Multi Banking Transaction Interface Using MVC2 Architecture

Vivek Agrawal
Computer Engineering
SINHAGD INSTITUTE OF TECHNOLOGY
Lonavala, Pune, India
Vivekagrawal438@gmail.com

Vinay Kumar
Computer Engineering
SINHAGD INSTITUTE OF TECHNOLOGY
Lonavala, Pune, India
Vinay.snk1@gmail.com

Vishal Ghodke
Computer Engineering
SINHAGD INSTITUTE OF TECHNOLOGY
Lonavala, Pune, India
Vghodke55@gmail.com

Komal Temdhare
Computer Engineering
SINHAGD INSTITUTE OF TECHNOLOGY
Lonavala, Pune, India
Komal.temdhare22@gmail.com

ABSTRACT—BANKING SYSTEM PLAYS AN IMPORTANT ROLE IN EVERYONE’S LIFE. WITH THE GROWTH OF THE INTERNET WORLDWIDE THE WORLD HAS COME CLOSER AND CLOSER. SO, BANKS ARE ALSO PROVIDING NET BANKING WHICH PLAYS AN IMPORTANT ROLE IN ONE’S LIFE. ALL MONETARY TRANSACTION AND OTHER FACILITIES ARE BEING PROVIDED BY BANKS. EVERY USER HAS TO MAINTAIN HIS/HER PROFILE FOR DIFFERENT BANKS AND HAS TO REMEMBER USERNAME AND PASSWORD WHICH CAUSES OVERHEAD FOR THE USERS. THIS PAPER PRESENTS THE TRANSACTION INTERFACE OF DIFFERENT BANKING SYSTEM TOGETHER. THE PROPOSED METHOD IN THIS PAPER IS OF USING MVC2 ARCHITECTURE FOR SECURE E-BANKING SYSTEM. TODAY WE HAVE LOTS OF BANK AND FOR TRANSACTION WE HAVE TO TAKE USERNAME AND PASSWORD FOR EACH BANK TO DO THE TRANSACTION. BUT BY USING COMMON USERNAME AND PASSWORD WE CAN DO THE TRANSACTION OF THE BANK BY ADDING THE BANK DETAIL IN THIS WEBSITE. FOR SECURE E-BANKING WE ARE USING MVC2 ARCHITECTURE AND J2EE TECHNOLOGY AND WE ARE ALSO USING THREE LEVEL SECURITIES TO MAKE MORE SECURE AND RELIABLE.

KEYWORDS- MVC2 ARCHITECTURE, J2EE TECHNOLOGY, E-BANKING, JSP, TRANSACTIONS.

I. INTRODUCTION

A number of techniques have been developed for providing ease of business and for capability of the people. To provide more and more comfort levels, banks are also providing E-Banking [1] system which is the most important role plays in one’s life. As E-Banking is gaining popularity so bank has to provide a secure and authentication to the person.

But there are many banks and for each bank one’s has to remember separate username and password. There is no such platform where all banks come together and have a single username and password to do their transactions.

Multibanking Transaction Interface using MVC2 Architecture is a web based application which provides the common Interface for Multi Bank Accounts. Bank facilitates Online Transactions.

Multibanking transaction interface also provides support for other banking services provided by the banks. Multibanking transaction interface provides ease to the customer and at the same time assures high level security. Multibanking transaction interface also customized with different banking facilities.
II. MOTIVATION

Due to multibank in the market the user has to remember different user name and password of different banks for the online banking system which is not possible for the user to have it. To remember the user name and password for each bank is not so easy and its make the user irritation to have it. Due to which the users the neglecting the use of online transaction system.

To make the user friendly with the transaction system and online banking here we are introducing the MultiBanking Transaction Interface Using MVC2 Architecture.

III. SCOPE

I. Basic Features
   a. Create different system users and assign different roles with related permissions.
   b. Authenticate all the user of the system.
   c. Track all unsigned member of the system.
   d. Graphical report on user and sub admin and other details generated in easily downloadable format.
   e. All activities performed by the system will be maintained in the form of report/log for auditing and maintaining the integrity of the system.
   f. Java based client for the super user.
   g. Maintain centralized data base to provide security to information which can be accessed only by admin.
   h. Customizable Profile feature with predefined skins to choose from.
   i. Transaction details and all other activity will be maintained and watch by the admin.
   j. Admin will block and unblock the activity of the Sub-Admin and the User.
   k. Transaction will be done by single Interface.
   l. Allow to perform the different operations using different facilities provided by the banks.

II. Additional Features
   a. Easily changeable Database Location in between servers.
   b. Optimized AJAX usage for better user interaction.

IV. DRAWABACKS

The present MVC [2][4] Architecture did not give the complete separation between the application logic and business logic. If any one of the module we want to change then we have to change the whole module and it gets time consumed to do so this architecture was completely based on the page centric approach. In this model a Java Server Pages to program to control the Presentation, Business Logic and flow of the program. In this model the concept of the Business.

Logic was introduced. The business logic was hard coded in the form of the Java Beans and script lets and expressions. All this codes was used to write within the JSP page. So, here we are proposing the MVC2 Architecture which gives the complete separation between the application logic and business logic.
V. LITERATURE SURVEY

<table>
<thead>
<tr>
<th>Sr.No.</th>
<th>Paper</th>
<th>Author</th>
<th>Year</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.</td>
<td>MVC Design Pattern for the multi framework distributed applications using XML, spring and struts framework</td>
<td>Praveen Gupta</td>
<td>2010</td>
<td>How to use J2EE and Multi Frame Work and MVC Architecture</td>
</tr>
<tr>
<td>3.</td>
<td>Architecture of E-Commerce Systems Based on J2EE and MVC Pattern</td>
<td>Yanfang Wang, Chunyan Guo, Lei Song</td>
<td>2009</td>
<td>Architecture For E-Banking</td>
</tr>
</tbody>
</table>

Table 1. Literature Survey

VI. EXISTING SYSTEM

Currently we are having lot of banks in the market and any person can do transactions of any individual bank either manually or in online. But no one can do all banks transactions in a single portal or in single bank. The current online banking system is only for individual banks in which the customer has to remember his/her own user name and password for each bank which increase the complexity for the user. This is the main disadvantage in existing system to avoid this problem we are introducing “multi banking transaction system using MVC2 Architecture”.

VII. PROPOSED SYSTEM

The Multi Banking System Interface using MVC2 Architecture is targeted to the future banking solution for the users who is having multiple bank accounts in multiple banks. This interface integrates all existing banks and provides business solutions for both retail and corporate.

This system acts as a standard interface between the clients and all the banks, By using this portal any client who maintain accounts in various banks can directly log on to Multi Banking System Interface and make any kind of transactions. In the backend, system will take care of the entire obligation required in order to carry on transaction smoothly.
VIII. SYSTEM ARCHITECTURE

![System Architecture Diagram](image1)

IX. THREE LEVEL SECURITY

![Three Level Security Diagram](image2)

This unique and user-friendly 3-Level Security System is involving three levels of security. Where the preceding level must be passed in order to proceed to next level.

Security at this level has been imposed by using Text based password (with special characters), which is a usual and now an anachronistic approach.

At this level the security has been imposed using Image based authentication (IBA), where the user will be asked to select from the two difficulty levels. Both the levels will be having three unique Image grids, from where the user has to select three images, one from each grid.

After the successful clearance of the above two levels, the 3-Level Security System will then generate a one-time numeric password that would be valid just for that login session. The authentic user will be informed of this one time password on his signed up email-id.

Any hacker if in the extreme case, suppose (although difficult) will cross through the above two mentioned security levels, will definitely not be able to cross the third security level, unless he has access to the original user’s email-id.

1. Advantages:
   1. This system use only security purpose, it uses to all security place.
   2. Hackers are not very easily to hack the security, Bcoz there levels are more useful this concept.
   3. Any hacker if in the extreme case, suppose (although difficult) will cross through the above two mentioned security levels, will definitely not be able to cross the third security level, unless he has access to the original user’s emailed.
4. The user will be authenticated as an authentic user, and will be awarded access to the stored information, only after crossing the three security levels (Security level1-Text password, Security level2-Image Based password, and Security level3- One-Time Automated password).

X. MVC2 ARCHITECTURE

The MVC1 architecture was able to solve some of the problem of the web and internet programming but still there were a lot of things missing from it. It was centered on the navigation of the JSP [3] pages so there was the scope of the further development in the architecture point of view. During this process the next development is MVC2. We were trying to solve this problem using the Servlet and JSP together. The Serves handles the Initial request and partially process the data. It set up the beans then forward the result to the one of the jsp page. The Servlet decide the one of the page to be displayed from the list of pages.

The JSP Pages were used to Presentation purpose only. The Business logic has been removed from the page. This makes the pages easier to represent and light weight pages which were easy to display on the internet.

In this model all Control and application logic were handled by the Servlet. The Servlet was written in the java programming language. So it was also easier to handle the programming part of the Servlet. In this scenario the Servest becomes the power full for the complete application and It has emerged as the center point for the application.

In a MVC 2 application, requests from the client browser are passed to the controller. The controller performs any logic necessary to obtain the correct content for display. It then places the content in the request (commonly in the form of a JavaBean or POJO) and decides which view it will pass the request to. The view then renders the content passed by the controller. MVC 2 is recommended for medium- and large-sized applications.

XI. CONCLUSION

Multi Banking Transaction Interface Using MVC2 Architecture is used to provide a common interface for Multiple Accounts. In this system, Security and all other services are being kept in acknowledged to serve the Customer. Future Implementation for any middleware service is easy by using MVC2 Architecture.

ACKNOWLEDGMENT

We would like to thank our college management and staff for extending their continuous support for us to carry out this part of work successfully.

REFERENCES

[3] JSP 1.1 Syntax Reference