle Corporation, "MySQL::MySQL for Visual Studio::2 Installing and Configuring," 28 November 2017. [Online]. Available: https://dev.mysql.com/doc/visual-studio/en/visual-studio-install.html.

[9] M. Markowski, R. Soares De Almeida and R. McMurray, "ASP.NET Identity: Using MySQL Storage with an EntityFramework MySQL Provider (C#)," 12 October 2013. [Online]. Available: https://docs.microsoft.com/en-us/aspnet/identity/overview/getting-started/aspnet-identity-using-mysql-storage-with-an-entityframework-mysql-provider.

Edgar Lukemba Fernandes Agostinho is currently enrolled in the master's degree program in Computer Applied Technology from Hefei University of Technology, China, and is conducting research in Software Engineering. In 2014, he obtained bachelor's degree in Computer Science and Technology from the University of Science and Technology of China. He is also a 2008 National contest winner of the SADC secondary schools essay competition, under the topic "What measures should be taken to ensure that the SADC free trade zone could benefit the citizens of the region in fighting poverty"; and was awarded full scholarship by the Angolan Education Ministry and China Scholarship Council in 2009 and again in 2015.

Dr. An Ning has a PhD in Computer Science and Engineeringfrom the Pennsylvania State University and is an IEEE Senior Member and ACM Life Member. After working for Oracle USA, Inc. as technical staff for ten years, Dr. An joined Hefei University of Technology in China as a "Yellow Mountain" Professor and founded the Gerontechnology Lab which is the first of its kind in mainland China. He is the PI for the International Research Base for Developing Innovative Gerontechnology sponsored by Chinese Ministry of Education and State Administration of Foreign Experts Affairs, and serves as the PI for several research projects including one sponsored by International S&T Cooperation Program of China. His current research interests include data sciences & engineering, gerontechnology, mobile health technologies, health communication, and digital learning. In addition to publishing more than thirty papers in leading international journals (including VLDBJ and TKDE) and top international conferences (including VLDB, SIGIR, ICDE, WWW and ICDM), Dr. An has published two books and one book chapter, and wrote three US patent applications (one granted) and three Chinese patent applications (one granted).

Jamal Jason Alwin Solomon is a student of Anhui Normal University in China, currently pursuing a bachelor's degree in Computer Science & Technology; and is a website developer specializing in the development of ASP.NET websites. In 2012, he was awarded a full scholarship by China Scholarship Council (CSC) in collaboration with the Prime Minister's Office of Antigua & Barbuda, after coming out as one of island's top students in the 2012 Caribbean Examination Council's (CXC) Caribbean Advanced Proficiency Examinations (CAPE); obtaining a CAPE associates degree in Computer Science. He obtained 1st place in the Jaycees Public Impromptu Speaking Competition and 2nd place in the Jaycees Public Speaking Competition in January 2009 in Antigua. During his years at the Antigua Gramma School, he was a part of the Junior Science Quiz team which won the National Inter-School Science Quiz Competition in 2009. He graduated from the Antigua Grammar School with honors in Information Technology in 2010.