

An EBA-Web based Application for Banking Information System

Khaldoun Mohammad Batiha

Business Networking & Systems Management, Philadelphia University, Amman-Jordan

Abstract: In the current wake of growing technology and modernization the biggest challenge remains the optimization of our time, particularly for the business executives. For the executives the vital information at their fingertips plays an important role. In the computer era, there are many financial institutions like wise banks which provide web-based application to access all their functionalities associated with particular task. An Enterprise Bank Agent (EBA) is developed, which resolves such problems being faced by user, it provides one window for that enables them to experience and enjoy the flavors of banking functionality. The key feature of this application is its ability to handle all registered Customers business requirements. In order to handle all customer services, it shall act as a smart answering machine.

Key words: EBA, agent technology, enterprise application integration (EAI)

INTRODUCTION

Delivering high levels of reliability and availability required of business-to-consumer & business-to-business requires not only great technology, but also great operational processes with concrete security. Building a scalable, highly reliable & secure application is a necessary process for financial sector. The greatest concern of vendor today is to provide his products and services available 24 hours a day, seven days a week. Enterprise Bank Agent (EBA) is a secure business component, which provides a business tunnel among different financial institution. It provides solution to all registered banks to perform there cross banking business activities. The EBA provide component base solution, which easily integrate itself to bank's legacy system. Its ultimate goal is to facilitate the end customer and provide different Services.

Today most people are carrying online transactions in their regular daily life. This is typically called consumer-to-business (C-to-B) web-based application^[1]. There are no doubt that sites like these are gaining critical market space around the globe and will continue to grow. Web-based application also includes the business-to-business (B-to-B) market space, which accounts for a significant amount of activity on the Internet. Just think of all the supply chain purchasing that takes place to manufacture and support many of the products and services which are being used daily. Examples of B-to-B e-commerce include wholesale companies selling to end retailers, e.g., a PC manufacturer selling to distributors and large retailers^[2].

Until recently, e-commerce was feasible only for large companies. The Internet and the World Wide Web made it possible for even small businesses to compete

with large companies^[3]. E-commerce allows the companies to conduct business 24 hours a day, 7 days a week worldwide. As B-to-B commerce grows, businesses will come to rely on this type of e-commerce as an everyday business solution. Very soon the Internet will be a standard place for business to communicate with other businesses.

Agent technologies: The primary objective behind EBA (Enterprise Bank Agent) is to facilitate the customers with complete features of banking no matter where ever they are, what ever the time is and so on, to enhance the users to get benefit of complete services of banking virtually on the internet without any charges. EBA gives the users a perfect virtual banking service to ensure that they are aware of what they can and can not do.

EBA as a banking agent provides the following objectives to its users:

- * Account Balance
- * Account Summary
- * Account details.
- * Account Mini Statement.
- * Account Statement by period.
- * Schedule Funds Transfer.
- * Transfer Funds.
- * Funds Low Alert.
- * Bills Payment for different utilities.

As EBA is web based application and the main objective and proposal behind EBA creation was that, there should be a system of enables all banking users to have an online facility to do banking no matter what types of bank account they have, EBA provides the ability to connect multiple banks to one agent which allows the bank users to get connected and obtain updated information all the time, information such as

transaction summary, bank statements, bill payments, funds transfer. There is no need to stuck in big queue outside the banks to pay the bills payments, its EBA which came with this facility to pay almost all the utility bills online through EBA bill payments facility, as EBA is an agent with a heterogeneous environment so its also has a beauty of EBA that users can transfer funds from one bank account to another quite easily with any documentation or approvals that is why an EBA type of thing was required in the market which facilitates the users such kind of facilities which are discussed above.

So EBA came with all these solutions to give its users a complete taste of virtual banking in today's world of globalization and technology so they can move up and save their time and can grow them as modern and develop countries do.

EBA architecture: EBA architecture designed by focusing the current state of technologies and cutting age methodologies which includes the number of technologies to come up with the crucial needs of business for instance from load balancing to 128 bit encryption.

The architecture of EBA encapsulates two architecture too, one from client side, which allow the in and out data on https protocol and on other side it communicate the Remote Method call (RMI) communication with other the legacy system application of financial institutions.

The Enterprise Bank Agent (EBA) itself logically divided into three parts, as shown in Fig.1. one is application server, it listen http request from browser based client to call the respective Java Server Pages (JSP) located locally and also invoke the methods on business logic server maintain the business process logic via business objects.

The third and last component of Enterprise Bank Agent (EBA) is database, it store the general purpose of information of registered customer. The components can be replaced or altered easily without affecting the adjacent one. So the data storage which used on EBA architecture RDBMS.

Software customization: This section discusses the customization factors of Enterprise Bank Agent (EBA) and provides an insight into software architecture. For EBA; software customization can be divided into three different levels, as shown in Fig. 2.

- * Web Tier
- * Business Tier
- * Database Tier

Figure 2 shows the different tiers of the EBA i.e. the physical and/or logically tiers.

Web tier customization: The framework is based on well known(MVC) 2 architecture and the primary goal of this architecture is to separate the model (the application state) from the components

which executes the *application* logic and renders presentation thus providing a system which is less prone to changes and additions. Furthermore, the entire configuration is modeled in XML files which mean that any current or future changes in configuration would not require any recompilation.

Business tier customization: The most common mistake that always creeps in at design time is to mix the application logic with business logic. Let me give an example, an application XYZ is developed as desktop application and now the management requires this to be available to internet users too, if developers have separated the application logic from business logic then they can easily take business logic tier out of desktop application and with some minor changes the whole business logic can be reused by a newly developed web tier.

The entire business logic of EBA is implemented in JavaBeans, which can be utilized anywhere and can be modified without affecting the components of web tier thus resulting in a maximum customizability and manageability feature.

It can be seen from Fig. 2 Vertical Partitioning; the business logic is completely separate logically from the components of web tier.

Database tier customization: Changes at this level are usually not advised and often not required, but at the same time customers can change and/or add requirements as and when they think business requires some changes to this tier. These changes always affect the business tier because business logic components access database and any structural change in metadata can introduce incompatibilities in the data structure.

EBA usually takes into considerations the effect of these radical changes during design and for this reason a special layer called Data Access layer is introduced as seen in Fig. 2.

Security: This section deals with security issues of the distributed system, in this type of system data-protection and access control plays an important part. A simplest form of authentication, authorization and encryption are usually guaranteed by https and Secure Socket Layer (SSL).

Transactions: In this case high availability in spite of fast processing transaction mechanisms have to be used. Standardized interfaces are also used. The standard interfaces are divides into two-phase-commit and offers new concepts of nested transactions.

Technical infrastructure: There are currently a number of strategies available which helps to realize a 3-tier architecture. For example the distributed object systems, with RMI is considered to be the best and or

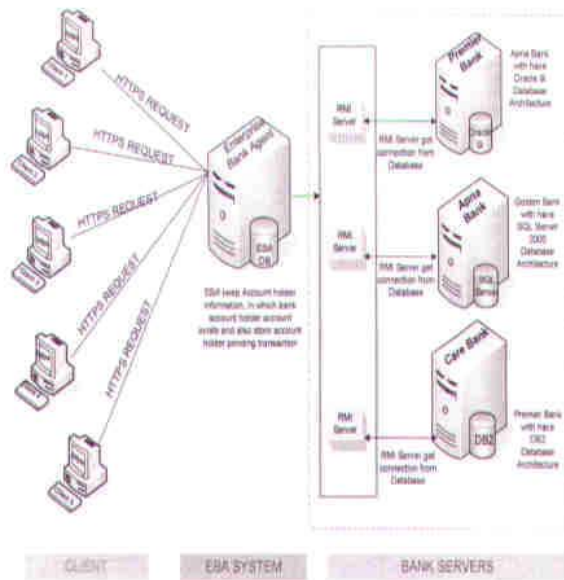


Fig. 1: EBA Architecture

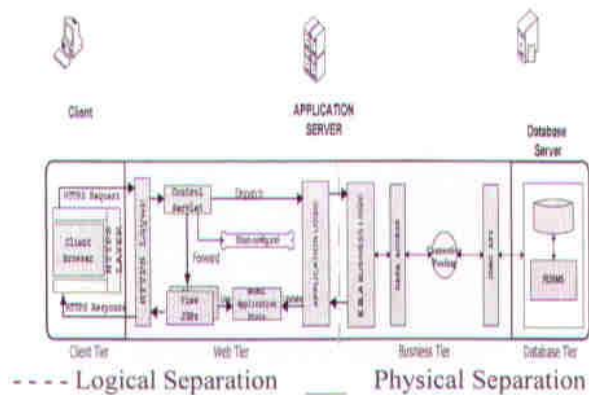


Fig. 2: Vertical Partitioning

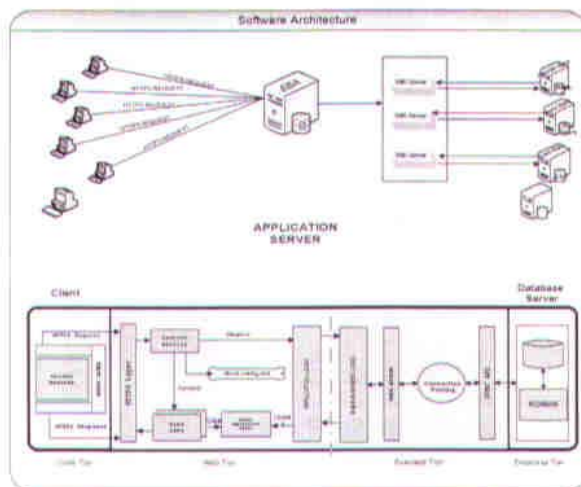


Fig. 3: Internal software architecture

the most efficient concept currently available. It connects the object oriented paradigm to tried and trusted communication technologies like TCP/IP. In

addition to basic functions of RMI offers a multitude of useful and complementing services.

EBA Database Architecture: This section introduces the relevant database schema of a suitable EBA database. Class Diagram (Specification View) provides the mapping which realizes the design of the database this is implemented using object oriented design methodologies.

PK and FK notations are used for Primary Key and foreign Key attributes. The design is normalized up to third normal form (3NF).

Relational Database Schema

USER_REGISTRATION: User Registration is the main and primary entity through out the Enterprise Bank Agent (EBA) User_id is treated as primary key in this entity and Cust_No as foreign key from User_Profile entity.

USER_PASSWORD: User_Password entity maintains Pass_ID as primary key and password, User_ID as Foreign Key. Login will be verified from this entity.

CONFIDENTIAL_DETAILS: Confidential_Details entity maintains Ser_No as Primary Key along with Mother_Maiden_Name, Secret_Code and Cust_No as Foreign Key. The purpose of this entity is to maintain the security by means of Secret_Code value.

TRANSACTION: Transaction this entity maintains the Transaction_No as Primary Key besides Date_Of_Trans, Source_Acc, Beneficiary_Acc, Amount, Remarks. There is two Foreign Keys User_ID from User_Registration entity and TT_ID from Tran_Type entity. The purpose is to maintain the all ATM as well as Manual Transactions.

TRAN_TYPE: Tran_Type entity maintains two columns one is TT_ID as Primary Key and DESCRIPTION. The purpose is to check the transaction type whether it is Credit or Debit.

USER_PROFILE: User_Profile entity maintains all Bio-data of the registered customer along with Cust_No as Primary Key.

ACC_INFO: Acc_Info entity maintains the Acc_No, Bank name, BRANCH name, The Nature of the Accounts, The title of Account, ATM card number, ACCOUNT_INFO_ID as Primary Key and CUSTOMER_NUMBER as Foreign Key.

BENEFICIARYACC: BeneficiaryAcc entity defines the beneficiary accounts in order to maintaining the funds transfer along with Account number, Bank Branch as well as currency.

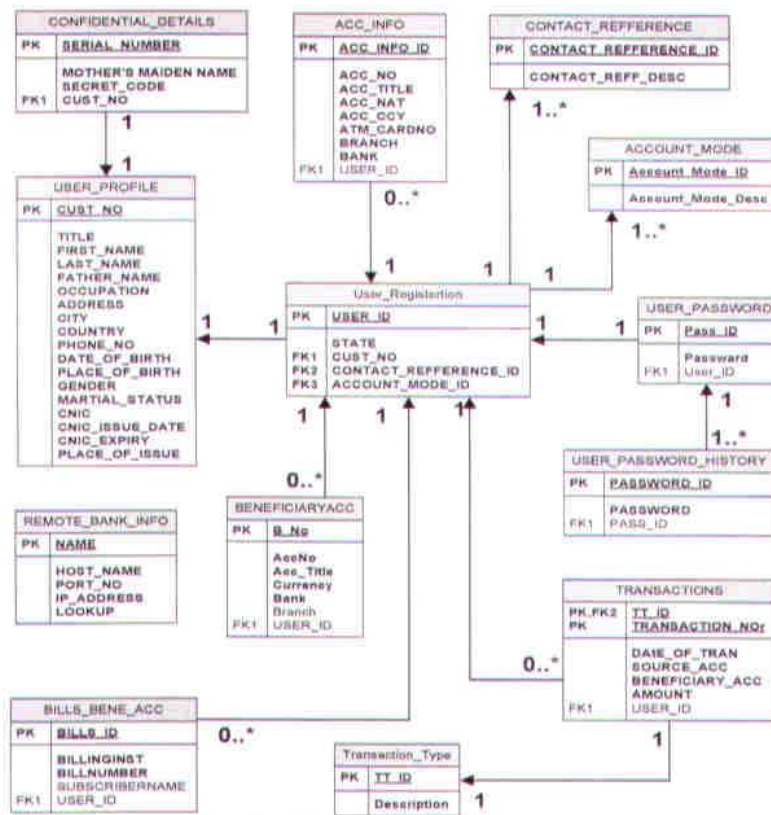


Fig. 4: Relational database schema

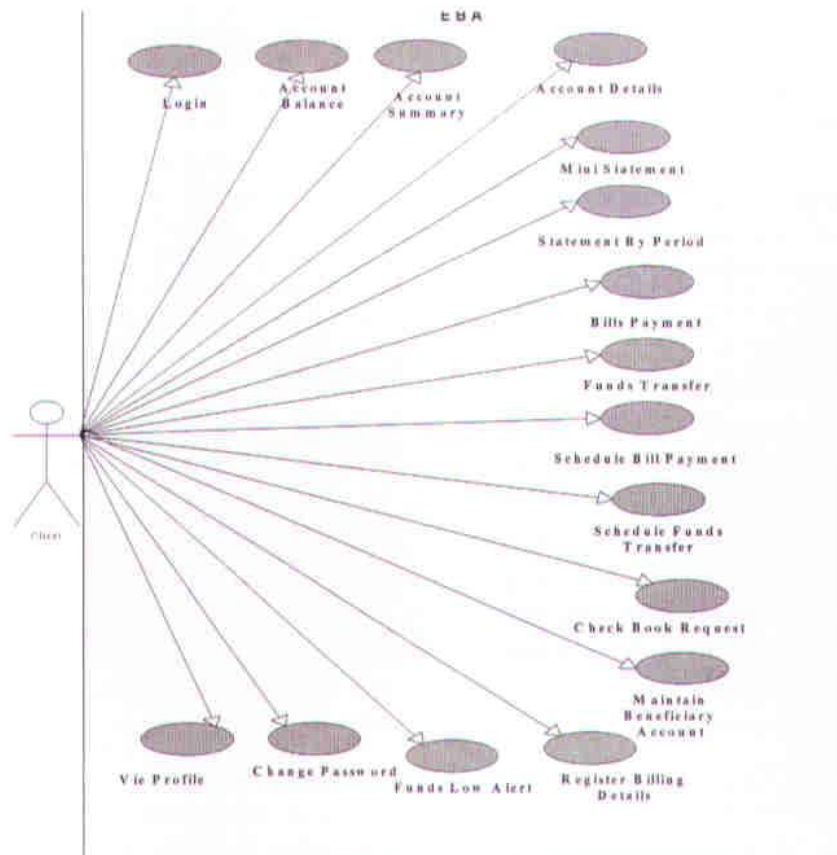


Fig. 5: EBA use case diagram

Object Interaction Diagram

Sequence Diagram: Login User, Method: getUserValidate.java, Comments: Either Exist or Not



Fig. 6: Login sequence diagram

Sequence Diagram of EBA (Register)

Sequence Diagram: Register, Method: getInsertDatabaseBean.java, Comments: Returns The Registration

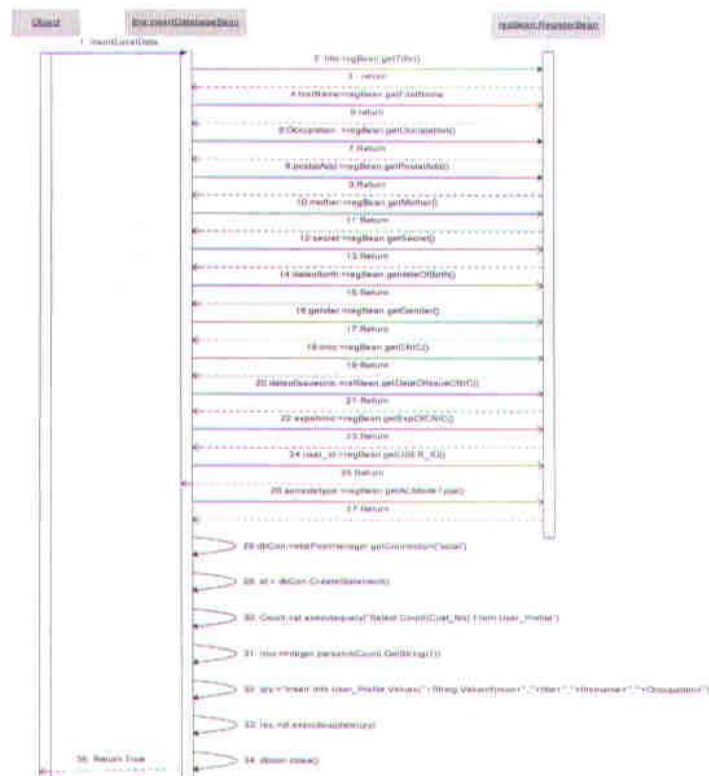


Fig. 7: Register sequence diagram

Sequence Diagram of EBA (Mini Statement)

Sequence Diagram: Mini Statement, Method: getMiniStatement.java, Comments: Returns The Mini Statement

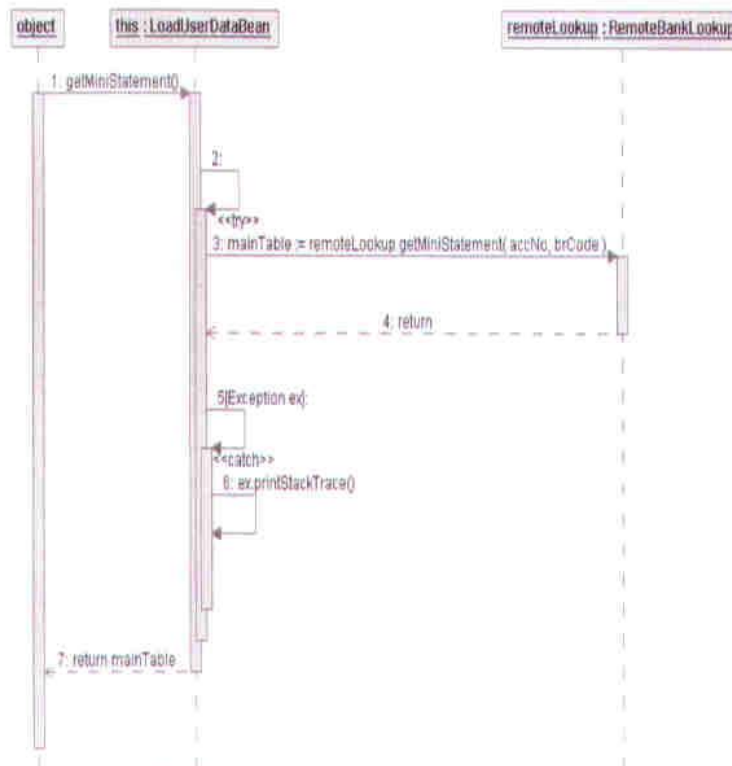


Fig. 8: Mini statement sequence diagram

Sequence of EBA (Statement by Period)

Sequence Diagram: Statement by Period, Method: getStatementByPeriod. Java, Comments: Returns Statement by Period

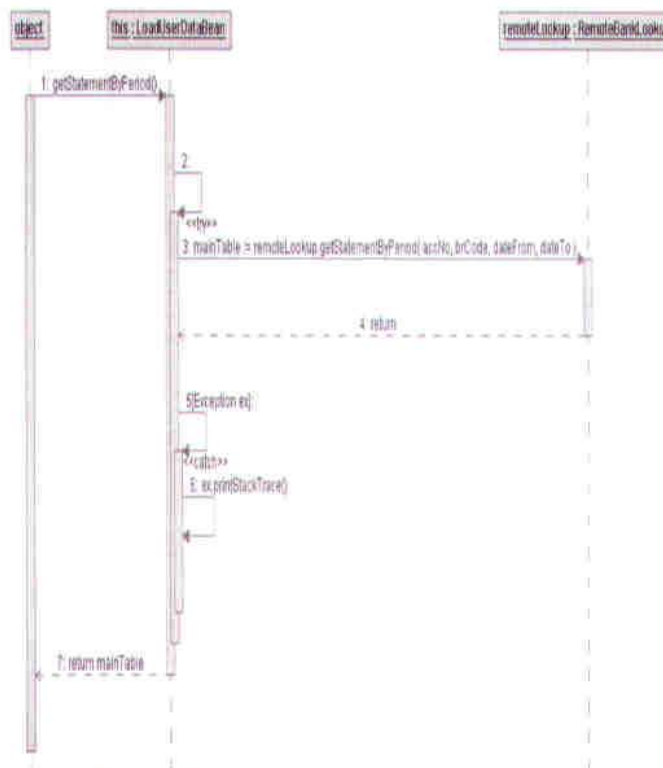


Fig. 9: Statement by period sequence diagram

Sequence Diagram of EBA (Account Balance)

Sequence Diagram: Account Balance, Method: getAccountBalance. Java. Comments: Returns The Balance

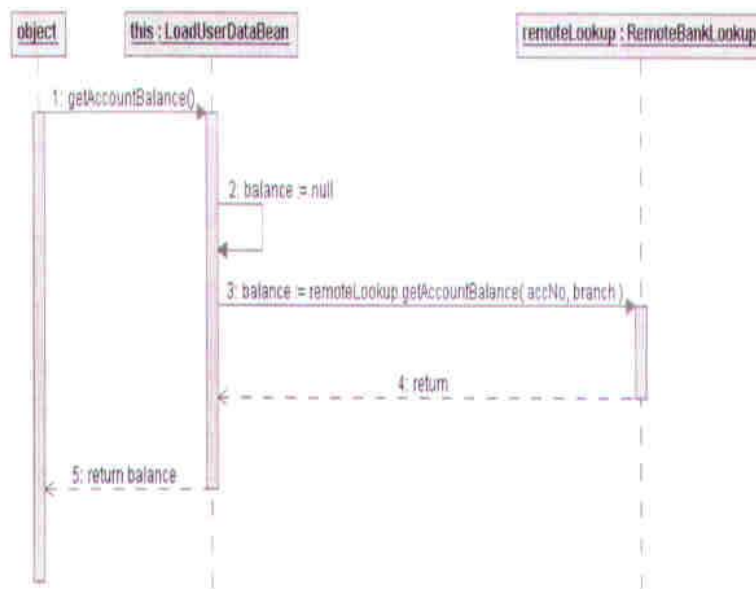


Fig. 10: Account balance sequence diagram

BILLSBENEACC: BillsBeniAcc entity maintains Bills_ID and User_ID as primary key and foreign key respectively. The purpose of this entity is to keep the paid billing history.

ACCOUNT_MODE: Account_Mode entity keeps the Account_Mode_ID as primary key. The objective of this entity is what kind of mode the user wants to perform on Account.

CONTACT_REFERENCE: Contact_Reference entity has two fields one Contact_Reference_ID as Primary key and other Contact_Ref_Desc. This entity shows the type of contact either SMS or E-mail.

Logical presentation of EBA: The internal logical transaction is provided in this section.

EBA use case diagram: Presents all the services the system will provide.

CONCLUSION

It was shown in this paper that the EBA can provide a suitable solution which combines security with a comprehensive internet banking. It also provides an easy mean of integration with other banks and or similar organizations.

Future work

- * EBA Integration with Server & open Source Technology i.e Linux
- * Dynamically Connect Different Bank Server host and Port Nos. (in case there host name and port are changed)
- * EBA Server sends and receives data in the form XML based Query.
- * Completely handle each exception of different EBA Option.
- * Adding java script.
- * Make a SMS Component which informs customer on his / her mobile when his / her account is update.
- * Make EBA more intelligence to have ability to add new banks and Utility bill servers.
- * More Secure & Robust implementation of SSL

REFERENCES

1. Attaran, M., 2000. Managing legal liability of the Net: A ten steps guide for IT managers. Information Management & Computer Security.
2. Farhoomand, A.F., V.K. Tuunainen and L.W. Yee, 2000. Barriers to global electronic commerce: A cross-country study of Hong-Kong and Finland. J. Organizational Computing and Electronic Commerce, 10: 23-48.
3. Larsen, T.R.K. and A.P. Bloniarz, 2000. A cost and performance model for Web service investment. Communications of the ACM, 43: 109-116.