



INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X

Impact factor: 4.295

(Volume 5, Issue 2)

Available online at: www.ijariit.com

Multibanking system

Sudharshana Rajendran

sucha.rajendran@gmail.com

Velammal Engineering College,
Chennai, Tamil Nadu

Harshavartini D.

harshavartini@gmail.com

Velammal Engineering College,
Chennai, Tamil Nadu

P. S. Smitha

smithaps.ap@gmail.com

Velammal Engineering College,
Chennai, Tamil Nadu

ABSTRACT

MultiBanking System is a web application on which the end users perform various bank accounts transactions on a single platform. This idea paves a way for improving future banking domain. In general, users first log into different online bank sites to perform transactions which have many security problems and takes more time. But this website has all characteristics that are available in every bank. Transactions are maintained by admin and admin provides user support for customers. This system will act as an intermediate between banks and users. Users can maintain a single username and password and can make a transaction in many bank accounts and contact only one customer care service for any problems. The main feature here is the dynamic service composition interface which is used for transferring money from all the secondary bank accounts to the primary account and later withdrawing the money from the primary account. If there are 'n' banks for a user, then one of the banks act as the primary account and others play the role as secondary accounts. When the user's withdrawal amount is more than the amount in his/her primary account balance and no banks have the required amount, the system checks whether 60% of the amount is available in any of the secondary banks. If so, it takes the amount from the respective bank. Otherwise, it checks for 50% of the amount and so on until a specified limit is reached. The residual amount is obtained in the same manner and the amount is transferred to the primary account. At last, the user can withdraw the required amount from the primary account.

Keywords—Multibanking, Single interface, Integrated system, Dynamic service composition, Service-oriented architecture, Service Composition

1. INTRODUCTION

Service Oriented Architecture is one of the highlights of distributed computing. It describes the web system by request-response design method for synchronous and asynchronous web applications. SOA is termed as coordination of services and communication within one another. The communication involved can be either the coordination between two or more services or the just information passing. The Service Oriented Architecture focuses on business services. Creating business

solutions for these services require a combination of many existing services. These services play a vital role in Service-Oriented Systems thereby paving a way for creating newer solutions.

2. SERVICE ORIENTED ARCHITECTURE

Service is a self-contained, well-defined function, with functional and non-functional requirements, that is independent of other services. The services here are called stateless. They make the environment efficient. SOA is chosen as the enterprises must reply to the business changes to meet the newer business requirements. It is loosely coupled and it allows adding newer services in a better manner. Figure 1 depicts the Service Oriented Architecture.

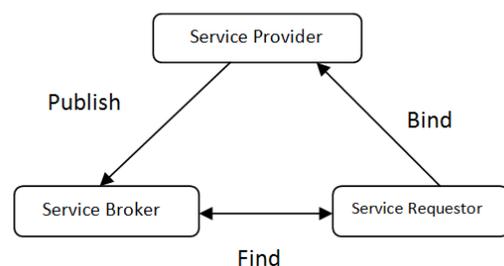


Fig. 1: Service Oriented Architecture

- Service-oriented architecture allows software devices to communicate with the applications. The main components of SOA are given below.
- Service Requestor: The service requestor will request the service from other components by sending a request to the service provider.
- Service Broker: Service broker also called as the service contractor gives permission to the service requestor to identify services provided by the service provider.
- Service Provider: Service provider provides the services to the service consumer and it broadcasts its contract to the service registry and makes itself available to the service requestors.
- Service Registry: It accepts and stores the service contracts from the service providers and makes them visible to the service requestors.

2.1 Characteristics of SOA

The characteristics of service-oriented architecture are given below.

- Loosely couple
- Vendor diverse
- Flexible
- Interoperability
- Agile development
- Stateless
- Reusable
- Composition

There are many other characteristics of SOA and the essential attribute for the service-oriented interface is the service composition. For this service composition, service interoperability is needed. Here we highlight the role of service composition.

3. SERVICE COMPOSITION

Service composition is a collection of services that are made to automate a specific task or business process. For the composition of services, there must be at least two services taking part and one initiator which initiate the service composition.

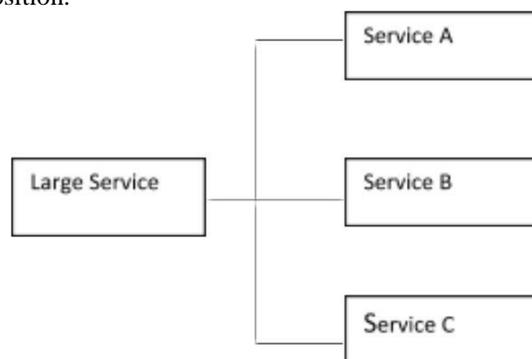


Fig. 2: Service Composition

Figure 2 depicts the composition of small services A, B, and C respectively into a large service.

The services communicate with one another via a network because inter-service communication is too slow when compared to inter-component communication within the same application.

Service composition can be classified as primitive and complex variations. The primitive composition is similar to the point-to-point exchanges and as technology grew complex compositions became famous. In order to be reusable, the services need to be platform independent. This paves way for the enterprises to establish a business process by composing the existing web services. Web Service Description Language (WSDL) provides a description of the services.

Service composition has the following advantages:

- Reduces the development of code for service composition.
- It provides a simple way to reuse the available services.
- Lowers duplicity.
- Availability of single place changes in functionality over the enterprise to get the desired functionality.

In this paper, we have used service composition to integrate multiple bank services into a single service. The entire banking transactions are made available in a single webpage. The client can perform transactions such as withdrawal, depositing, check

the balance and check their transaction details using a single username and password.

4. EXISTING SYSTEM AND ISSUES

The existing system allows the user to log into many banks with many security components like smart card, debit cards, credit cards, username, passwords, and certain biometrics. This allows the user to perform transactions in online banking sites. The existing system consists of master servers like servers from multiple banks, servers for authenticating the end users, huge databases, mobile service provider and mobile devices. The authentication server allows the verification of the user and provides the services to the user. If there is no authentication, some illegal access can occur. Here different users from different ends are allowed to communicate with the authentication server. This server plays a vital role in authenticating the users. The data in the database server can be accessed only if the username and card information matches. This matching is done by using various algorithms and programs. Once the authentication and verification are done, the data is accessed from the database server else the database access will be declined. The data in the database server consists of users' biometric data, username and other information of the end user which allows the user to make a transaction.

5. PROPOSED SYSTEM

There is a greater number of researches going on the online banking system. Every bank is now offering online banking service options to all the users but if a user and his family have accounts in the same bank then remembering every username and password for each account is a complex task for the end user. In this process, we have introduced a multi-banking system model where the user can maintain many accounts and can perform all the transactions between the various accounts using a single username and password. Here the tedious task of remembering all the username and passwords for all the available accounts can be avoided. This paves way for future banking solutions for the user who are having multiple bank accounts. They can use this interface to access money from different bank accounts and transfer the amount among the multiple bank accounts. Everything is done by logging into the integrated system using a single username and password. The user can also view the transaction that he has made using the site.

The main feature is that, if the user wants to withdraw an amount which is greater than all the balances in all his/her accounts, the system uses dynamic service interface composition. Here, consider if the user has 'n' number of bank accounts and one of the banks accounts acts as a primary account. If the user requires an amount which is greater than the amount in the primary account and no other bank has the full required amount, the system checks whether 60% of the amount that is needed is available in any of the secondary banks. Then, it takes the amount from the respective bank otherwise it checks for 50% of the money and it goes on up to a particular limit. The remaining amount is taken from the banks similarly. The multi-banking system transfers the amount from the secondary banks to the primary bank account. At last, the user will be able to withdraw the amount from the primary account.

5.1 Steps in the proposed system

Step 1: The user will create an account by registering their personal details and then they are allowed to select the multiple banks where they can perform their transactions.

Step 2: The admin will generate the user id and password for the user.

Step 3: Every time when the user logs in, the admin will generate a random verification code either to the users' mail id or to their mobile.

Step 4: By using the users' id, password and verification code, the user is authenticated to access the multi-banking system.

Step 5: The user is allowed to perform the following tasks.

- Bank transactions
- Account Details
- Transaction history

Step 6: The account details will allow the user to view the entire bank balance and the details of his account.

Step 7: Transaction process involves the user to perform transferring the money from one account to another account, withdraw and deposit the amount into the desired account.

Step 8: In transaction history, the user can view the history of all the transactions that he has made using the system.

6. IMPLEMENTATION AND OBSERVATION

The proposed system has two stages. The aim of the project which uses dynamic service composition is done using two phases.

6.1 Registration stage

In this stage, there are two processes. First, the client registration process takes place and then the administration process takes place. In the first stage, the user registers him and selects the required banks. The main feature of the registration phase is to add the user to the integrated banking system. Here, the user registers him by giving his first name, last name, email id, address, mobile number, bank1, bank2, and other bank details, and select the primary account from all the accounts he has given. The user will send a request to the admin.

The admin will check the information of the user and sends the user id and password for the corresponding user through email. Finally, the user id and password will be stored in the database. For each time the user logs in to the integrated system, the user will get a verification id to the registered email id.

6.2 Integration Stage

In this stage, the users' multiple bank accounts are integrated. The user is allowed to do the transaction processes which include transferring the amount, viewing the transactions made by him/her, withdrawal of amount, depositing and also viewing the account details. The manager can also log into the integrated phase and can also view the transaction details.

Using a single username and password, the user can do all the above transaction processes. The services provided for the user are:

- Withdrawal
- Transaction Details
- Money transfer between various accounts
- Deposit
- Manager features

6.2.1 Withdrawal: The user logs into the multi-banking system using his/her login credentials and does the withdrawal process. The user first selects the option from the options that are available in the user home page. Then he gives the required details about the account and enters the amount to be withdrawn. The amount is deducted from the users' balance and updated in the database.



Fig. 3: (a) Shows the UI of withdrawal operation

| Branch | Username | Password | Amount | Date |
|---------|-------------|----------|--------|------------|
| T Nagar | 14222140999 | 9d7965 | 500000 | 2019-04-12 |
| T Nagar | 14222140999 | 9d7965 | 500000 | 2019-04-12 |
| T Nagar | 14222140999 | 9d7965 | 500000 | 2019-04-12 |
| T Nagar | 14222140999 | 9d7965 | 500000 | 2019-04-12 |

Fig. 3: (b) Depicts the withdrawal process

6.2.2 Transaction Details: The user using his username and password credentials logs into the multi-banking site. The end user then enters his account number and fetches his details about the transactions he made in his account. His account number acts as the primary key.

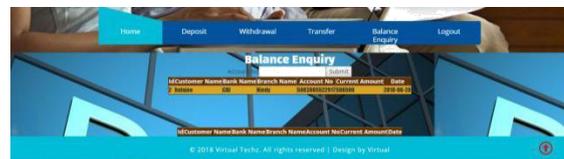


Fig. 4: Depicts the account details of the user

6.2.3 Money transfer between various accounts: This feature allows the user to transfer the amount within multiple banks which he mentioned during the registration process. For transferring the amount, first, the user enters the amount to be transferred and enters the account number of the other bank, to which he wants to transfer the amount. Before all these, the system checks the balance in the current account and whether all the details are given are correct. If the entered details are correct, then the transferring is done successfully.

The main implementation is done here, that is if the amount that is entered is greater than the amount in the primary account, and then the amount is checked in the secondary banks. Then the amounts are taken from the secondary banks and it is transferred to the primary account from which the user can withdraw or transfer it to some other account.

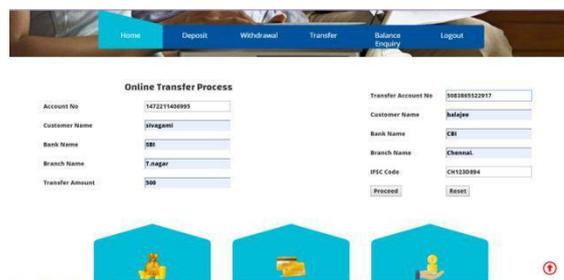


Fig. 5: (a) shows the UI of transferring the amount

| Username | Branch | Amount | Date |
|-------------|---------|--------|------------|
| 14222140999 | T Nagar | 500 | 2019-04-12 |
| 14222140999 | T Nagar | 500 | 2019-04-12 |
| 14222140999 | T Nagar | 500 | 2019-04-12 |
| 14222140999 | T Nagar | 500 | 2019-04-12 |

Fig. 5: (b) shows the database view which depicts successful transferring

6.2.4 Deposit: The user can also deposit the amount into his desired account by entering the amount and the account details and the database is updated with the amount. There are reset operations provided so that the user is flexible to undo the actions he performs.



Fig. 6: (a) shows the UI of deposit feature

| Branch | Customer Name | Password | Amount |
|-------------|---------------|----------|--------|
| T Nagar | 14721140095 | 9k3k95 | 500 |
| KVdy | 588388522917 | y0a9e94 | 50000 |
| Chidambaram | 416654 | 416654 | 5000 |
| Manabachari | 51222721659 | Bk9k82 | 5000 |
| T Nagar | 23796262189 | 9ff943 | 5000 |

Fig. 6: (b) shows the database updating of deposit operation

6.2.5 Manager Operations: The additional feature in this multi-banking system is that the managers of the banks are allowed to log in by creating an account and can view the details of the customer's accounts. He/she can also have the details of last account transactions made in their bank accounts.



7. CONCLUSION AND FUTURE WORK

This system provides a solution to withdraw and transfer the amount between various accounts a user has even when the amount is lesser than his account from which he initiates the transaction. This also allows the user to add banks additional to his existing accounts using composite services dynamically. This can be used in different domains like Internet Banking, billing etc., where the user finds difficulty in security

perspective related to responsiveness and databases. This proposed system allows the user to do all the transactions using a single username and password. In future, by following this dynamic service composition, security and interoperability characteristics of service-oriented architecture can be focused in a number of applications.

8. REFERENCES

- [1] Maurice HTerBeeck, Antonio Bucchiarone and Stefania Gnesi, "A Survey on Service Composition Approaches: From Industrial Standards to Formal Methods" 2008.
- [2] Document Flow Model Handbook, "A Formal Notion for Modeling Asynchronous Web Services Composition".
- [3] Schahram Dustdar, and Wolfgang Schreiner, "A survey on web services composition and Distributed Systems Group, Vienna University of Technology".
- [4] E. Saravanakumar and Anupriya Mohan, "Single Password Multiple Accounts, International Conference on Computing, Communication and Networking", ICCCN 2008.
- [5] PonHarshavardhan, lAkilandeswari and M.Manjari, "Dynamic Web Service Composition Problems and Solution", 2010.
- [6] Stephan Kassel, "An Architectural Approach forService Interoperability", International Conference on Interoperability for Enterprise Software and Applications, 2009.
- [7] Rania Khalaf, Nirmal Mukhi and Sanjiva Weerawarana, "Service-Oriented Composition in BPEL4WS".
- [8] World Wide Web Consortium W3C. Latest SOAP versions, 2003.
- [9] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web Services Description Language WSDL", 2001.
- [10] Clark, "XML Path Language XPath (Version 1.0)", 1999.
- [11] Erik Christensen, Francisco Curbera, Greg Meredith, and Sanjiva Weerawarana, "Web Services Description Language (WSDL) 1.1".
- [12] Sahana K. Bhosale, "Architecture of a Single Sign-on (SSO) for Internet Banking", Proc of International conference on wireless and multimedia networks, 2008.