A Repository Application Developed using .NET MVC and Angularjs for Interoperability Enabling Platform Sales Improvement in Healthcare Industry

Jnaneshwari Math[1] and Dr. Shobha G[2]

Department of Computer Science and Engineering, Rashtreeya Vidyalaya College of Engineering, Bangalore, Karnataka, India.

---***---

Abstract - *Healthcare Industry provides healthcare products and solutions for diagnostic, treatment, and preventive care. Interoperability between these products and solutions becomes the biggest challenge in the healthcare industry because of their different architectures. The success of interoperability enabling platform depends on the customer satisfaction, support and services by the vendors or organization, finding the opportunity in improving the sales. This is achieved with the visibility or analysis of the available data. There is need for a single application, which stores all the information and serves needs of the internal customer (other business units), sales team and financial team, implantation and service team, support team.*

1. INTRODUCTION

Healthcare Industry provides healthcare solutions for diagnostic, treatment, and preventive care. In healthcare, interoperability is ability to make the healthcare products and solutions to communicate and exchange the data. Interoperability becomes the biggest challenges in the healthcare industry because the different systems have the different architecture and infrastructure. The various software are present in the industry that provides standards-based point of interoperability, which enables interoperability between solutions and reduces the complexity and the cost in the healthcare environment. Such interoperability-enabling platform is installed on the thousands of the hospitals who wants interoperability feature between the devices. These support hundreds of product lines hence it is very difficult to track, remember the sites and products installed on the sites. There is need for a single application, which stores all the information and serves needs of the internal customer (other business units), sales team and financial team, implantation and service team, support team.

An application developed that keeps a record of the detailed information of the customers, the products with its configuration and settings. With visibility provided by application into 'what the customer has', implementation and services team has ability to plan their effort more efficiently and deliver the services with minimal efforts and allows reuse of assets already available in the customer environment. It allows faster troubleshooting of the product issues reported by customer. Repository application is developed using .NET MVC Frame Work and Angularjs

A. ARCHITECTURE OF REPOSITORY APPLCATION

The architecture of Repository application is shown in the Figure. 1. The overall system architecture consists of three layers. The first user interface layer made of Angular JS and HTML to interact with the user. The second intermediate layer consisting of the java classes, which handles the interaction between the user and the database. The third layer is the SQL database that holds all data for the application. The second layer consists of four layers (Model, View, Controller class) for better performance.

As seen in Figure.1 the user sends the requests (POST, PUT, GET, DELETE) to the front end, which will pass it on to the middle layer where depending on the type of request, made the corresponding function is called. The Controller, which is a part of the middle layer, makes a request to the database (MySQL) and the fetched result will traverse back to the user in the same direction. Database contains the three types of data product related data, sales data and the configuration files.

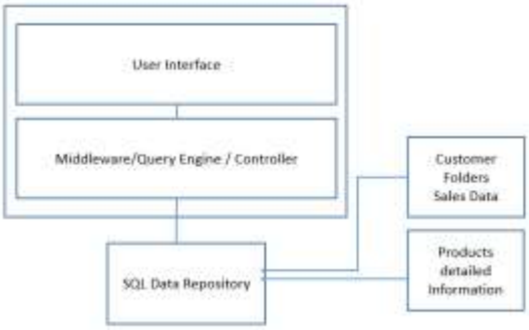
User Interface
Middleware/Query Engine / Controller
Customer Folders Sales Data
SQL Data Repository
Products detailed Information

Figure 1. Architecture of repository application

2. LITERATURE REVIEW

A concise background study was done to understand about MVC architecture and Angularjs framework. This helped to know the best solutions for development of application using MVC architecture and Angularjs framework. Each approach follows different methodology. Extracts from a couple of the papers/journals are as follows.

With the rapid changes that occur in the area of Web technologies, the porting and adaptation of existing Web applications into new platforms that take advantage of modern technologies has become an issue of increasing importance[1][2]. A reengineering framework provided to Transform Legacy Web applications to the MVC Architecture and enabled for the Java™ 2 Platform, Enterprise Edition (J2EE).

Diana M.Selfa et.al. proposed a database and web application based on MVC architecture which has had wide acceptance for corporation software development. It plans to divide the system in three different layers that are in charge of interface control logic and data access, this facilitates the maintenance and evolution of systems according to the independence of the present classes in each layer[3].

Rich Internet Applications (RIA) have been proposed as a response to growing user needs. RIA is based on the MVC and Model driven architecture[4].

The following are the some of the advantages of the MVC architecture, 1) MVC provides parallel and rapid development of the application. One programmer can work on the view while the other programmer working on the controller to crate business logic of the application hence it is faster than the other design pattern. 2) One can create the multiple view to the same model with minimum code duplication because it separates the data and business logic from the display. 3) There are more changes to the UI as compared to the business rules as time passes. View changes made without affecting the Model or entire application. 4) It returns the data without formatting. 5) It supports SEO friendly Development platform.

3. METHODOLOGY

The following steps form the methodology,

1. Create a back end database for customer's details, equipment details, sales, Users details, products details, sales orders, and metadata tables using MySQL. Site of the data is loaded by running the PowerShell Script.

2. Develop application based on the ASP.NET MVC architecture where it has 3 parts:
    a. Model: It maintains application data and business logic
    b. View: It is a user interface, which displays data to the user and allow the modification of the data. It is to be created using HTML and Angularjs
    c. Controller: It handles the request from the users.

3. Build and Publish ASP.NET code and deploy it on the IIS (Internet Information Services) Manager.

A. EXPERIMENTAL SETUP

Broadly, the files are divided into two parts, one that consists of the C# files and the second part includes the Angular JS and HTML files. The file structure is shown in the Figure 2.

App_Data: This folder contains application data files like LocalDB and xml files etc. IIS never serves these files. **App_start:** This files executed when the application starts. These are configuration files. **Content:** The folder contains static files like CSS, icon and images files etc. **Controllers:** This folder contains class files of controller, which handles the request from the user and returns response back to the user. **Fonts folder:** This contains all the costume font files of the application **Model:** This older contains the class files which includes public properties used by the application to hold and manipulate the data. **Script:** The folder contains .js files of the application. **Views:** This contains the html files of the application the view file is .cshtml file where we can write html code. **.Golbal.asax:** This file allows writing the code that runs in the response to the application level events. **Packages.config:** This folder managed by the Nugent to keep track of the packages are installed along with their versions in the application. **Web.config:** This file contains the configuration files at application level

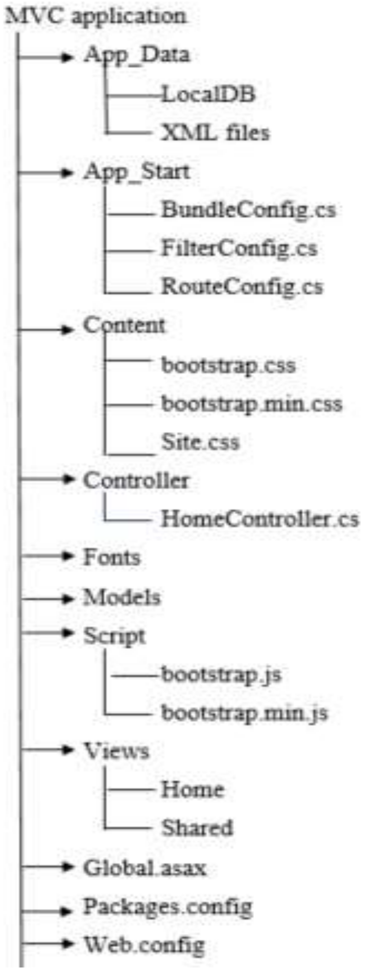

Figure 2. File Organization of InstallBase application.

B. EVALUATION METRIC

The performance of the application depends on the quality of data entered by the user in case of a search. Partially correct inputs are accepted while filtering the data as Angular JS make a provision for this. In case there are any spelling errors in the input the application will fail to fetch the results. To avoid spelling errors application is provided with the dropdown for the each input. . In case of right inputs the results are 100% accurate. The time taken to fetch the results or display any changes made is less that one second.

4. RESULTS AND OBSERVATIONS

MVC architecture provide accurate results within a short duration. Development of the MVC application is faster. Angular JS is helpful in retrieving results even with partial inputs. A constant Internet connection is needed for extending Angular JS and Bootstrap scripts. In case it is not available the view will not have the enhanced user appeal.

5. CONCLUSION AND FUTURE WORKS

The main aim of developing the single application to serve the needs of sales team, service and implementation team and the customer is achieved by developing the repository application using .NET MVC framework and Angularjs. With detailed visibility of data in the application, sales of interoperability enabling platform can be increased.

For future work, we try to include metadata architecture to provide ease of use, maintainability, supportability and zero downtime. Metadata driven architecture makes the UI Customizable.

REFERENCES

[1] N. Jain, P. Mangal, and D. Mehta, "AngularJS: A modern MVC framework in JavaScript," J. Global Research in Computer Science, vol. 5, no. 12, 2015, pp. 17-23.

[2] Yu Ping Kostas Kontogiannis Terence C. Lau, "Transforming Legacy Web Applications to the MVC Architecture", Dept. of Electrical & Computer Engineering University of Waterloo Waterloo, ON. Canada.

[3] Diana M. Selfa, Maya Carrillo, Ma. del Rocío Boone, "A Database and Web Application Based on MVC Architecture" , Benemérita Universidad Autónoma de Puebla 14 Sur y Av.San Claudio, Edif. 135, Ciuda Universitaria.

[4] Lay-Ki Soon, Sang Ho Lee, Enhancing URL Normalization using Metadata of Web Pages, in International Conference on Computer and Electrical Engineering, 2008.

[5] Voon Yang Nen, Onn Ching Ann, Pigeon-Table: AQuick Prototyping Tool using Twitter Bootstrap and Angular JS for Data-Driven Web Application Development, IEEE, 2017.

[6] Runsha Dong, Fei Su, Shan Yang, Lexi Xu, Xinzhou Cheng, Weiwei Chen, Design and Application on Metadata Management for Information Supply Chain, IEEE, 2016.

[7] LIU Yuan-jie, HUANG Ting-lei, WANG Xin, Research and Application of Metadata in Information Management, IEEE, 2010.

[8] Aryo Pinandito, Hanifah Muslimah Az-zahra, Lutfi Fanani, Anggi Valeria Putri, Analysis of Web Content Delivery Effectiveness and Efficiency in Responsive Web Design Using Material Design Guidelines and User Centered Design in International Conference on Sustainable Information Engineering and Technology, IEEE, 2017.