International Journal of Innovations in Engineering and Technology (IJIET)
http://dx.doi.org/10.21172/ijiet.82.039

Dynamic Web Based Online Banking and Fund Management Application

Pranay Bhasin
Department of Electronics and Communication Engineering
SRM University, Kattankulathur

K. Suganthi
Department of Electronics and Communication Engineering
SRM University, Kattankulathur

S. Nivash
Department of Electronics and Communication Engineering
SRM University, Kattankulathur

Abstract- This paper observes the current banking application development strategies and proposes an advancement of fund managing capabilities along with a good design construct for online banking applications.

Keywords – banking, web devlopment, java, fund management, javaee

I. INTRODUCTION

As the banking sector is expanding, quality and efficiency of web-applications play a major role in its success. The use of strong web-application architecture with strong development platform will not only make the application robust but also more secure giving the web-application an ability to meet changing and demanding customer requirements in efficient manner. Incorporating a Model View Controller (MVC) design pattern as fundamental architectural design construct gives an edge on user interactive applications.

Online banking applications have upgraded to feature more advanced protocols and services and have been molded regularly to enhance the performance of the banking sector's operations and management. A more sophisticated and user friendly interfaces are being adopted. The best way to architecturally secure a banking app is to follow the MVC design pattern.

The Model/View/Controller design pattern is very useful for architecting interactive software systems. This design pattern is partition-independent, because it is expressed in terms of an interactive application running in a single address space. Applying the Model/View/Controller design pattern to web-applications is therefore complicated by the fact that current technologies encourage developers to partition the application as early as in the design phase. [2]

In this paper, we discuss the current online banking web application development and propose advancements on its design strategy while incorporating a more secure and feature rich approach on Java EE platform. This contribution will help small as well as large banks to have a more interactive and efficient web application for its clients. [1]

II. DESIGN STRATEGIES

An Online Banking Web Application Development Process organizes a very systematic and structural approach. The following should be considered before going into development phase;

(A) Defining the Application and the Purpose it's going to serve.

(B) Researching and Defining the clientage.

(C) Creating Functional Specific Documentation and the Services it's going to provide.

(D) Project Management Documentation.

(E) Technology Selection, Technical Specifications, Illustrative Diagram of Web Application Architecture and Structure, Development Methodology, Versions Control, Backups, Upgrades, Expansion and Growth Planning Document, Server Hardware / Software Selection

(F) Application Visual Guide, Design Layout, Interface Design, Wire Framing

(G) Database Structure Design and Web Application Development

(H) Testing: Quality Assurance, Multiple Browser Compatibility, Security, Performance - Load and Stress Testing, Usability

(I) Maintenance [3]

For a Banking application, the basic necessity comes at security. Using a normal JavaServerPages (JSP) scripts in a web page to communicate between the client and database is a major security drawback. To remove this drawback we follow the divided concept of Dynamic Web Applications called Model View Controller.

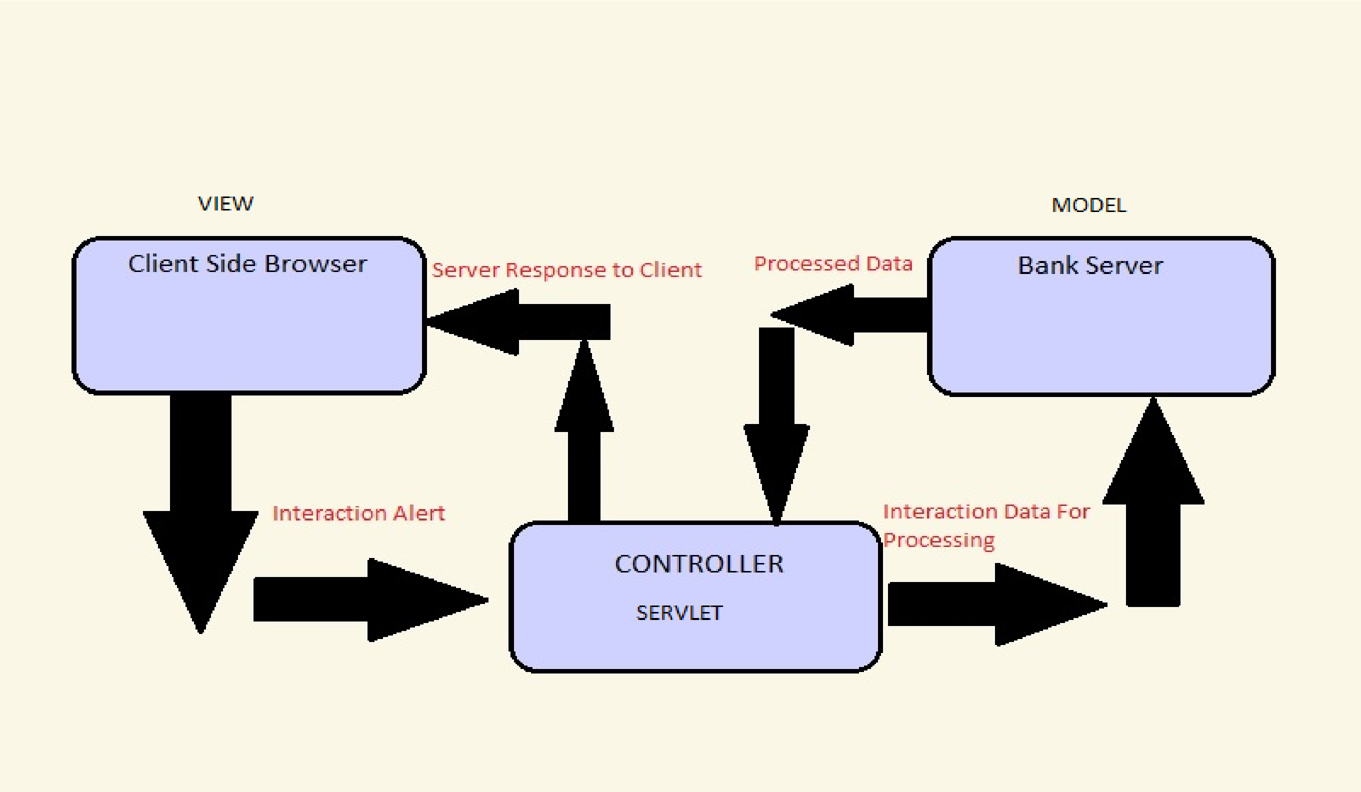

Fig. 1 Model View Controller flowchart

JavaEE Web Applications are dynamic web projects incorporating;

(A) Server Side Programming: These serve the functionality to the web servers. Provides backend functionalities such as processing the queries, fetching the results, communication with the database and other servers. Java provides the programmer with a lot of flexibility in incorporating itself either with the server side classes or directly with the client side webpages.

(B) Client Side Programming: Provides the user interface which are then processed by the browser. These are browser independent. The client side is commonly designed using HTML, CSS, Javascript, AJAX or JQuery.

The Backend serves as the Model of the Application while the front end serves as the view. The backend does not interact with the frontend directly to provide a more secured communication channel. Rather, it uses to the controller to fetch and retrieve the form inputs related to the transactions.

The key idea is to separate client from the backend data processing. The MVC design pattern applies to low-level user interaction such as individual keystrokes or activation of mouse buttons. The View conveys information to the client and the Controller handles the client's interaction with the application interface. The Model is the portion of the application that contains both the information represented by the View and the logic that changes this information in response to user interaction.[2]

III. Proposed Model

With the above design structures followed, an online banking and fund management system is proposed. The system is an online application that can be accessed throughout the organization and outside as well with proper login provided. The existing models are only transaction based. This paper proposes an advancement of money managing capabilities along a standard web based Banking System.

The developed system is an innovation in the area of private banking. The data entry process requires the operator to transact on the paper, which is then fed into the application by the operator thus increasing the chances of in accuracies in the typed content. Also the process includes higher transportation cost, increased handling cost, more time delays, low accuracy, more usage of resources like registers, books, papers, etc.

This application resolves this problem and the problem of management of expenses too.

Clients till now only had access to the standard transactional services like deposit, withdrawal, transfers etc. An advancement to that model brings along a database managed system which enables the client to manage or keep a record of his expenses throughout the day through the same application. It eliminates the need to keep traditional log books or diaries to keep the records. It helps the client to upload each of his expenses through an online form which is then stored in the database on the server.

The application has the view of distributed architecture, with centralized storage of the database handling and all the user interfaces have been designed using the HTML AND JSP.

Developed using Javaee MVC structure in view of the distributed client server computing technology. The user interfaces are browser specific to give distributed accessibility for the overall system. Database tables have been created for each user independently to take care of the bank records as well as to enter and save each users fund entries. At all proper levels care has been taken to check that the system manages the data consistency with proper business rules and validations. This project will help the customer in operating his bank account in a usual manner with additional features to keep a track of funds such as expenses using cash or card and will provide a database structured transaction sheet for the month to review where all the funds have been either transferred, received or spent.

IV. Database Management

Separate database tables would be maintained for each stage to keep a neat record of the client. The first table would keep a record of the entire bank database of the number of users, usernames and account numbers. The second would be a personal database where the user's personal details, account passwords and balance would be stored. The third would be maintained for each users expenses where date, cash/card, purpose of expense and spent amount would be stored.

I. USER DETAILS TABLE

Account Number	**Client Details**		
	Name	***Password***	***Balance***
1	Sample	123456	$1000

FIG 2. Example of Database Table

II. USER TABLE

Client ACCNo	
	Username
1	Sampleuse

FIG 3. User Table Example

III. EXPENSES TABLE

DATE	**Client Details**		
	Purpose	***Cash/Card***	***Amount***
1	Grocery	Cash	$1000

Fig. 4. Fund Management Table

V.CONCLUSIONS

This paper describes how the modern online banking web application design pattern can be used in the intrinsic environment of partitioned web-applications. This enables the code used to develop an independent stand-alone application and in compact address-space, to be deployed to various client platforms. It encourages the approach to web-application deployment in a way where it can be studied to get insight about how clients actually use the application. The paper was worked upon to let the clients have the facility of managing their expenses and along with having the transactional services provided by the bank.

VI. References

[1] Harshad B. Prajapati and Vipul K. Dabhi, Advance Computing Conference, 2009. IACC 2009. IEEE International "High Quality Web-Application Development on Java EE Platform"

[2] Avraham Leff and James T. Rayfield ,Enterprise Distributed Object Computing Conference, 2001. EDOC '01. Proceedings. Fifth IEEE International , "Web-application development using the Model/View/Controller design pattern "

[3] Bernard Kohan, Article Commentum 360 "http://www.comentum.com/guide-to-web-application-development.html" , "Guide to Web Application Development"

[4] Beck, K., Extreme Programming Explained: Embrace Change (XP Series), Addison-Wesley Pub Co., 1999.

[5] Bennett, B. et al, A distributed object oriented framework to offer transactional support for long running business processes, IFIPIACM Intemational Conference on Distributed Systems Platforms and Open Distributed Processing (Middleware 2000).

[6] Bergsten, Hans, JavaServer Pages, O'Reilly, 2000. 6. Betz, K., Leff, A., Rayfield, J., Developing Highly-Responsive User Inte$aces with DHTML and Servlets", Proceedings of the 19th IEEE International Performance, Computing, and Communications Conference -- IPCCC-2000, 2000.

[7] Buschmann, F. et al, Pattern-Oriented Software Architecture: A System of Patterns, John Wiley and Sons, 1996, 123- 168.

[8] Coutaz, J., PAC, An Object-Oriented Model for Dialog Design, Elsevier Science Publishers, Proceedings of HumanComputer Interaction - INTERACT, 1987,43 1-436.

[9] Enterprise JavaBeans Specipcations, http://java.sun.com/ products/ejb/docs.html, 2001.

[10]Flanagan, David, JavaScript: The Definitive Guide, 3rd, O'Reilly, 1998.

[11]Gray, G and Reuter, A. Transaction Processing: Concepts and Techniques, Morgan Kaufmann, 1993.