

# Distributed Online Banking

**Submitted by: Mahmood Akhtar**

University of Illinois at Springfield  
2356 William Maxwell Lane Apt # 403,  
Springfield, Illinois-62703

Day Time: (217)206-6190[Office] / (217)206-1413 [Home], (217)638-4221[cell]  
[makht1@uis.edu](mailto:makht1@uis.edu)

**Advisor: Kamyar Dezhgosa**

University of Illinois at Springfield  
One University Plaza, MS HSB137  
Springfield, IL 62703-5407

Day time: (217) 206-7243[Office]  
[kdezh1@uis.edu](mailto:kdezh1@uis.edu)

## Abstract

The purpose of this project is to develop an on-line banking system that provides customers with the facility to check their accounts and do transactions on-line. The system will provide all the banks facilities to its customers when their authentications [user id and password] match, including viewing account information, performing transfers, giving the customer an option of changing address, paying bills on-line, password retrieval, performing transactions, viewing transactions and the locations the bank and its branches. The system should also support an online enrollment facility for credit card customers and should allow customers to view their personnel accounts and to pay bills online from their account.

The system should assign a unique transaction number to every transaction that a user makes. It should also generate credit card numbers automatically when administrator, insert credit card customer's information. The Administrator will administer both normal bank account and credit card bank accounts. The administrator should have the ability to perform various operations like creating a normal bank account for the customer and performing functions like transfers, withdrawals and deposits when the customers want teller transactions. The administrator also has the privilege to close the customer's account on the request of the bank customer. The customer should be able to access his/her account from anywhere just by inputting the correct user-id and password. All this process will take place on a secure channel using SSL technology.

**Key Words:** API (Application Programming Interface), JDBC (Java Database Connectivity), SSL (Secure Socket Layer), ODBC (Open Database Connectivity), JSP (Java Server Pages)

## Introduction

In today's world of emerging technologies, enterprises are moving towards the Internet for businesses. People are rushing towards the e-commerce applications for their day-to-day needs, which in turn are making the Internet very popular. Online Banking has given both an opportunity and a challenge to traditional banking. In the fast growing world, banking is a necessity, which in turn takes a lot of time from our busy schedule. Going to a branch or ATM or paying bills by paper check and mailing them out, and balancing checkbooks are all time-consuming tasks. Banking online automates many of these processes, saving time and money. For all banks, online banking is a powerful tool to gain new customers while it helps to eliminate costly paper handling and manual teller interactions in an increasingly competitive banking environment. Banks have spent generations gaining trust of their customers, and the goal for this project is to develop a user friendly, secure Online Banking Application. The application will be built using Java Server Pages (JSP), tomcat as the application server, and Microsoft Access / SQL Server as a database.

### *Architecture of Online Banking:*

The Online Banking Application is based on 3-tiered model. The Enterprise architecture for Online Banking Application is shown below.

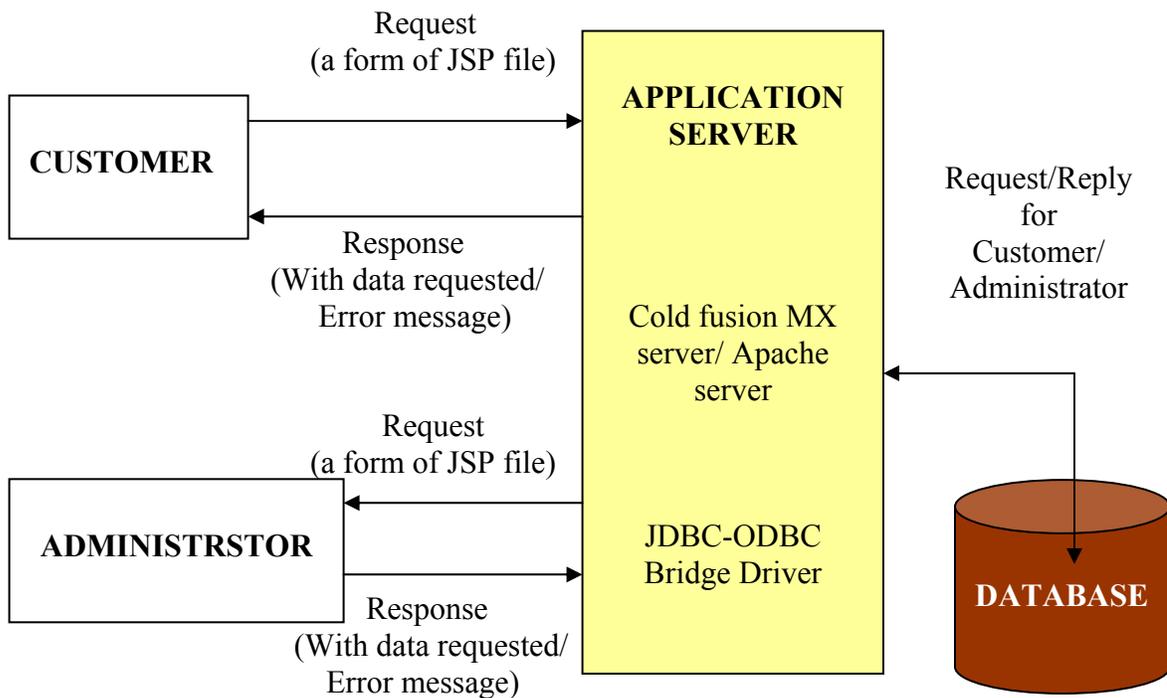


Figure 1: Architecture of Online Banking Application

The 3-tiered architecture shown above has the following major components:

1. **Client:** There will be two clients for the application. One will be a web-based user-friendly client called bank customers. The other will be for administration purposes.
2. **Application Server:** It takes care of the server script, takes care of JDBC-ODBC driver, and checks for the ODBC connectivity for mapping to the database in order to fulfill client and administrator's request.
3. **Database:** Database Servers will stores customer's and bank data.

Simply stated, the application works based on a request/response protocol. A client initiates a request to the server. The server responds by executing the business logic hosted inside the JSP program and if required, communicates with the Database Server to fulfill a client's request.

The Online banking Application project will be divided into 4 modules namely:

1. Bank Account
2. Bank Account Administrator
3. Credit Card Customer
4. Credit Card Account Administrator

Each module is discussed in more detail under design phase.

## **Background and Problems**

### **Similar banking applications available in the market**

A considerable amount of research has been done in the past few months on this project. Many banks had migrated from paper based banking system to electronic / online banking. Each bank had its own, user friendly interface, which helps its customers to interact with their account at their ease. A wide variety of online banking applications are available in the market, which in turn help the bank to function smoothly without reducing the quality of service. All banks which are using online banking application use the same basic principle.

*National City Bank* has excellent features, which allows customers to check their accounts and view their statements. The best thing about this bank's system allows us to schedule payments and do online transactions. The security feature is the best; it gains the trust of the customer and allows them to do their transitions in an efficient and secure manner. However, the interface is very complicated for novice users. The interface for credit card customer is confusing [1]

*Bank One* is the fastest growing bank in United States with millions of customers, who perform their transactions online. The security issues are wonderful, and it allows the customer to view their transactions, pay bills online, ATM/branch locator and provide

calculators and educators that will help customers to determine savings, mortgages and loan amounts. Educators are learning materials covering several financial topics which help the customer to learn more about the facilities that the bank has for them. The user interface is not as useful for the novice customers [2]

**TCF bank** Online banking is a safe, fast and convenient way to access the accounts. It has all functionality but was lacking in user interface [3]

### **Comparison between JSP and other technologies**

- **JSP vs. Active Server Pages (ASP).** ASP is a similar technology from Microsoft. The advantages of JSP are Saturday, March 13, 2004two fold. First, the dynamic part is written in Java, not Visual Basic or other MS-specific language, so it is more powerful and easier to use. Second, it is portable to other operating systems and non-Microsoft Web servers.
- **JSP vs. Pure Servlets.** JSP doesn't give you anything that you couldn't in principle do with a servlet. But it is more convenient to write and to modify regular HTML than to have a million println statements that generate the HTML, by separating the look from the content, you can put different people on different tasks. Your Web page design experts can build the HTML, leaving places for your servlet programmers to insert the dynamic content.
- **JSP vs. JavaScript.** JavaScript can generate HTML dynamically on the client. This is a useful capability, but only handles situations where the dynamic information is based on the client's environment. With the exception of cookies, HTTP and form submission data is not available to JavaScript. And, since it runs on the client, JavaScript can't access server-side resources like databases, catalogs and pricing information, to understand the distinction between Java Script and Java Server Pages. *Java Script code is generally executed by the web client (browser) after the web server sends the HTTP response. Java Server Pages are executed by the web server before the web server sends the HTTP response.* In fact, JSP is what creates the HTTP response. Thus, Java Script is said to be a “Client Side” technology. Its underlying code can be viewed by the web user, while Java Server Pages are a “Server Side” technology and its underlying code is not exposed to web users. JSP is processed by the web server before result reaches the client. [4]
- **JSP vs. Static HTML.** Regular HTML, of course, cannot contain dynamic information. JSP is so easy and convenient that it is quite feasible to augment HTML pages that only benefit marginally by the insertion of small amounts of dynamic data. Previously, the cost of using dynamic data would prevent its use in all but the most valuable instances.

## Software Development Methodology

### Analysis

The table below lists the functionalities to be included in the Online Banking Application, as well as certain features that will not be supported. This list is a tentative, since it may be discovered during development that additional features are required or some existing features may prove to be unworkable or impractical due to time limitations. The application will be written Java Server Pages (JSP), as it is easier to write and maintain pages. All the necessary hardware and software requirement for the complete application are included in this paper. Table 1 represents the features supported by Online Banking Application.

<b>Feature</b>	<b>Support</b>
Support 3-tier architecture(Client, Server, Database)	<b>yes</b>
Creation of design for all common components	<b>yes</b>
Creation of design for all Server Side Components	<b>yes</b>
Support session tracking	<b>yes</b>
Easy means of navigation through pages containing proper session tracking.	<b>yes</b>
Support Application Servers	<b>Tomcat</b>
Support Secure Socket Layer (SSL)	<b>yes</b>
Easy means of navigation through different pages, which are secured using SSL.	<b>yes</b>
Support the ODBC connections and JDBC-ODBC as well	<b>yes</b>
Database	<b>MS-SQL Server or MS-Access</b>
Establish database connection	<b>yes</b>
Generation of Use case and Data flow diagrams	<b>yes</b>
Programming Language support	<b>JSP</b>
Operating System support	<b>Windows'2000 or higher</b>

Table 1: Functionalities for an Online Banking Application

## **Design**

Dataflow diagrams are depicted below to give the clear understanding of Online Banking Application.

The Online banking Application project will be divided into 4 modules namely:

1. Bank Account
2. Bank Account Administrator
3. Credit Card Customer
4. Credit Card Account Administrator

### ***Module 1***

In this module the customer is allowed to logon to the website and can access his/her account by getting user name and password which will be verified with the server and the database. Once he/she gets verified then they are allowed to view their personal account and perform operations such as change of address, paying bills online, viewing transactions and transferring money into other accounts. The data will be highly secured using Secure Socket Layer (SSL) technology. Once the customer finishes the task the update information instantly gets stored into the database. The customer is then allowed to sign out from his/her account.

### ***Module 2***

In this module the administrator is allowed to log on to the website and can access his/her administrative account by using the user name and password which will then be verified with the database. Once he/she gets verified the administrative interface will be displayed, where the administrator can perform operations for both new customers and existing customers. Administrator will help a new customer in opening their account by taking complete information from them. Administrator provides services like withdrawal, deposit, transfer and deleting customer during the time of closing the account. In this module administrator provides great customer service to the customers who want to do phone banking or teller banking. The data will be highly secured using Secure Socket Layer (SSL) technology. The interface for administrator will be both very user friendly and efficient. The data gets stored in the database instantly when the administrator hits the submit button.

### ***Module 3***

In this module, the customer is allowed to apply for the credit cards (Student, Premium and Basic) or for the loan (home, auto and education). Depending upon their selection the corresponding page will be shown when they will be asked to fill out the form. Depending on the credit rating the customer will be either accepted or rejected. Usually the credit rating will be checked with national credit bureau, which will be internally connected to INS and Social Security office database. Once the customer gets approved

for the credit card or loan, their credit card will be sent to them by postal service. Once they decided to register themselves to an online credit card banking they enroll by using the enroll form, where they will be asked to enter their credit card number, their Social Security Number and user defined password(Numbers only).

To access the account, customers should visit the credit card website and get verified with the database by entering the user name and the password they have created. If the verification is successful then they will be allowed to view their credit card account which will display information about their credit limit and balance. They are also allowed to make online bill payments using their credit card account. As this module contains all-important data like credit card numbers, account user name and password and online bill payments, it needs to be secure, therefore Secure Socket Layer (SSL) will be used. It encrypts the data before it is sent and gets decrypted at the server and vice-versa. This will prevent the hackers to view the data, which is being transferred through any media.

#### ***Module 4***

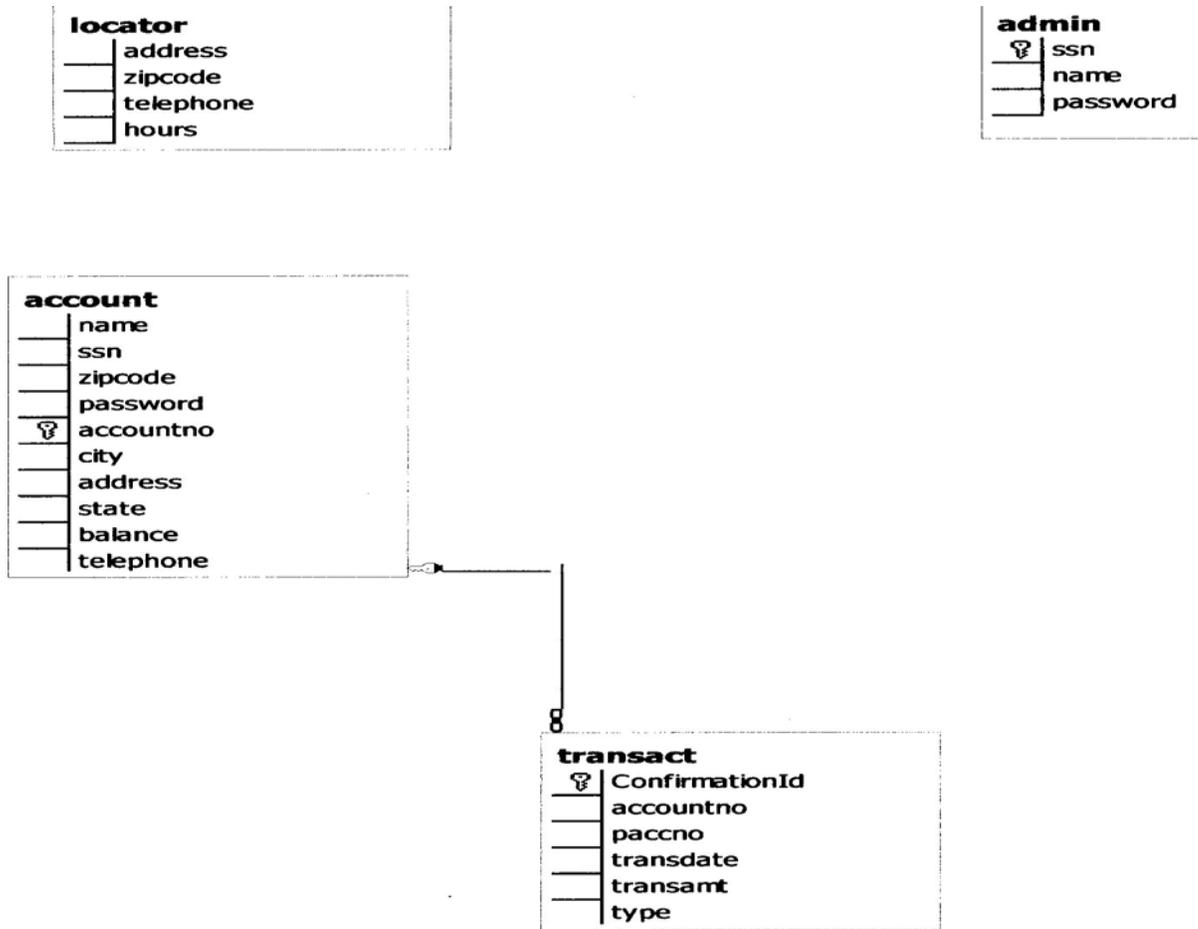
In this module the administrator is allowed to log on to the website and will be allowed to access his/her administrative account by using its user name and password which will then gets verified with the database. Depending upon their authentication, the administrative page will be displayed, where administrator is allowed to assign credit limits for the customer depending upon what kind of request the customer has made. The administrator will be allowed to put a hold on the credit card account for security reasons, like lost/stolen credit card and this facility will protect customer information from getting misused by others. Once the administrator selects the type of card or loan requested and the social security number from the database on a single click the credit limit will be assigned. There will be a special field in the database, which will allow the administrator to keep track of the customers who have already been issued their cards.

The application will be having other functionalities such as:

- Password retrieval for existing customer.
- Locator [in case of teller banking]
- Career at the bank.
- Enrollment for online banking.

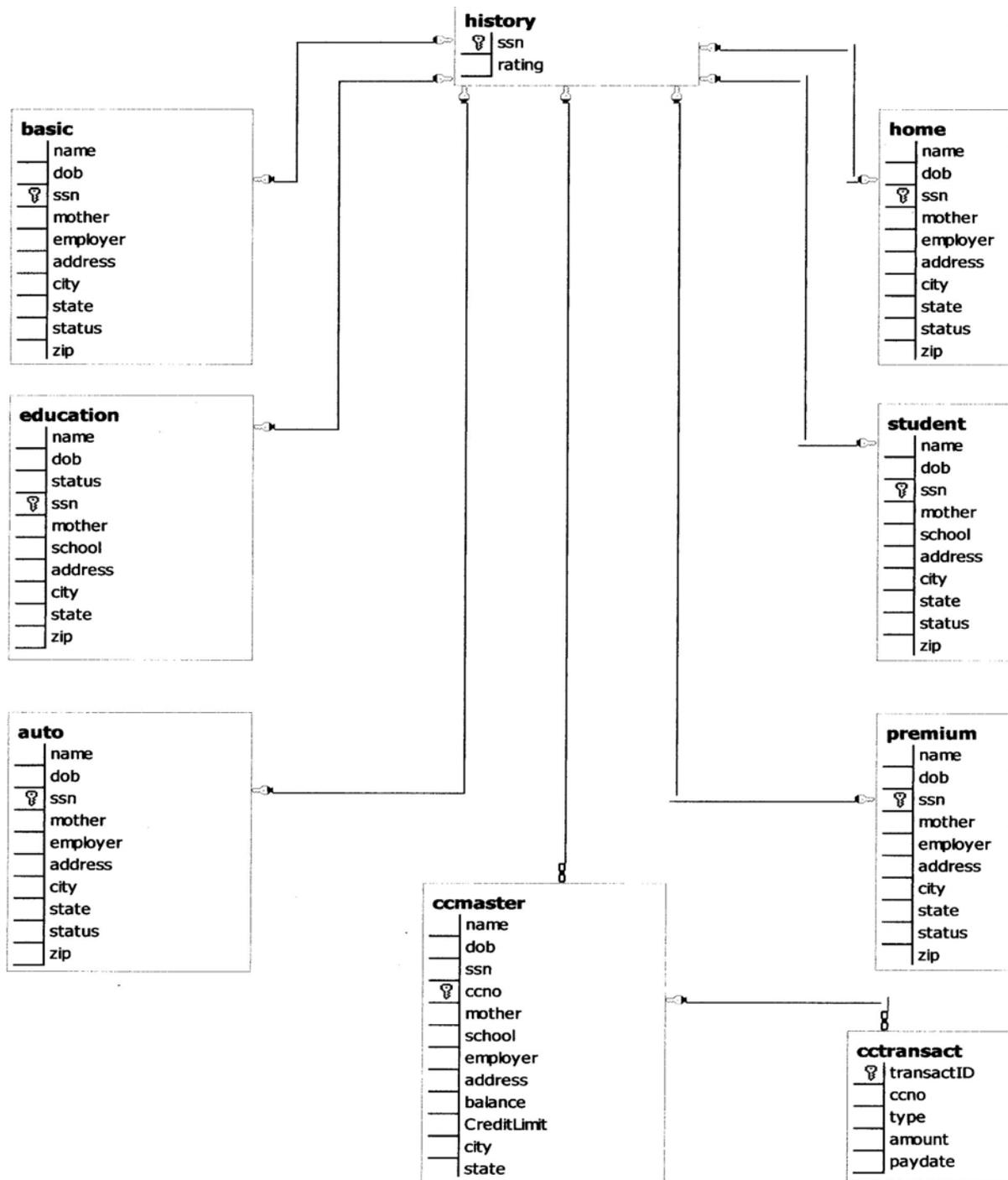
The application will be using a backend as MS- Access database, all the front-end will be written in Java Server Pages (JSP) and Jakarta Tomcat application server will be used as a middle ware which will take care of the connection between front-end and backend.

**Database Design:**



1-1

Figure 2: Entity- Relationship diagram for Online Bank account customer and Administrator



1-2

Figure 3: Entity-Relationship diagram for Online Credit Card Customer and Administrator

*Dataflow diagram for bank account customer*

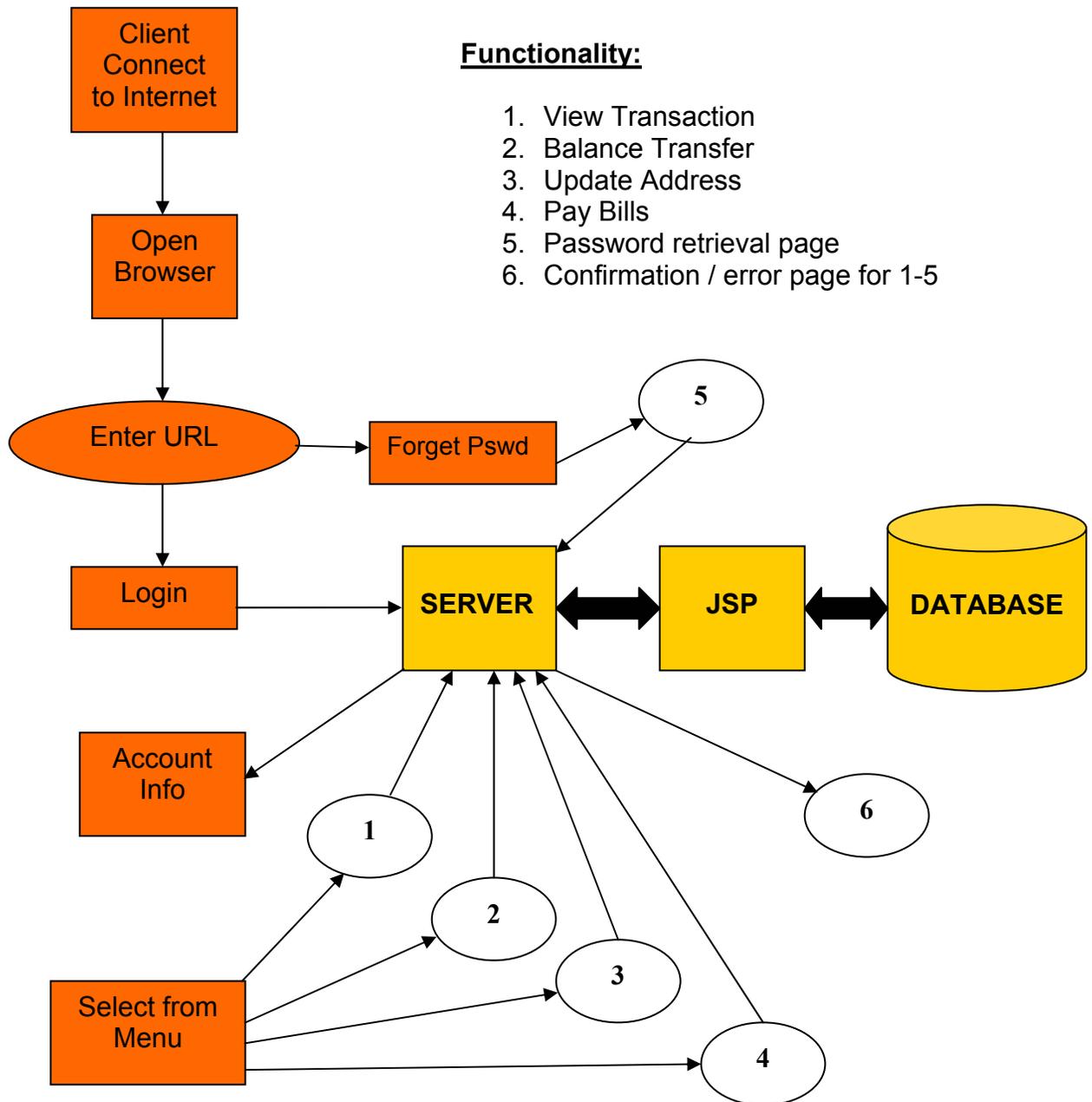


Figure 4: Dataflow diagram for bank account customer

Similarly, we have the dataflow diagram for Credit card customer, Bank Administrator, Credit card administrator and some extra features. The web link for more information <http://students.uis.edu/makht01s/research27/diagrams.htm>

## Implementation

As the Online Banking Application is develop in increments, features will be coded as soon as the designs for that feature have been completed. If design changes are required due to requirements changes of flaws that are discovered, work on this phase will be suspended and the analysis and design phases will be reinitiated to make the necessary design changes. After those changes have been made, the implementation phase will continue. A new design version is created every time that a design change is required.

The following technologies are used in this project:

1. Java Server Pages (JSP)
2. Database (MS-SQL Server2000)
3. Tomcat-apache server.
4. Secure Socket Layer (SSL)
5. Java Database Connectivity (JDBC)

**Java Server Pages (JSP)** is use create dynamic web content even easier, Java has introduced Java Server Pages (also called JSPs). While to write Servlets can require some pretty extensive knowledge of Java, a newbie to Java can learn how to do some pretty neat things with JSPs in a hurry.

*“The JSP technology is actually built on top of Servlets; the two technologies actually work well together. It is common to use both in the same web application”. [4]*

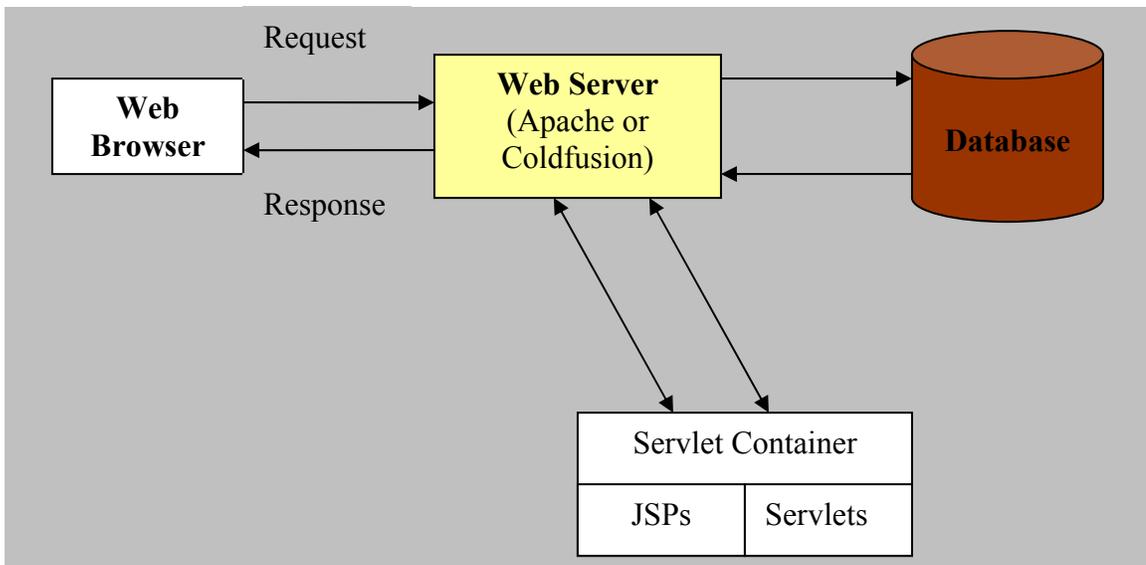


Figure 5: Architectures of JSP

### *Advantages of Java Server Pages*

1. Easier to use JSP technology without needing to learn the Java language
2. Easier to extend the JSP language
3. Easier to write and maintain pages

### ***Database***

A collection of programs that enables you to store, modify, and extract information from a database. Information of the users gets stored in a relational database. The application works well with MS- Access as database, it also works well with SQL Server 2000 and Oracle.

Data Manipulation Language (DML) is used to modify the data present in database:

- SELECT - extracts data from a database table
- UPDATE - updates data in a database table
- DELETE - deletes data from a database table
- INSERT INTO - inserts new data into a database table

The Data Definition Language (DDL) is used to define the data:

- CREATE TABLE - creates a new database table
- ALTER TABLE - alters (changes) a database table
- DROP TABLE - deletes a database table
- CREATE INDEX - creates an index (search key)
- DROP INDEX - deletes an index

The relationship diagram for the application is shown under Database Design. It contains 13 tables, relationship present in the table are one-to-many and one-to-one. The database design is fully normalized in third normal form 3NF to maximum extent.

### ***Java Database connectivity (JDBC)***

JDBC technology is an API that lets us access virtually any tabular data source from the Java programming language. It provides cross-DBMS connectivity to a wide range of SQL databases, and now, with the new JDBC API, it also provides access to other tabular data sources, such as spreadsheets or flat files.

### ***Secure Socket Layer (SSL)***

SSL, or Secure Socket Layer, is a technology, which allows web browsers and web servers to communicate over a secured connection. This means that the data being sent is encrypted by one side, transmitted, and then decrypted by the other side before processing. This is a two-way process, meaning that both the server AND the browser encrypt all traffic before sending out data.

Another important aspect of the SSL protocol is Authentication. This means that during your initial attempt to communicate with a web server over a secure connection, that server will present your web browser with a set of credentials, in the form of a "Certificate", as proof that site is who and what it claims to be. In certain cases, the server may also request a Certificate from your web browser, asking for proof that *you* are who you claim to be. This is known as "Client Authentication," SSL uses public key encryption for encrypting data. The figure below will give a better understanding about public key encryption.[5]

# Public Key Encryption

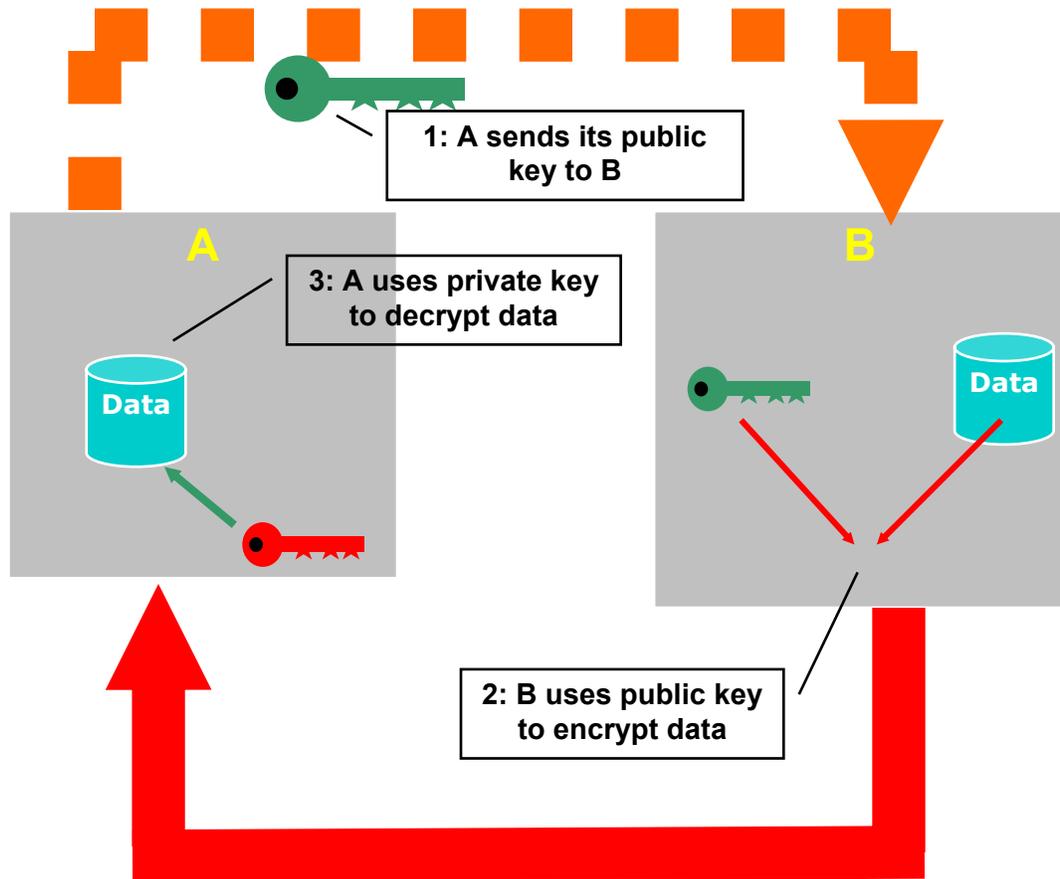


Figure 6: Public key encryption

## System Requirement

*Minimum system requirements are listed below:*

<b>Processor</b>	Pentium III 550 MHz or Athlon AMD 1GHz
<b>RAM:</b>	256 MB or more
<b>Operating System:</b>	Windows 2000 Server, Windows XP with Java Virtual Machine enabled
<b>Database</b>	Oracle/MS Access/SQL Server 2000
<b>Hard Disk space:</b>	20 MB
<b>Web Server</b>	Cold fusion Server or Jakarta-Tomcat-Apache
<b>Web Browser</b>	Internet Explorer 5.0 or higher
<b>Software</b>	Macromedia Dream waver MX, SSL, MS-Access,

Table 4: Hardware and Software Requirements

## Conclusion

This paper shows, how Distributed applications like online Banking, can easily be developed using Enterprises Java Technologies and its distributed architectures [Page-Centric, Page-View, and Model 1]. A more efficient approach would be to utilize Model View Controller 2 architecture, which makes application reusable, robust, and more object-oriented. J2EE, as distributed component architecture, is the right solution in developing web applications.

## References

1. Online banking of National City bank [www.Nationalcity.com](http://www.Nationalcity.com)
2. Online banking of Bank one [www.BankOne.com](http://www.BankOne.com)
3. Online banking of TCF bank [www.TCF.com](http://www.TCF.com)
4. Falkner Jayson, Galbraith Ben, Irani Romin, Kochmer Casey, Moidoo Meeraj Kunnumputath, Punduranga Sathya Narayana, Perrumal Krishnaraj, Timney John. *Beginning JSP web Development (Wrox Press. 2001)*
5. <http://jakarta.apache.org/tomcat/tomcat-4.0-doc/ssl-howto.html>
6. <http://java.sun.com>
7. <http://jguru.com>

## Acknowledgements

I would sincerely like to thank my advisor, Dr. Kaymar Dezhgosha for providing invaluable insights, encouragement as well as guidance during the preparation of this paper. He has always been a source of enthusiasm and inspiration for me.

I extend my special thanks to friend Sherjeel M Ghouse, Dr. Ted Mims and Prof. Mike Parker for sparing their crucial time and effort in guiding me when I get stopped at some point.

The Department of Computer Science has supported me with numerous resources during my stay at University of Illinois, Springfield. I am grateful for their assistance.

Words cannot truly express my deepest gratitude and appreciation to my father, mother and sisters who always gave me their love and emotional support. I want to express my gratitude to my brothers, who has always encouraged me and guided me to independence, never trying to limit my aspirations. I am grateful to my parents and amazed at their generosity.