

Developing Application Framework for Secure and Distributed Banking

Submitted in partial fulfillment of the Degree of

Bachelor of Technology

in

Information Technology

Under the Supervision of

Dr. Vivek Kumar Sehgal

Associate Professor, Information Technology

By

Ashish Gupta

Enrollment no.: 111439

To



Jaypee University of Information and Technology

Waknaghat, Solan – 173234, Himachal Pradesh

Certificate

This is to certify that project report entitled “Application Framework for Secure and Distributed Banking”, submitted by Ashish Gupta (111439) in partial fulfillment for the award of degree of Bachelor of Technology in Information Technology Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or Bachelor.

Date:

Dr. Vivek Sehgal
(Associate Professor)

Acknowledgement

On the very outset of this report, I would like to extend my sincere & heartfelt obligation towards all the personages who have helped me in this endeavor. Without their active guidance, help, cooperation & encouragement, I would not have made headway in the project.

I would like to show my greatest appreciation to **Dr. Ashish Gupta**. I feel motivated every time I get her encouragement. For her coherent guidance throughout the tenure of the project, I feel fortunate to be taught by **Dr. Ashish Gupta**, who gave me her unwavering support. Besides being my mentor, she taught me that there is no substitute for hard work.

I express my gratitude and sincere thanks to **Prof. R.M.K. Sinha**, Dean, Department of Computer Science and Engineering, for allowing me to undertake this project.

I deeply express my sincere thanks to **Brig. (Retd.) S. P. Ghreera**, Head of Department, Department of Computer Science and Engineering, for encouraging and allowing me to present this project

I would also thank my parents **Mr. Mahesh Chand Gupta** and **Mrs. Mamta Gupta** for teaching me how to be successful and how to achieve my goal which helped in completing this project.

They have supported me in every step of my life.

Date:

Ashish Gupta

ABSTRACT

This project is aimed at developing an Online Banking for customer. The system is an online application that can be accessed throughout the organization and outside as well with proper login provided.

The project has been planned to be having the view of distributed architecture, with centralized storage of the database. The application for the storage of the data has been planned. Using the constructs of Oracle 10g and all the user interfaces have been designed using the JAVA. The database connectivity is planned using the “Database” methodology. The standards of security and data protective mechanism have been given a big choice for proper usage. The application takes care of different modules and their associated reports, which are produced as per the applicable strategies and standards that are put forwarded by the administrative staff.

The entire project has been developed keeping in view of the distributed client server computing technology, in mind. The specification has been normalized up to 3NF to eliminate all the anomalies that may arise due to the database transaction that are executed by the general users and the organizational administration. The user interfaces are browser specific to give distributed accessibility for the overall system. The internal database has been selected as Oracle 10g. The basic constructs of table spaces, clusters and indexes have been exploited to provide higher consistency and reliability for the data storage. At all proper levels high care was taken to check that the system manages the data consistency with proper business rules or validations. The database connectivity was planned using the latest “Database connection” technology provided by MS ACCESS. The authentication and authorization was crosschecked at all the relevant stages. The user level accessibility has been restricted into two zones namely.

This project aims at creation of a secure Internet banking system. This will be accessible to all customers who have a valid User Id and Password. This is an approach to provide an opportunity to the customers to have some important transactions to be done from where they are at present without moving to bank. In this project we are going to deal the existing facts in the bank i.e.; the transactions which takes place between customer and bank.

Table of Content

S.No	Topic	Page No.
1	Introduction	1
1.1	Overview	1
1.2	Motivation	1
1.3	Problem Statement	1
1.4	Project Scope	2
2	System Analysis	3
2.1	Present System	3
2.2	Proposed System	3
2.3	Benefits of the system	3
2.4	Banking system can be used extensively	4
3	Feasibility report	5
3.1	Understanding feasibility	5
3.2	Economic feasibility	5
3.3	Technical feasibility	5
3.4	Behavioral feasibility	5
4	Security and Authentication techniques	7
4.1	Security and Privacy Issues	7

4.2	Authentication techniques	8
4.3	Firewall	9
4.4	Cryptography	10
4.4	Digital Signature and Certification	10
4.5	AES Encryption	11
5	Risks and Attacks associated with Online Banking	17
5.1	Operational risks	17
5.2	Security risks	17
5.3	Legal	18
5.4	Different types of Attacks	18
6	Application Design	20
6.1	Website	20
6.2	Android app	25
6.3	Code Snippets	32
6.4	Database Tables	46
6.5	Login process	48
6.6	Registration process	48
6.7	DFD level 0	49
6.8	Use case	49
7	Conclusion	50
8	References	51

List of Figures

S.No	Topic	Page No.
1	Home page	20
2	Registration page	21
3	After login page	22
4	Balance transfer page	23
5	Balance view page	24
6	Transaction history page	24
7	Android Home page	27
8	Android Login page	28
9	Android Registration page	29
10	Android Profile page	30
11	Android Balance transfer page	31
12	Android Balance view page	32

List of Tables

S.No	Topic	Page No.
1	User_info	47
2	Balance	47
3	Transaction_details	48
4	Acc_details	48

Chapter 1: Introduction

1.1 Overview

Internet Banking is all about knowing our customer need and provide them with the right service at the right time through right channel 24*7 day a week.

Being “electronic”, it not only provides its customers with faster and better facilities, it even reduces the manual overhead of accounts maintenance.

1.2 Motivation

Online banking is one of the projects topics which has considerable Commercial significance. It's coding is not much difficult, however, it will require considerable knowledge on web page designing, applets & graphics. This makes the project challenging yet approachable .As continuously there have been cyber attacks on the online banking website trying to access the accounts unofficially . Thus we decided to take up this as our project topic.

1.3 Problem Statement

The major concern for an Internet -banking is the ‘security’. There are many remote customers accessing the system and placing various requests/queries to get the required information or to make transactions with the bank at the time demanded. There are various aspects that are needed to address in this application. There should be a report generating Balance Enquiry the system need to guarantee the funds transfer to another account of the same bank. The system should provide assistance for request for:

1. Cheque book
2. Change of address

The system must generate various reports for the customers to view monthly and annual Statements.

1.4 Project Scope

This Project investigates the entry threshold for providing a new transaction service channel via the real options approach, where the entry threshold is established by using an Internet banking system designed for the use of normal users (individuals), Industrialists under transaction rate uncertainty.

1. Customer must have a valid User Id and password to login to the system
2. If a wrong password is given thrice in succession, that account will be locked and the customer will not be able to use it.
3. When an invalid password is entered a warning is given to the user that his account is going to get locked.
4. After the valid user logs in he is shown the list of accounts he has with the bank.
5. On selecting the desired account he is taken to a page which shows the present balance in that particular account number.
6. User can request for the details of the last 'n' number of transactions that he has performed.

A report can also be taken of this.

1. User can make a funds transfer to another account in the same bank. User is provided with a transaction password which is different from the login password.
2. User can transfer funds from his account to any other account with this bank. If the transaction is successful a notification should appear to the customer, in case it is unsuccessful, a proper message should be given to the customer as to why it failed.
3. User can request for cheque book/change of address/stop payment of cheque's.
4. User can view his monthly as well as annual statements. He can also take print out of the same.
5. Generate reports at every section
6. Administrator can take a back up of the database for every instance that is happening, periodically.

Chapter 2: System Analysis

2.1 Present System

The developed system is an innovation in the area of private banking. In the existing system the no. of staff required for completing the work is more, while the new system requires lesser staffs generally.

The data entry process requires the data on the paper, which is then feed into the application by the operator while doing so; the data entry operator has to look into the paper again & again and thus the chances of in accuracies in the typed contents increases. Also the process includes higher transportation cost, increased handling cost, more time delays, low accuracy, more usage of resources like registers, books, papers, etc.

2.2 Proposed System

“Why an Automated Private Banking System?”

1. Almost 60% of today’s information is still paper based.
2. 30% of all office time is spent finding documents.
3. The average time to manage a single document is 12 minutes, 9 minutes to re-file and 3 minutes to process.

Hence the requirement is to develop a system that minimizes all these overheads included while giving the maximum output for the organization.

The basis for the project is to develop a fully automated banking system that includes depositing of amount, withdrawal of amount and exporting the outcome back to the client while considering all the tools and facilities than a client may need for efficient and effective output.

2.3 Benefits of the system

1. Quick, authenticated access to accounts via the desktop.
2. Easily scalable to grow with changing system requirement.
3. Enterprise wide access to information.
4. Improved information security, restricting unauthorized access.

5. Minimize Storage Space

In manual system, much storage space for data files is required so to overcome this problem, an automated well managed database is developed for saving storage space. This s/w saves space and stores information efficiently. It ends the burden of having large manual filing storage system.

2.4 Banking System can be used extensively:

1. Withdrawal of amount by the client.
2. Deposition of amount by the client.
3. Faster balance enquiry.

Chapter 3: Feasibility report

3.1 Understanding Feasibility

Feasibility study means the analysis of problem to determine if It can be solved effectively. In other words it is the study of the possibilities of the proposed system it studies the work ability, impact on the organization ability to meet user's need and efficient use of resources.

3.2 Economical feasibility

The economical analysis checks for the high investment incurred on the system. It evaluates development & implementing charges for the proposed "Banking Project". The S/W used for the development is easily available at minimal cost & the database applied is freely available hence it results in low cost implementation.

3.3 Technical feasibility

This aspect concentrates on the concept of using Computer Meaning, "Mechanization" of human works. Thus the automated solution leads to the need for a technical feasibility study. The focus on the platform used database management & users for that S/W. The proposed system doesn't require an in depth technical knowledge as the system development is simple and easy to understand. The S/W (VB.NET) used makes the system user friendly (GUI). The result obtain should be true in the real time conditions.

3.4 Behavioral feasibility:

Behavioral feasibility deals with the runtime performance of the S/W the proposed system must score higher than the present in the behavioral study. The S/W should have end user

in mind when the system is designed while designing s/w the programmer should be aware of the conditions user's Knowledge input, output, calculations etc.

Chapter 4: Security and Authentication techniques

4.1 Security and Privacy Issues

Terminology:

1. **Security:** Security in Internet banking comprises both the computer and communication security. The aim of computer security is to preserve computing resources against abuse and unauthorized use, and to protect data from accidental and deliberate damage, disclosure and modification. The communication security aims to protect data during the transmission in computer network and distributed system.
2. **Authentication:** It is a process of verifying claimed identity of an individual user, machine, software component or any other entity. For example, an IP Address identifies a computer system on the Internet, much like a phone number identifies a telephone. It may be to ensure that unauthorized users do not enter, or for verifying the sources from where the data are received. It is important because it ensures authorization and accountability. Authorization means control over the activity of user, whereas accountability allows us to trace uniquely the action to a specific user. Authentication can be based on password or network address or on cryptographic techniques.
3. **Access Control:** It is a mechanism to control the access to the system and its facilities by a given user up to the extent necessary to perform his job function. It provides for the protection of the system resources against unauthorized access. An access control mechanism uses the authenticated identities of principals and the information about these principals to determine and enforce access rights. It goes hand in hand with authentication. In establishing a link between a bank's internal network and the Internet, we may create a number of additional access points into the internal operational system. In this situation, unauthorized access attempts might be initiated from anywhere. Unauthorized access causes destruction, alterations, theft of data or funds, compromising data confidentiality, denial of service etc. Access control may be of discretionary and mandatory types.
4. **Data Confidentiality:** The concept of providing for protection of data from unauthorized disclosure is called data confidentiality. Due to the open nature of Internet, unless otherwise protected, all data transfer can be monitored or read by others. Although it is difficult to monitor a transmission at random, because of numerous paths available, special programs such as "Sniffers", set up at an opportunity location like Web server, can collect vital information. This may include credit card number, deposits, loans or password etc. Confidentiality extends beyond data transfer and include any connected data storage system including network storage systems. Password and other access control methods help in ensuring data confidentiality.

5. **Data Integrity:** It ensures that information cannot be modified in unexpected way. Loss of data integrity could result from human error, intentional tampering, or even catastrophic events. Failure to protect the correctness of data may render data useless, or worse, dangerous. Efforts must be made to ensure the accuracy and soundness of data at all times. Access control, encryption and digital signatures are the methods to ensure data integrity.

6. **Non-Repudiation:** Non-Repudiation involves creating proof of the origin or delivery of data to protect the sender against false denial by the recipient that data has been received or to protect the recipient against false denial by the sender that the data has been sent. To ensure that a transaction is enforceable, steps must be taken to prohibit parties from disputing the validity of, or refusing to acknowledge, legitimate communication or transaction.

7. **Security Audit Trail:** A security audit refers to an independent review and examination of system's records and activities, in order to test for adequacy of system controls. It ensures compliance with established policy and operational procedures, to detect breaches in security, and to recommend any indicated changes in the control, policy and procedures. Audit Trail refers to data generated by the system, which facilitates a security audit at a future date.

4.2 Authentication Techniques:

Authentication is a process to verify the claimed identity. There are various techniques available for authentication. Password is the most extensively used method. Most of the financial institutions use passwords along with PIN (Personal Identification Number) for authentication. Technologies such as tokens, smart cards and biometrics can be used to strengthen the security structure by requiring the user to possess something physical.

1. **Token technology** relies on a separate physical device, which is retained by an individual, to verify the user's identity. The token resembles a small hand-held card or calculator and is used to generate passwords. The device is usually synchronized with security software in the host computer such as an internal clock or an identical time based mathematical algorithm. Tokens are well suited for one-time password generation and access control. A separate PIN is typically required to activate the token.

2. **Smart cards** resemble credit cards or other traditional magnetic stripe cards, but contain an embedded computer chip. The chip includes a processor, operating system, and both Read Only Memory (ROM) and Random Access Memory (RAM). They can be used to generate one-time passwords when prompted by host computer, or to carry cryptographic keys. A smart card reader is required for their use.

3. **Biometrics** involves identification and verification of an individual based on some physical characteristic, such as fingerprint analysis, hand geometry, or retina scanning. This technology is advancing rapidly, and offers an alternative means to authenticate a user.

4.3 Firewall

The connection between internal networks and the outside world must be watched and monitored carefully by a gatekeeper of sorts. Firewalls do this job. Otherwise, there is a risk of exposing the internal network and systems, often leaving them vulnerable and compromising the integrity and privacy of data. Firewalls are a component or set of components that restrict access between a protected network and the outside world (i.e., the Internet). They control traffic between outside and inside a network, providing single entry point where access control and auditing can be imposed. All firewalls examine the pieces or packets of data flowing into and out of a network and determine whether a particular person should be given access inside the network.

As a result, unauthorized computers outside the firewall are prevented from directly accessing the computers inside the internal network. Broadly, there are three types of firewalls i.e. Packet filtering firewalls, Proxy servers and stateful inspection firewall.

1. Packet filtering routers:

Packet filtering routers are the simplest form of firewalls. They are connected between the host computer of an internal network and the Internet gateway. The bastion host directs message accepted by the router to the appropriate application servers in the protected network. Their function is to route data of a network and to allow only certain types of data into the network by checking the type of data and its source and destination address. If the router determines that the data is sourced from an Internet address which is not on its acceptable or trusted sources list, the connection would be simply refused. The advantage of this type of firewall is that it is simple and cheaper to implement and also fast and transparent to the users. The disadvantage is that if the security of the router were compromised, computers on the internal network would be open to external network for attacks. Also, the filtering rule scan be difficult to configure, and a poorly configured firewall could result in security loopholes by unintentionally allowing access to an internal network.

2. Proxy servers:

Proxy servers control incoming and outgoing traffic for a network by executing specific proxy program for each requested connection. If any computer outside the internal network wants to access some application running on a computer inside the internal network, then it would actually communicate with the proxy server, and proxy server in turn will pass the request to the internal computer and get the response which will be given to the recipient (outside user). That is, there is no direct connection between the internal network and Internet. This approach allows a high level of control and in-depth monitoring using logging and auditing tools. However, since it doubles the amount of processing, this approach may lead to some degradation in performance.

3. Stateful Inspection firewall:

This type of firewalls thoroughly inspects all packets of information at the network level as in the case of proxy servers. Specifications of each packet of data, such as the user and the transportation method, the application used are all queried and verified in the inspection process. The information collected is maintained so that all future transmissions are

inspected and compared to past transmission. If both the “state” of the transmission and the “context” in which it is used deviate from normal patterns, the connection would be refused. This type of firewalls are very powerful but performance would also decline due to the intensive inspection and verification performed.

4.4 Cryptography

The process of disguising a message in such a way as to hide its substance is called encryption. An encrypted message is called cipher text. The process of turning a cipher text back into plain text is called decryption. Cryptography is the art and science of keeping messages secure. It uses a ‘key’ for encrypting or decrypting a message. Both the method of encryption and the size of key are important to ensure confidentiality of a message. There are two types of encryption: Symmetric key and Asymmetric key encryption. In the symmetric key cryptography scheme, the same key is used to encrypt and decrypt the message.

Common symmetric algorithms include One-time pad encryption, Data Encryption Standard (DES), Triple DES, LOKI, Twofish, Blowfish, International Data Encryption Algorithm (IDEA). DES and Triple DES are the commonly used techniques. Asymmetric key cryptography scheme is also known as Public key crypto-system. Here two keys are used. One key is kept secret and therefore it is referred as “private key”. The other key is made widely available to anyone who wants it, and is referred as “Public key”. The Public key and Private key are mathematically related so that information encrypted using the public key can only be decrypted by the corresponding private key and vice-versa. Importantly, it is near to impossible to find out the private key from the public key. Common and more popular public key cryptosystem algorithms are Diffie-Hellman, RSA, Elliptic Curve etc. In all these, the confidentiality is directly related to the key size. Larger the key size, the longer it takes to break the encrypted message.

- Diffie-Hellman: This is the first public key algorithm invented. It gets its security from the difficulty of calculating discrete logarithms in a finite field. Diffie-Hellman method can be used for distribution of keys to be used for symmetric encryption.
- RSA: Named after its three inventors, Ron Rivest, Adi Shamir and Leonard Adleman, who first introduced the algorithm in 1978, RSA gets its security from the difficulty of factoring large numbers. The public and private keys are function of a pair of large (100 or 200 digits or even larger) prime numbers. The pair is used for asymmetric encryption.

4.5 Digital Signature and certification:

Digital signatures authenticate the identity of a sender, through the private, cryptographic key. In addition, every digital signature is different because it is derived from the content of

the message itself. The combination of identity authentication and singularly unique signatures results in a transmission that can not be repudiated. Digital signature can be applied to any data transmission, including e-mail. To generate digital signature, the original, unencrypted message is processed through mathematical algorithms that generate a 'message digest' (a unique character representation of data). This process is known as "hashing".

The message digest is then encrypted with the private key and sent along with the message (could be encrypted also). The recipient receives both the message and encrypted message digest. The recipient decrypts the message digest using the sender's public key, and then runs the message through the hash function again. If the resulting message digest matches the one sent with the message, the message has not been altered and data integrity is verified. Because the message digest was encrypted using the private key, the sender can be identified and bound to the specific message.

Certification Authorities and Digital Certificates:

Certificate Authorities and Digital Certificates are emerging to further address the issues of authentication, non-repudiation, data privacy and cryptographic key management. A Certificate Authority (CA) is a trusted third party that verifies the identity of a party to a transaction. To do this, the CA vouches for the identity of a party by attaching the CA's digital signature to any messages, public keys, etc., which are transmitted. The CA must be trusted by the parties involved, and identities must have been proven to the CA beforehand. Digital certificates are messages that are signed with the CA's private key. They identify the CA, the represented party, and even include the represented party's public key.

Secure Socket Layer (SSL):

SSL is designed to make use of TCP to provide a reliable end-to-end secure service. The SSL servers have digital certificates issued by Certifying Authorities so that the clients can authenticate the service provider (a bank in our case). The servers use a password /PIN/digital certificate to authenticate clients. Once the clients and server have authenticated each other, they establish a session key for encryption of messages.

4.6 AES Encryption

The Advanced Encryption Standard (AES), also referenced as Rijndael (its original name), is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001.

AES is based on the Rijndael cipher developed by two Belgian cryptographers, Joan Daemen and Vincent Rijmen, who submitted a proposal to NIST during the AES selection process. Rijndael is a family of ciphers with different key and block sizes.

For AES, NIST selected three members of the Rijndael family, each with a block size of 128 bits, but three different key lengths: 128, 192 and 256 bits.

AES has been adopted by the U.S. government and is now used worldwide. It supersedes the Data Encryption Standard (DES), which was published in 1977. The algorithm described by AES is a symmetric-key algorithm, meaning the same key is used for both encrypting and decrypting the data.

AES is based on a design principle known as a substitution-permutation network, combination of both substitution and permutation, and is fast in both software and hardware. Unlike its predecessor DES, AES does not use a Feistel network. AES is a variant of Rijndael which has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits. By contrast, the Rijndael specification *per se* is specified with block and key sizes that may be any multiple of 32 bits, both with a minimum of 128 and a maximum of 256 bits.

AES operates on a 4×4 column-major order matrix of bytes, termed the *state*, although some versions of Rijndael have a larger block size and have additional columns in the state. Most AES calculations are done in a special finite field.

The key size used for an AES cipher specifies the number of repetitions of transformation rounds that convert the input, called the plaintext, into the final output, called the cipher text. The number of cycles of repetition are as follows:

- 10 cycles of repetition for 128-bit keys.
- 12 cycles of repetition for 192-bit keys.
- 14 cycles of repetition for 256-bit keys.

Each round consists of several processing steps, each containing four similar but different stages, including one that depends on the encryption key itself. A set of reverse rounds are applied to transform cipher text back into the original plaintext using the same encryption key.

High-level description of the algorithm

1. KeyExpansions—round keys are derived from the cipher key using Rijndael's key schedule. AES requires a separate 128-bit round key block for each round plus one more.
2. InitialRound
 1. AddRoundKey—each byte of the state is combined with a block of the round key using bitwise xor.
3. Rounds
 1. SubBytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.

2. ShiftRows—a transposition step where the last three rows of the state are shifted cyclically a certain number of steps.
 3. MixColumns—a mixing operation which operates on the columns of the state, combining the four bytes in each column.
 4. AddRoundKey
4. Final Round (no MixColumns)
 1. SubBytes
 2. ShiftRows
 3. AddRoundKey.

The Subbytes step

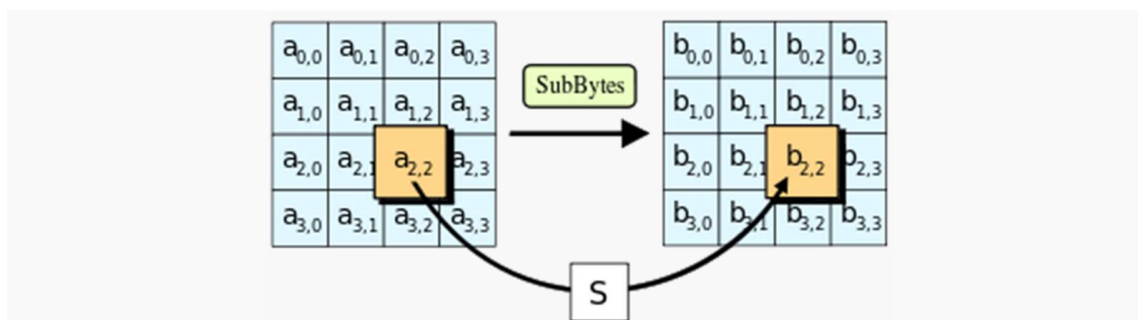


Figure 1

In the SubBytes step, each byte in the state is replaced with its entry in a fixed 8-bit lookup table, S ; $b_{ij} = S(a_{ij})$.

In the SubBytes step, each byte $a_{i,j}$ in the *state* matrix is replaced with a SubByte $S(a_{i,j})$ using an 8-bit substitution box, the Rijndael S-box. This operation provides the non-linearity in the cipher. The S-box used is derived from the multiplicative inverse over $GF(2^8)$, known to have good non-linearity properties. To avoid attacks based on simple algebraic properties, the S-box is constructed by combining the inverse function with an invertible affine transformation. The S-box is also chosen to avoid any fixed points (and so is a derangement), i.e., $S(a_{i,j}) \neq a_{i,j}$, and also any opposite fixed points, i.e., $S(a_{i,j}) \oplus a_{i,j} \neq 0xFF$. While performing the decryption, Inverse SubBytes step is used, which requires first taking the affine transformation and then finding the multiplicative inverse (just reversing the steps used in SubBytes step).

The ShiftRows step

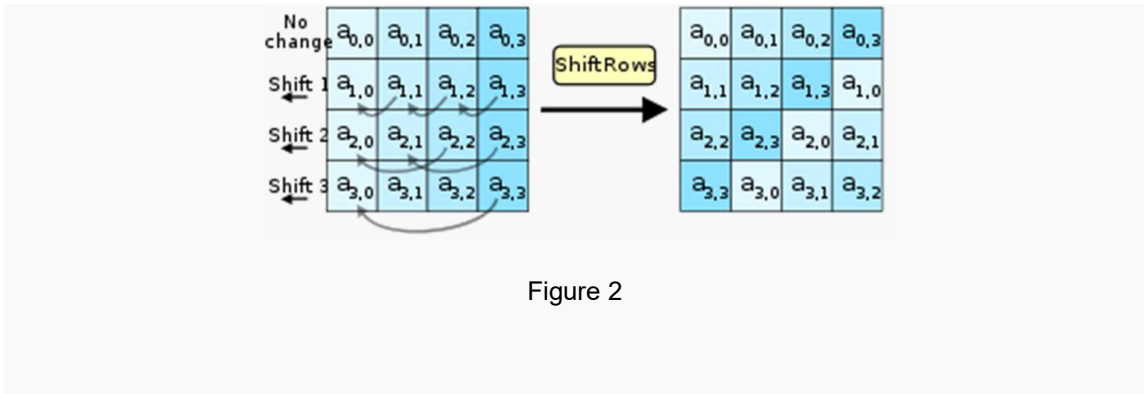


Figure 2

In the ShiftRows step, bytes in each row of the state are shifted cyclically to the left. The number of places each byte is shifted differs for each row.

The ShiftRows step operates on the rows of the state; it cyclically shifts the bytes in each row by a certain offset. For AES, the first row is left unchanged. Each byte of the second row is shifted one to the left. Similarly, the third and fourth rows are shifted by offsets of two and three respectively. For blocks of sizes 128 bits and 192 bits, the shifting pattern is the same. Row n is shifted left circular by $n-1$ bytes. In this way, each column of the output state of the ShiftRows step is composed of bytes from each column of the input state. (Rijndael variants with a larger block size have slightly different offsets). For a 256-bit block, the first row is unchanged and the shifting for the second, third and fourth row is 1 byte, 3 bytes and 4 bytes respectively—this change only applies for the Rijndael cipher when used with a 256-bit block, as AES does not use 256-bit blocks. The importance of this step is to avoid the columns being linearly independent, in which case, AES degenerates into four independent block ciphers.

The MixColumns step

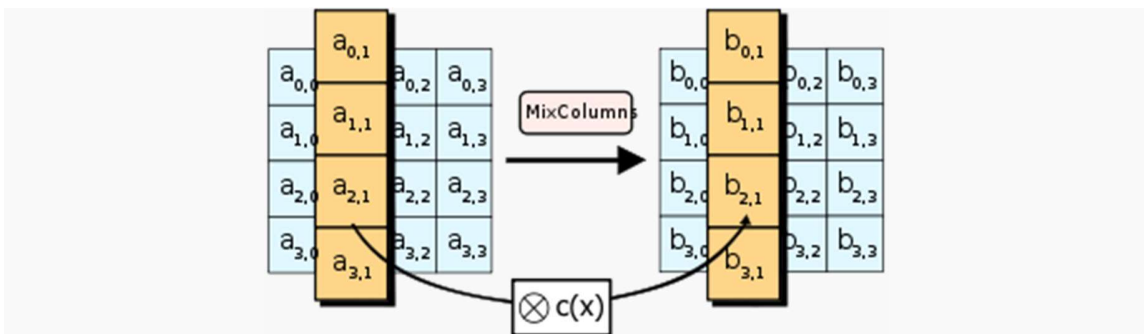


Figure 3

In the MixColumns step, each column of the state is multiplied with a fixed polynomial $c(x)$.

In the MixColumns step, the four bytes of each column of the state are combined using an invertible linear transformation. The MixColumns function takes four bytes as input and outputs four bytes, where each input byte affects all four output bytes. Together with ShiftRows, MixColumns provides diffusion in the cipher.

During this operation, each column is multiplied by a fixed matrix:

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

Matrix multiplication is composed of multiplication and addition of the entries, and here the multiplication operation can be defined as this: multiplication by 1 means no change, multiplication by 2 means shifting to the left, and multiplication by 3 means shifting to the left and then performing XOR with the initial unshifted value. After shifting, a conditional XOR with 0x1B should be performed if the shifted value is larger than 0xFF. (These are special cases of the usual multiplication in $GF(2^8)$.) Addition is simply XOR.

In more general sense, each column is treated as a polynomial over $GF(2^8)$ and is then multiplied modulo x^4+1 with a fixed polynomial $c(x) = 0x03 \cdot x^3 + x^2 + x + 0x02$. The coefficients are displayed in their hexadecimal equivalent of the binary representation of bit polynomials from $GF(2)[x]$. The MixColumns step can also be viewed as a multiplication by the shown particular MDS matrix in the finite field $GF(2^8)$. This process is described further in the article Rijndael mix columns.

The AddRoundKey step

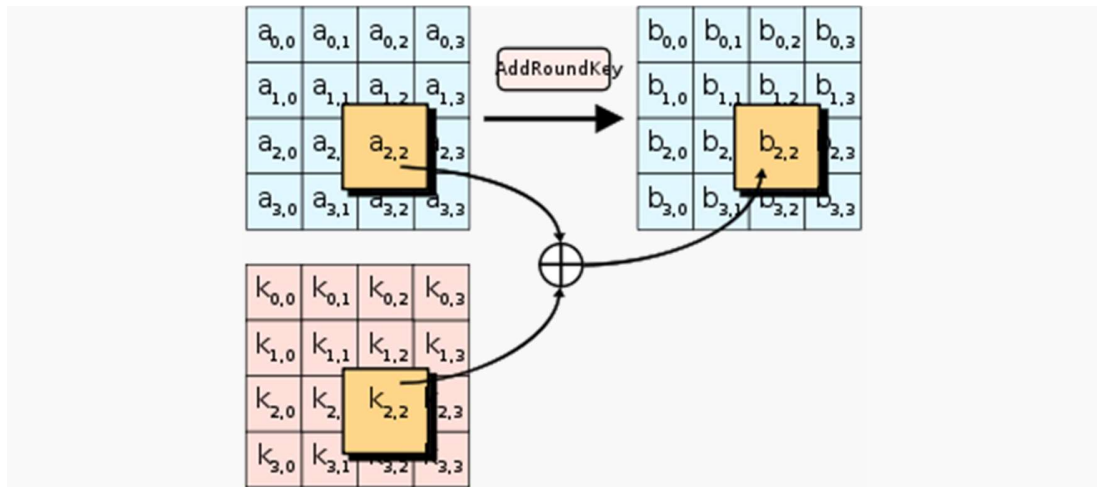


Figure 4

In the AddRoundKey step, each byte of the state is combined with a byte of the round subkey using the XOR operation (\oplus).

In the AddRoundKey step, the subkey is combined with the state. For each round, a subkey is derived from the main key using Rijndael's key schedule; each subkey is the same size as the state. The subkey is added by combining each byte of the state with the corresponding byte of the subkey using bitwise XOR.

Chapter 5: Risks and Attacks associated with Online Banking

5.1 Operational risks

Operational risk, also referred to as transactional risk is the most common form of risk associated with i-banking. It takes the form of inaccurate processing of transactions, non-enforceability of contracts, compromises in data integrity, data privacy and confidentiality, unauthorized access / intrusion to bank's systems and transactions etc. Such risks can arise out of weaknesses in design, implementation and monitoring of banks' information system. Besides inadequacies in technology, human factors like negligence by customers and employees, fraudulent activity of employees and crackers/hackers etc. can become potential source of operational risk. Often there is thin line of difference between operational risk and security risk and both terminologies are used interchangeably.

5.2 Security risks

Internet is a public network of computers which facilitates flow of data / information and to which there is unrestricted access. Banks using this medium for financial transactions must, therefore, have proper technology and systems in place to build a secured environment for such transactions. Security risk arises on account of unauthorized access to a bank's critical information stores like accounting system, risk management system, portfolio management system etc. A breach of security could result in direct financial loss to the bank. For example, hackers operating via the Internet, could access, retrieve and use confidential customer information and also can implant virus. This may result in loss of data, theft of or tampering with customer information, disabling of a significant portion of bank's internal computer system thus denying service, cost of repairing these etc. Other related risks are loss of reputation, infringing customers' privacy and its legal implications etc.

Thus, access control is of paramount importance. Controlling access to banks' system has become more complex in the Internet environment which is a public domain and attempts at unauthorized access could emanate from any source and from anywhere in the world with or without criminal intent. Attackers could be hackers, unscrupulous vendors, disgruntled employees or even pure thrill seekers. Also, in a networked environment the security is limited to its weakest link. It is therefore, necessary that banks critically assess all interrelated systems and have access control measures in place in each of them.

In addition to external attacks banks are exposed to security risk from internal sources e.g. employee fraud. Employees being familiar with different systems and their weaknesses become potential security threats in a loosely controlled environment. They can manage to acquire the authentication data in order to access the customer accounts causing losses to the bank.

5.3 Legal risks

Legal risk arises from violation of, or non-conformance with laws, rules, regulations, or prescribed practices, or when the legal rights and obligations of parties to a transaction are not well established.

Given the relatively new nature of Internet banking, rights and obligations in some cases are uncertain and applicability of laws and rules is uncertain or ambiguous, thus causing legal risk.

Other reasons for legal risks are uncertainty about the validity of some agreements formed via electronic media and law regarding customer disclosures and privacy protection. A customer inadequately informed about his rights and obligations, may not take proper precautions in using Internet banking products or services, leading to disputed transactions, unwanted suits against the bank or other regulatory sanctions.

5.4 Different types of attacks

When a bank's system is connected to the Internet, an attack could originate at anytime from anywhere. Some acceptable level of security must be established before business on the Internet can be reliably conducted. An attack could be any form like:

1. The intruder may gain unauthorized access and nothing more
2. The intruder gains access and destroys, corrupt or otherwise alters data
3. The intruder gains access and seizes control partly or wholly, perhaps denying access to privileged users
4. The intruder does not gain access, but instead forges messages from your system
5. The intruder does not gain access, but instead implements malicious procedures that cause the network to fail, reboot, and hang.

Modern security techniques have made cracking very difficult but not impossible. Furthermore, if the system is not configured properly or the updated patches are not installed then hackers may crack the system using security hole. A wide range of information regarding security hole and their fixes is freely available on the Internet. System administrator should keep himself updated with this information.

Common cracking attacks include:

1. E-mail bomb and List linking
2. Denial-of-Service
3. Sniffer attack
4. Utilizing security hole in the system software
5. E-mail bomb: This is a harassment tool. A traditional e-mail bomb is simply a series of message (perhaps thousands) sent to your mailbox. The attacker's object is to fill the mailbox with junk.

6. Denial-of-Service (DoS) attacks: DoS attacks can temporarily incapacitate the entire network (or at least those hosts that rely on TCP/IP). DoS attacks strike at the heart of IP implementations. Hence they can crop up at any platform, a single DoS attack may well work on several target operating systems. Many DoS attacks are well known and well documented. Available fixes must be applied.

7. Sniffer Attack: Sniffers are devices that capture network packets. They are a combination of hardware and software. Sniffers work by placing the network interface into promiscuous mode. Under normal circumstances, all machines on the network can “hear” the traffic passing through, but will only respond to data addressed specifically to it. Nevertheless, if the machine is in promiscuous mode then it can capture all packets and frames on the network. Sniffers can capture passwords and other confidential information. Sniffers are extremely difficult to detect because they are passive.

Chapter 6: Application Design

6.1 Website

6.1.1 Home Page

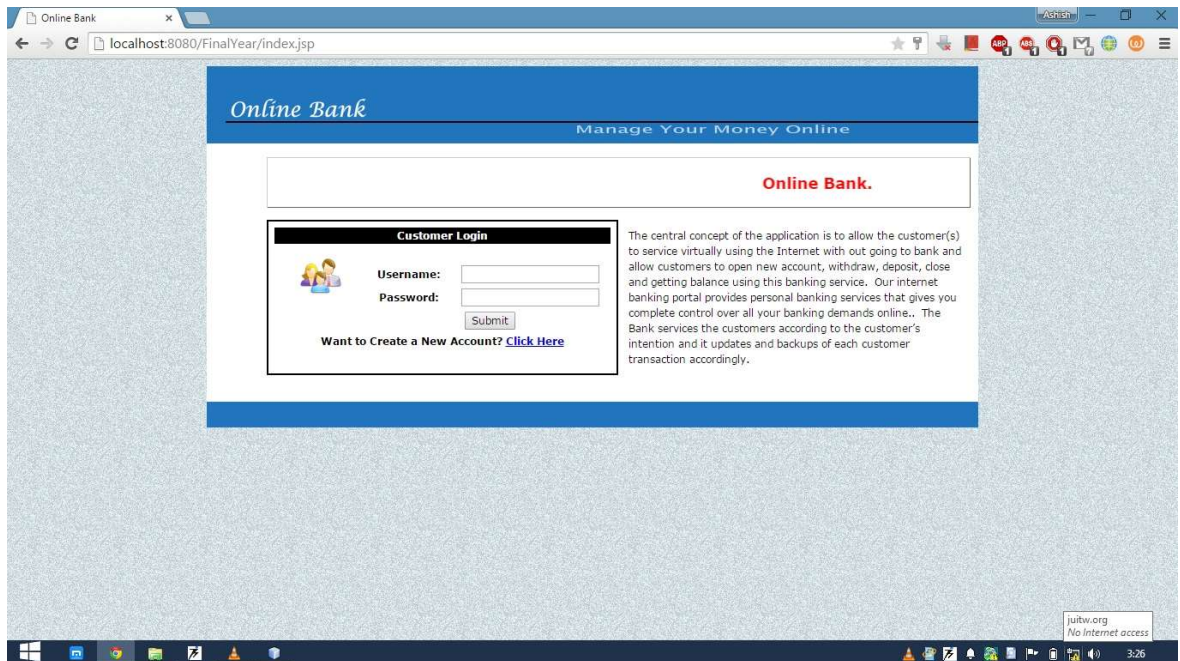


Figure 1

When the user enters the url of the website in the browser tab, the user will get above interface after loading the page.

It has two fields:

1. Username
2. Password

When the user enters the correct username and password which gets checked from the database as first username is checked whether username of name exists or not .If username exists then, password is converted to 32-bit Hexadecimal value and then checked from the database and if encrypted password matches then, session is created for the user using his name and then user is sent to his/her profile page.And if the user is not registered, he/she can click on the button **CLICK HERE** to create his/her online account.

6.1.2 Registration Page

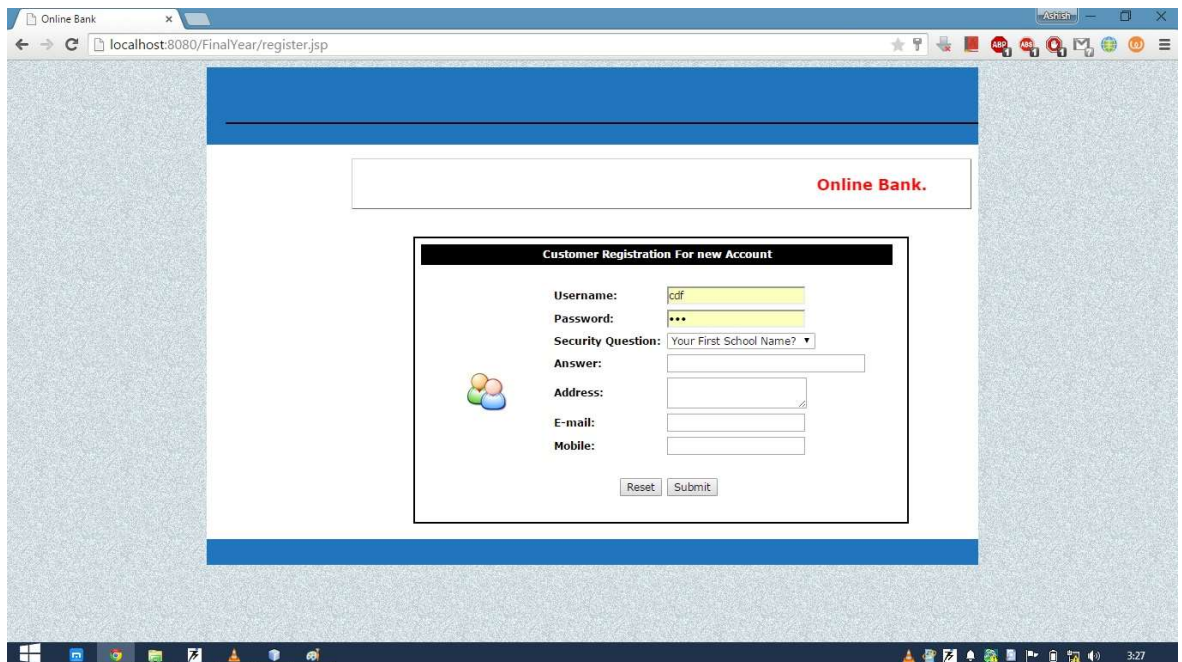


Figure 2

If the user is not registered, he/she can sign up on the website by filling up the details asked above in the image such as:

Username:

Password:

Security Question:

Answer:

Address:

Email:

Mobile:

If any of the field remained empty then, user can't create the account.

And if all the fields are filled, the user can click on the submit button and if no username exists with this username, then online account will be created.

6.1.3 After Login Page

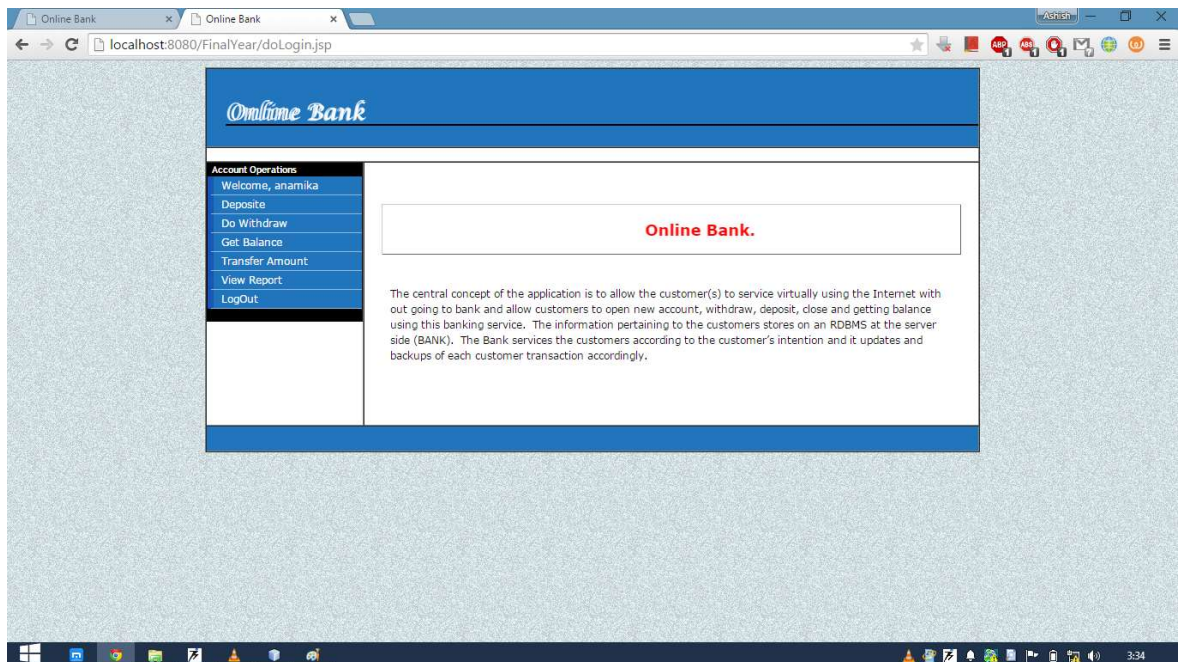


Figure 3

After user login to his/her account using username and password, his/her profile page will be opened.

Here user has various options to perform such as:

1. View balance
2. Transfer Balance
3. View Transaction History
4. Logout

6.1.4 Balance Transfer Page

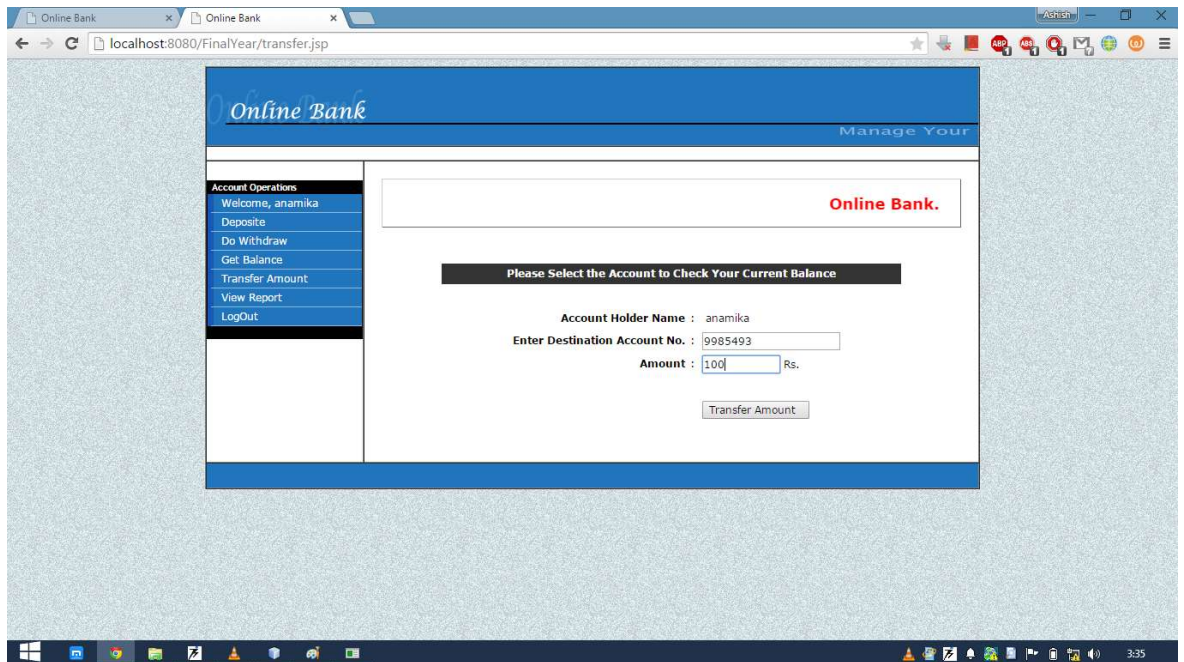


Figure 4

Here user can transfer his/her money anyone registered on this website. User has to enter some fields to transfer the money such as

1. Account Holder name
2. Destination Account no
3. Amount

Here first all the fields must be filled otherwise you will not be able to transfer the money. After that firstly, your name will be checked whether it is correct or not.

Then whether destination account no entered exists or not. If not, then transaction will be failed and message will be displayed.

Thirdly if destination account no entered exists then, amount to be transferred is checked whether that much amount is in your account or not.

6.1.5 Balance View

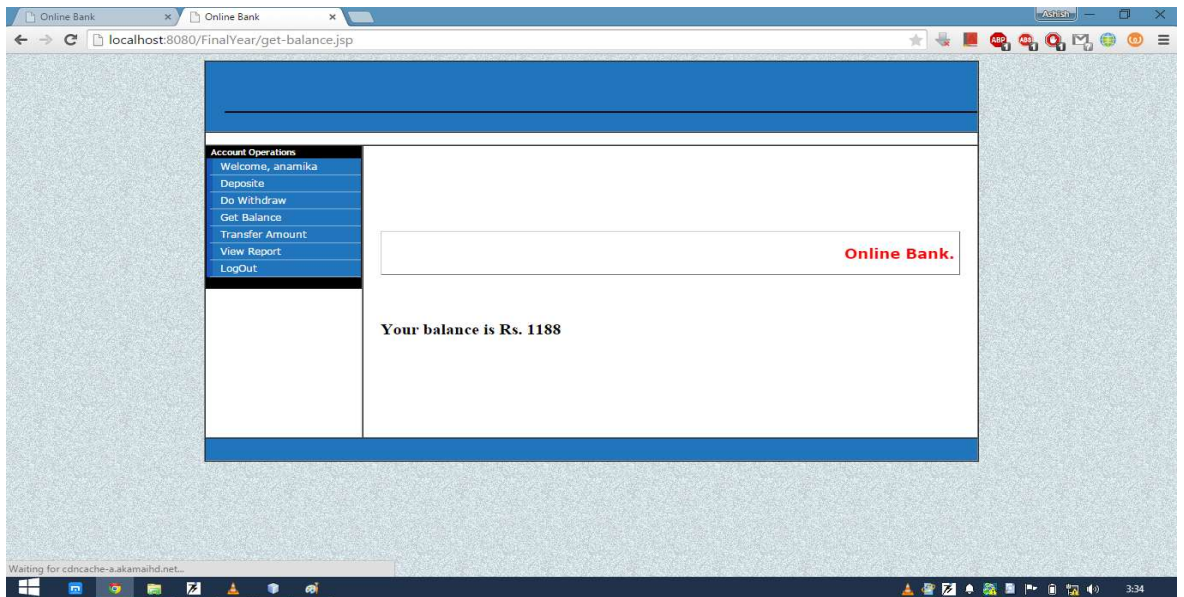


Figure 5

Clicking on the link view balance, your balance will be shown remaining in your account.

6.1.6 Transaction History

Following are the Repest of Your Account(=).

Name	Acc. No.	Operation	Amt	Balance	Date - Time
anamika	3029150	DEPOSITE	Rs.23000	Rs.23000	Thu Mar 18 23:40:41 IST 2010
anamika	3029150	WITHDRAW	Rs.2000	Rs.21000	Thu Mar 18 23:41:03 IST 2010
anamika	3029150	WITHDRAW	Rs.2000	Rs.19000	Thu Mar 18 23:43:45 IST 2010
anamika	3029150	DEPOSITE	Rs.2000	Rs.15000	Thu Mar 18 23:45:36 IST 2010
anamika	3029150	WITHDRAW	Rs.900	Rs.14100	Thu Mar 18 23:52:55 IST 2010
anamika	3029150	WITHDRAW	Rs.200	Rs.13900	Sat Mar 20 09:09:12 IST 2014
anamika	3029150	TRANSFERRED	Rs.676	Rs.13224	Mon Dec 01 18:30:32 IST 2014
anamika	3029150	TRANSFERRED	Rs.34	Rs.12188	Tue Dec 02 00:40:01 IST 2014
anamika	3029150	TRANSFERRED	Rs.45	Rs.12177	Tue Dec 02 00:41:30 IST 2014
anamika	3029150	TRANSFERRED	Rs.45	Rs.12177	Tue Dec 02 00:43:56 IST 2014
anamika	3029150	TRANSFERRED	Rs.45	Rs.12177	Tue Dec 02 00:43:56 IST 2014
anamika	3029150	TRANSFERRED	Rs.34	Rs.12188	Tue Dec 02 00:45:59 IST 2014
anamika	3029150	TRANSFERRED	Rs.100	Rs.12088	Tue Dec 02 00:47:05 IST 2014
anamika	3029150	TRANSFERRED	Rs.100	Rs.11988	Tue Dec 02 00:48:31 IST 2014
anamika	3029150	TRANSFERRED	Rs.1000	Rs.10988	Tue Dec 02 00:50:06 IST 2014
anamika	3029150	TRANSFERRED	Rs.2000	Rs.8988	Tue Dec 02 00:51:57 IST 2014

Figure 6

On clicking on the view report, all your transaction history will be generated from the database in tabular form.

6.2 Android App

6.2.1 Introduction

Android is one of an Open source platforms. It is created by Google and owned by Open Handset Alliance. It is designed with goal “accelerate innovation in mobile” As such android has taken over a field of mobile innovation. It is definitely free and open platform that differs hardware from software that runs on it. It results for much more devices be running the same application. Also it gives possibility of friendlier environment for developers and consumers.

Android it is complete software package for a mobile device. Since the beginning android team offers the developing kit (tool and frameworks) for creating mobile applications quick and easy as possible. In some cases you do not specially need an android phone but you are very welcome to have one. It can work right out of the box, but of course users can customize it for their particular needs. For manufactures it is ready and free solution for their devices. Except specific drivers android community provides everything else to create their devices. An Android app is a software application running on the Android platform. Because the Android platform is built for mobile devices, a typical Android app is designed for a smart phone or a tablet PC running on the Android OS.

The structure of android project is mostly the same, but also may differ depending on the project needs and IDE tool. When programmer uses ADT the project structure is generated automatically. Even further, ADT is also generating the ready- made application “Hello word”. The GUI version of ADT is the easiest way to create an Android project but the advance programmer can be also using the set of tools which can be run in terminal session. The terminal tool called “ant” can debug the Android project and create sample structure even if developer uses any other programming tool than Eclipse.

Basic Android project would have six directories such as: assets, bin, gen, libs, res, src. Also there are some files in project root directory such as: AndroidManifest.xml,

licenses, project.properties and other files. (Android Developers, 2013)

The most important for the developers are “res” and “src” directories. “res” directory contains all the current project resources such as: images, layouts, custom strings and other values. Images are stored in different directories depending on their size that application can automatically choose right image depending on the device specifications. Layouts are store in the “layout” folder. Basically layout file example would be an XML file which would specify elements and their position in current view. Also it is possible to code custom strings and colors so the parser can display them in application. It is recommended approach to store them in values directory rather than hard code to the actual code or XML file. It would make easy further development at translating the application to other languages.

The other important directory is “src”. This directory would usually consist of Java files which are adding functionality to the application. Then developer would create classes as separated Java files. If the class is created in GUI ADT environment the tool would generate automatically the statement in “AndroidManifest” file. If other programming environment is used, user must specify any new class activity by hard coding the “AndroidManifest” file. (Lee, 2012)

Android manifest file usually would be placed to the root directory of the project and state required version of android, needed permissions and all activities which are run within the application.

An emulator is hardware or software that enables one computer system (called the host) to behave like another computer system (called the guest). An emulator typically enables the host system to run software or use peripheral devices designed for the guest system

An Android emulator is an Android Virtual Device (AVD) that represents a specific Android device. You can use an Android emulator as a target platform to run and test your Android applications on your PC.

6.2.2 Snapshots:

6.2.2.1 Android Home Page

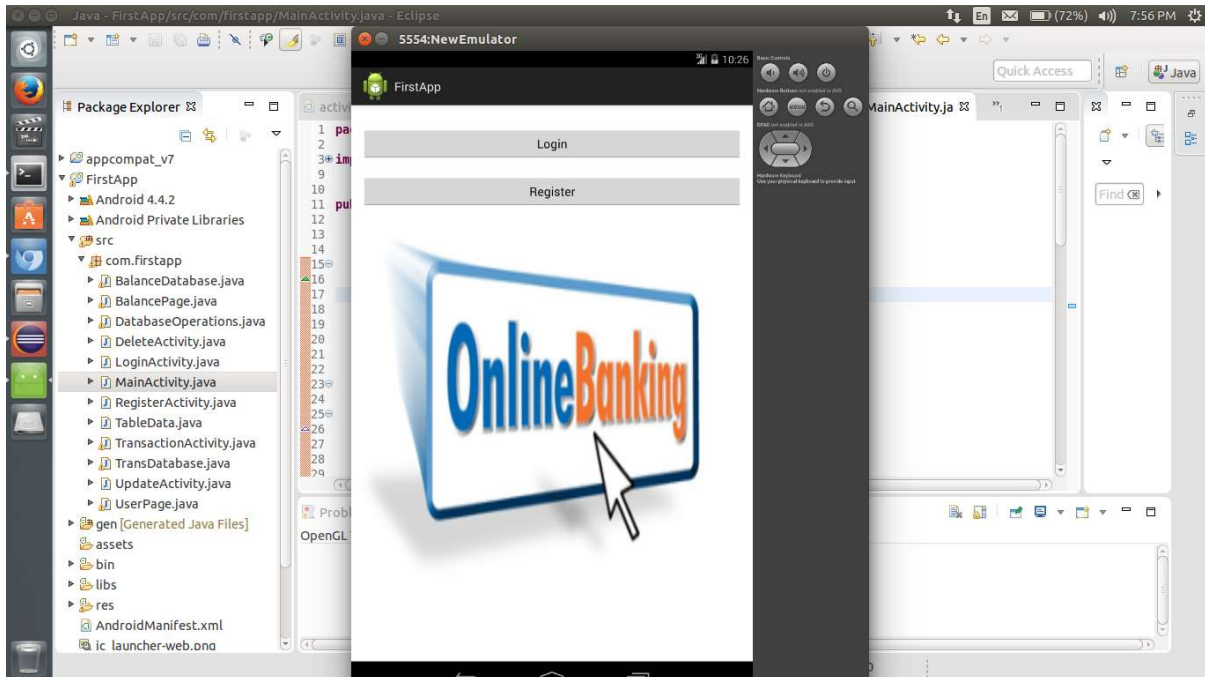


Figure 7

When you will run the app on emulator, above shown is the first interface that will be shown.

Xml file: MainActivity.xml

Java file: MainActivity.java

Emulator configurations:

6.2.2.2 Android Login Page

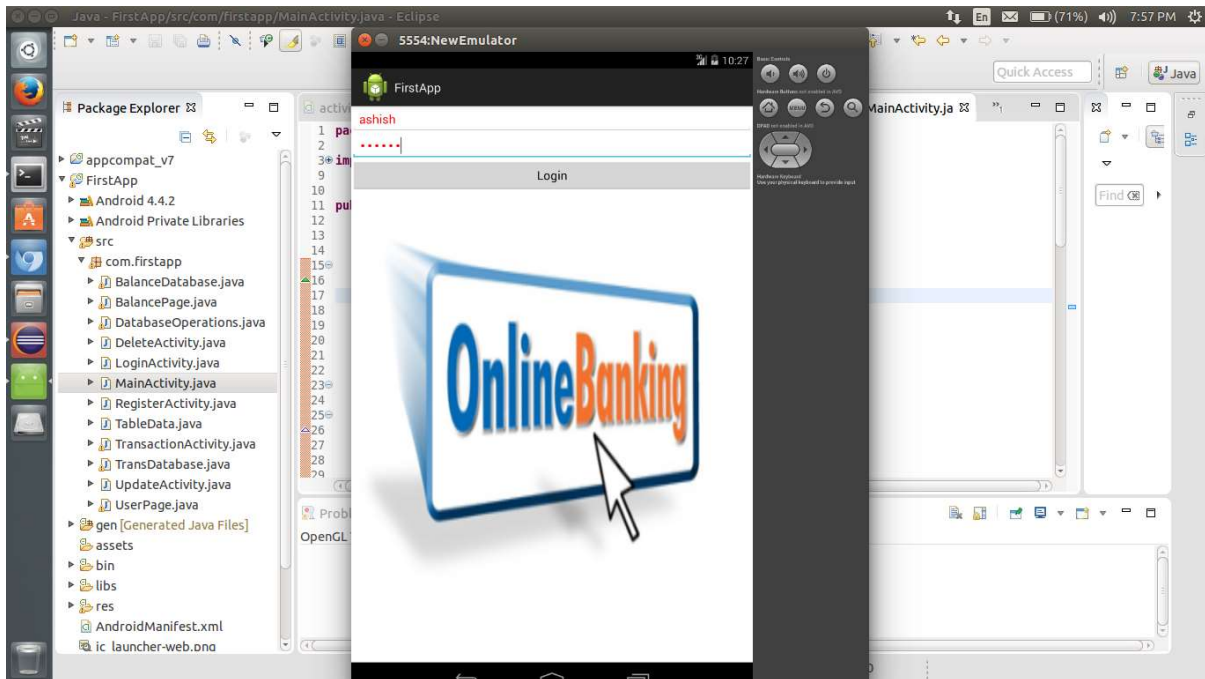


Figure 8

When you will click on the login button on the home page, login page will be shown where you have two fields to fill:

1. Username
2. Password

After filling this, your data will be checked from the database using LoginActivity.java file. If your username is not registered, then a message will be displayed showing you are not registered.

And if you are registered your data will be sent through the object of Intent and you will be sent to your profile page.

Xml file: LoginPage.xml

Java file: Login.java

6.2.2.3 Android Profile Page

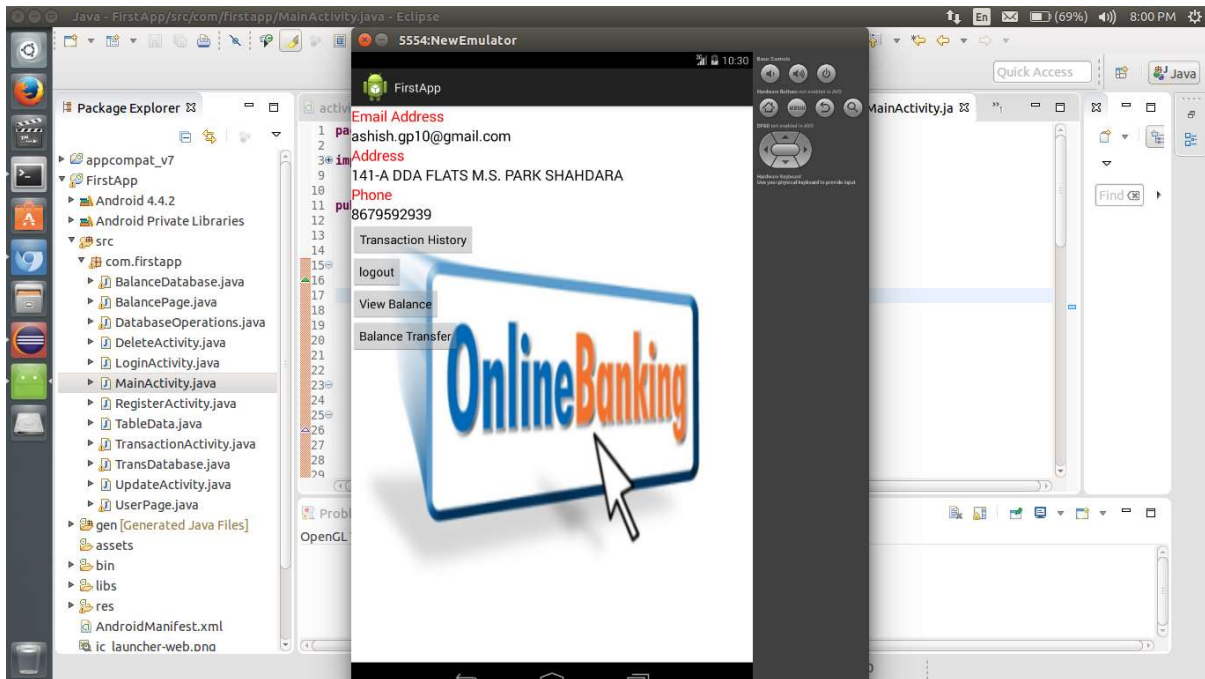


Figure 9

After you login to the app, you will be shown your basic information such as:

1. Email Address
2. Address
3. Phone

And other options to browse account such as:

1. Transaction history
2. View balance
3. Balance transfer
4. Logout

6.2.2.4 Android Balance Transfer Page

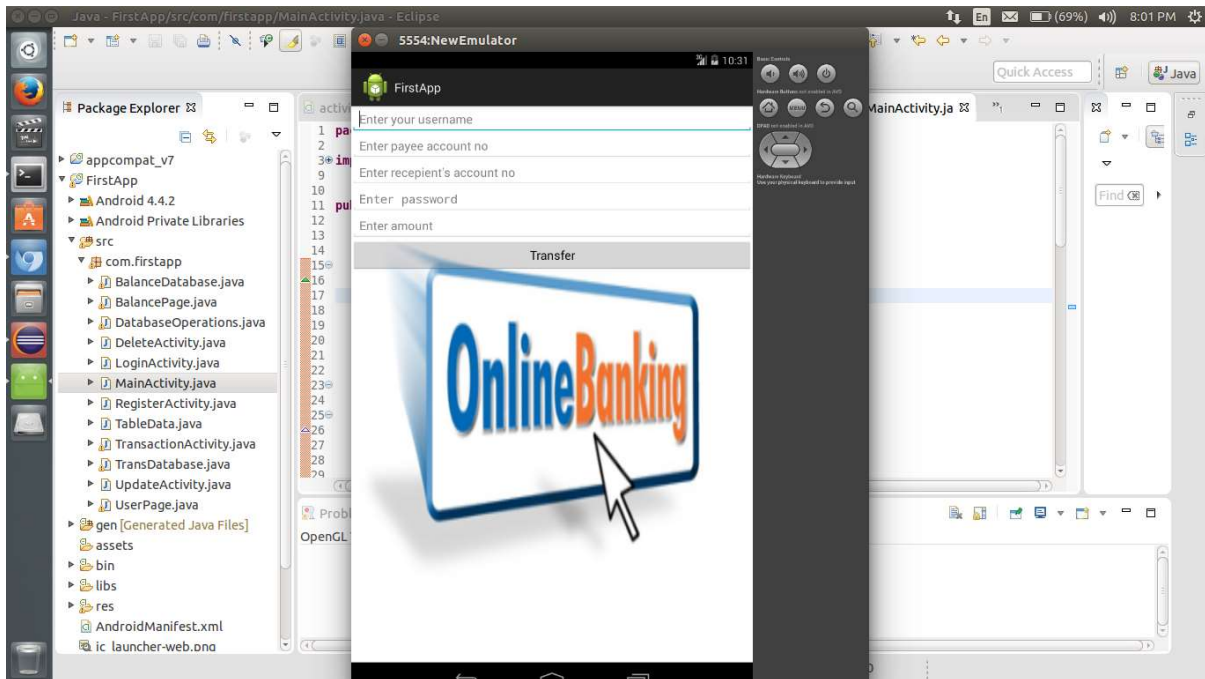


Figure 10

When you click on the balance transfer option, balance transfer page will be shown asking various fields such as:

1. Username
2. Payee account no.
3. Recipient account no
4. Password
5. Amount

All the fields will be checked from the database and then operations will be taken.

6.2.2.5 Android Balance View Page

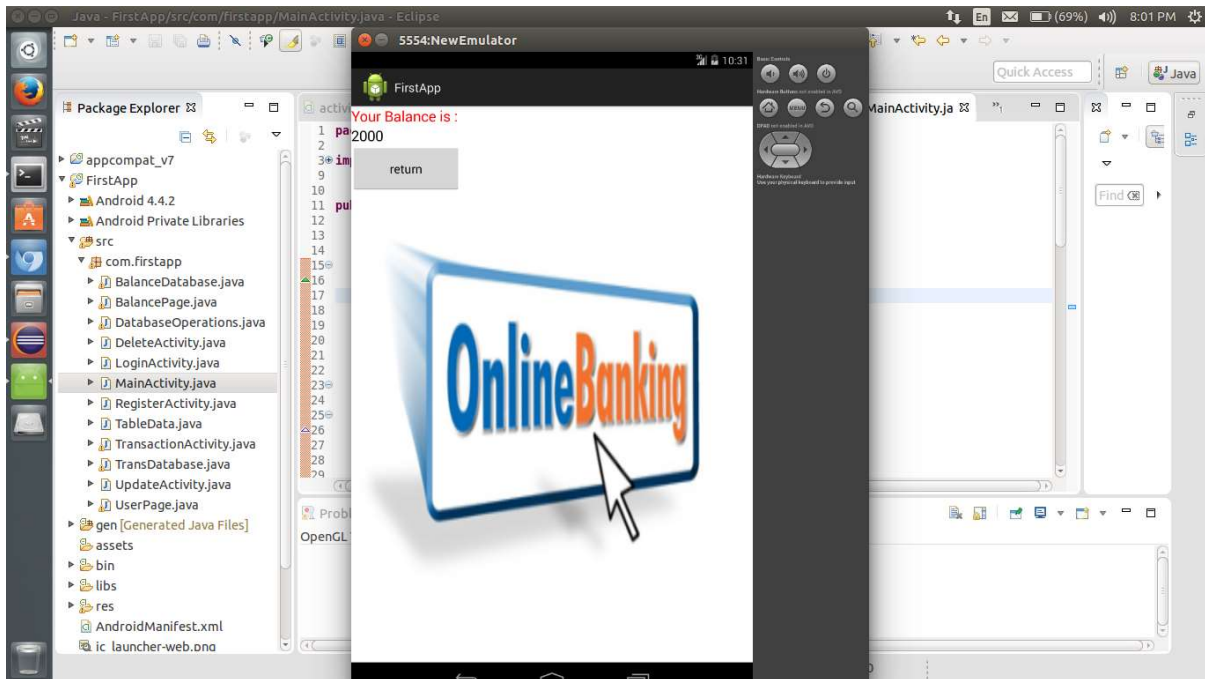


Figure 11

Using your username, the database will be searched and accordingly your balance will be displayed.

6.3 Code Snippets

6.3.1 Login page for Website

```
<%@page import="java.security.MessageDigest"%>

<%@ page contentType="text/html; charset=iso-8859-1" language="java"
pageEncoding="UTF-8" import="java.sql.*" errorPage="" %>

<%

//java Code

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

        Connection connection = DriverManager.getConnection("jdbc:odbc:db");

        Statement pst=connection.createStatement();

ResultSetrs=null;

String uname = request.getParameter("username");

String password = request.getParameter("password");

MessageDigestalg = MessageDigest.getInstance("MD5");

alg.reset();

alg.update(password.getBytes());

byte[] digest = alg.digest();

StringBufferhashedpasswd = new StringBuffer();

String hx;

for (inti=0;i<digest.length;i++){

        hx = Integer.toHexString(0xFF & digest[i]);

        //0x03 is equal to 0x3, but we need 0x03 for our md5sum
```



```

        if(hx.length() == 1){hx = "0" + hx;}

        hashedpasswd.append(hx);

    }

String password1 = hashedpasswd.toString();

String sql = "SELECT ID,uname, upass FROM user_details WHERE uname = '"+uname+"'
AND upass = '"+password1+"'";

//System.out.println(sql);

rs=pst.executeQuery(sql);

if(rs.next()){

    session.setAttribute("uname",uname);

        }

else

    {

response.sendRedirect("index.jsp");

    }

%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

<title>Online Bank</title>

<link href="css/menu.css" rel="stylesheet" type="text/css" />

```

```

<link href="css/main.css" rel="stylesheet" type="text/css" />

<style type="text/css">

<!--

html,body{

background-image: url(images/img.gif);

}

</style>

</head>

<body>

<table width="900" border="1" align="center" cellpadding="0" cellspacing="0" style="font-
weight:normal; background-color:#FFFFFF">

<tr>

<th colspan="3" scope="col" style="height:90px; background-color:#2175bc;"><object
classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version
=7,0,19,0" width="900" height="90">

<param name="movie" value="images/banks.swf" />

<param name="quality" value="high" />

<embed src="images/banks.swf" quality="high"
pluginspage="http://www.macromedia.com/go/getflashplayer" type="application/x-
shockwave-flash" width="900" height="90"></embed>

</object></th>

</tr>

<tr>

<td colspan="3">&nbsp;&nbsp;&nbsp;</td>

```

```

</tr>

<tr>

<td width="160" >

<div id="ddblueblockmenu">

<div class="menutitle">Account Operations</div>

<ul>

<li><a href="main.jsp">Welcome,&nbsp;<%= uname %></a></li>

<%

String sql1 = "SELECT * FROM acc_details WHERE uname='"+uname+"'";

try

{

rs=pst.executeQuery(sql1);

if(!rs.next())

{

out.print("<li>");

out.print("<a href='account.jsp'>");

out.print("Create Account");

out.print("</a>");

out.print("</li>");

}

}

catch(Exception e)

{

```

```

out.println(e);

    }

    %>

    <li><a href="deposit.jsp">Deposit</a></li>

    <li><a href="withdraw.jsp">Do Withdraw</a></li>

    <li><a href="get-balance.jsp">Get Balance</a></li>

    <li><a href="transfer.jsp">Transfer Amount</a></li>

    <li><a href="view-reports.jsp">View Report</a></li>

    <li><a href="logOff.jsp">LogOut</a></li>

</ul>

<div class="menutitle">&nbsp;</div>

</div>

<p>&nbsp;</p>

<p>&nbsp;</p>

<p>&nbsp;</p>

<p>&nbsp;</p>      </td>

<td colspan="2" style="padding:20px;">

    <div class="box1">

        <marquee><h2><font color="#FF0000">Online Bank.</font></h2></marquee>

    </div>

    <br/><p align="left" style="line-height:18px; padding:10px; font-weight:normal">The
central concept of the application is to allow the customer(s) to service virtually using the
Internet with out going to bank and allow customers to open new account, withdraw, deposit,
close and getting balance using this banking service.&nbsp;  The information pertaining to the

```

customers stores on an RDBMS at the server side (BANK). The Bank services the customers according to the customer’s intention and it updates and backups of each customer transaction accordingly.</p>

```
</td>

</tr>

<tr style="height:30px;">

<td colspan="3" style="background-color:#2175bc;">&nbsp;</td>

</tr>

</table>

</body>

</html>
```

6.3.2 Login page for Android App

```
packagecom.firstapp;

importandroid.app.Activity;

importandroid.content.Context;

importandroid.content.Intent;

importandroid.database.Cursor;

importandroid.os.Bundle;

importandroid.view.View;

importandroid.view.View.OnClickListener;

importandroid.widget.Button;

importandroid.widget.EditText;
```

```

import android.widget.Toast;

public class LoginActivity extends Activity {

    Button Login;

    EditText USERNAME, USERPASS;

    String username, userpass;

    Context CTX = this;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        // TODO Auto-generated method stub

        super.onCreate(savedInstanceState);

        setContentView(R.layout.login_layout);

        Login = (Button) findViewById(R.id.b_login);

        USERNAME = (EditText) findViewById(R.id.user_name);

        USERPASS = (EditText) findViewById(R.id.user_pass);

        Login.setOnClickListener(new OnClickListener() {

            @Override

            public void onClick(View arg0) {

                Bundle b = getIntent().getExtras();

                int status = b.getInt("status");

                if(status == 1)

                {

                    Toast.makeText(getApplicationContext(), "Please wait...",

                    Toast.LENGTH_LONG).show();

```

```

username = USERNAME.getText().toString();

userpass = USERPASS.getText().toString();

DatabaseOperations DOP = new DatabaseOperations(CTX);

Cursor CR = DOP.getInformation(DOP);

CR.moveToFirst();

booleanloginstatus = false;

String NAME = "";

do

{

if(username.equals(CR.getString(0))&& (userpass.equals(CR.getString(1))))

{

loginstatus = true;

NAME = CR.getString(0);

}

}while(CR.moveToNext());

if(loginstatus)

{

Toast.makeText(getBaseContext(), "Login Success---\n Welcome "+NAME,

Toast.LENGTH_LONG).show();

finish();

//Intent

intentAfterLogin=new Intent(getApplicationContext(),UserPage.class);

//startActivity(intentAfterLogin);

```

```

Intent i= new Intent(getApplicationContext(),UserPage.class);

i.putExtra("NAME",NAME );

startActivity(i);

}

else

{

Toast.makeText(getApplicationContext(), "Login Failed---- ",
Toast.LENGTH_LONG).show();

finish();

}

}

else if(status == 2)

{

Toast.makeText(getApplicationContext(), "Please wait...",
Toast.LENGTH_LONG).show();

username = USERNAME.getText().toString();

userpass = USERPASS.getText().toString();

DatabaseOperations DOP = new DatabaseOperations(CTX);

Cursor CR = DOP.getInformation(DOP);

CR.moveToFirst();

booleanloginstatus = false;

String NAME = "";

do

```



```

{
if(username.equals(CR.getString(0))&& (userpass.equals(CR.getString(1))))
{
loginstatus = true;
NAME = CR.getString(0);
}
}while(CR.moveToNext());

if(loginstatus)
{
Toast.makeText(getBaseContext(), "Login Success----\n Welcome "+NAME,
Toast.LENGTH_LONG).show();

Intent i = new Intent("update_filter");

Bundle BN = new Bundle();

BN.putString("user_name",NAME );

BN.putString("user_pass",userpass );

i.putExtras(BN);

startActivity(i);

finish();

Intent intentAfterLogin=new Intent(getApplicationContext(),UserPage.class);

startActivity(intentAfterLogin);

}

else

{

```

```

Toast.makeText(getBaseContext(), "Login Failed---- ",
Toast.LENGTH_LONG).show();

finish();

}

}

else if(status == 3)

{

Toast.makeText(getBaseContext(), "Please wait...",
Toast.LENGTH_LONG).show();

username = USERNAME.getText().toString();

userpass = USERPASS.getText().toString();

DatabaseOperations DOP = new DatabaseOperations(CTX);

Cursor CR = DOP.getInformation(DOP);

CR.moveToFirst();

booleanloginstatus = false;

String NAME = "";

do

{

if(username.equals(CR.getString(0))&& (userpass.equals(CR.getString(1))))

{

loginstatus = true;

NAME = CR.getString(0);

}

}

```

```

}while(CR.moveToNext());

if(loginstatus)

{

Toast.makeText(getBaseContext(), "Login Success----\n Welcome "+NAME,
Toast.LENGTH_LONG).show();

Intent i = new Intent("delete_filter");

Bundle B = new Bundle();

B.putString("user_name",NAME );

i.putExtras(B);

startActivity(i);

finish();

}

else

{

Toast.makeText(getBaseContext(), "Login Failed---- ",
Toast.LENGTH_LONG).show();

finish();

}

//Intent i = new Intent("delete_filter");

//startActivity(i);

}

}

});}}

```

6.3.3 MainActivity.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/online1"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.firstapp.MainActivity" >
    <Button
        android:id="@+id/Login"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/Reg"
        android:layout_alignParentTop="true"
        android:layout_marginTop="18dp"
        android:text="Login" />
    <Button
        android:id="@+id/Reg"
        android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
android:layout_below="@+id/Login"
android:layout_centerHorizontal="true"
android:layout_marginTop="23dp"
android:text="Register" />
</RelativeLayout>
```

6.4 Database Tables:

6.4.1 User_info

	Field Name	Data Type
🔑	ID	Number
	uname	Text
	upass	Text
	color	Text
	answer	Text
	address	Text
	email	Text
	mobile	Text
	image	Text

Table 1

It stores the general information about each user registered on the website. Each entry is uniquely identified by the primary key ID. Product names and brand names are stored. Also, for better retrieval and for facilitation of analysis process the products are classified at several stages into main group, sub group and type. This table also stores valuable attributes like uname, upass, answer, address, email, mobile etc. which are essential for exercising user profile.

6.4.2Balance

	Field Name	Data Type
🔑	uname	Text
	balance	Number

Table 2

6.4.3 Transaction_details

Field Name	Data Type
uname	Text
acc_no	Text
operation	Text
amt	Number
balance	Number
time1	Date/Time
id	AutoNumber

Table 3

It is the relationship table between main tables user_info, balance and acc_details. This table stores the each transaction made by the user. It also stores the uname to identify the customer.

6.4.4 Acc_details

Field Name	Data Type
uname	Text
acc_type	Text
details	Text
acc_no	Number

Table 4

Primary key is uname. This table stores the acc_type entered by the user and also the acc_no which is randomly generated .

6.5 Login process

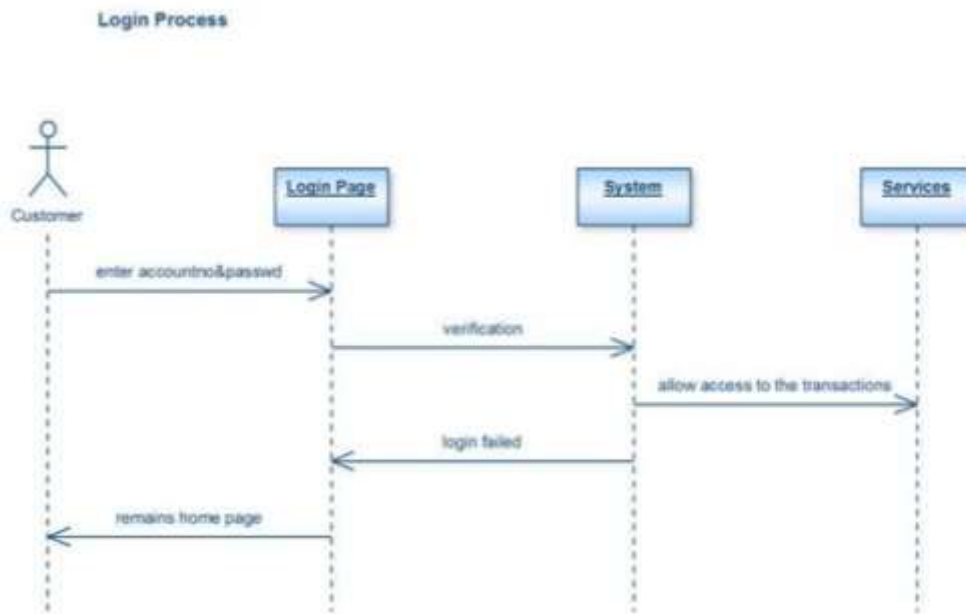


Figure 13

6.6 Registration process

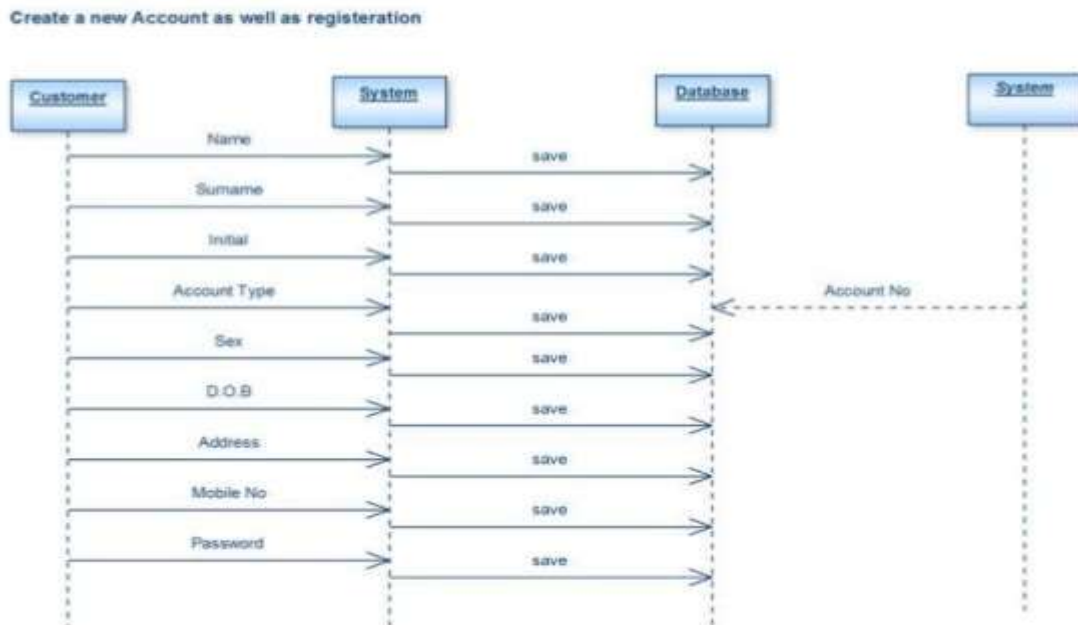


Figure 14

6.7 DFD level 0

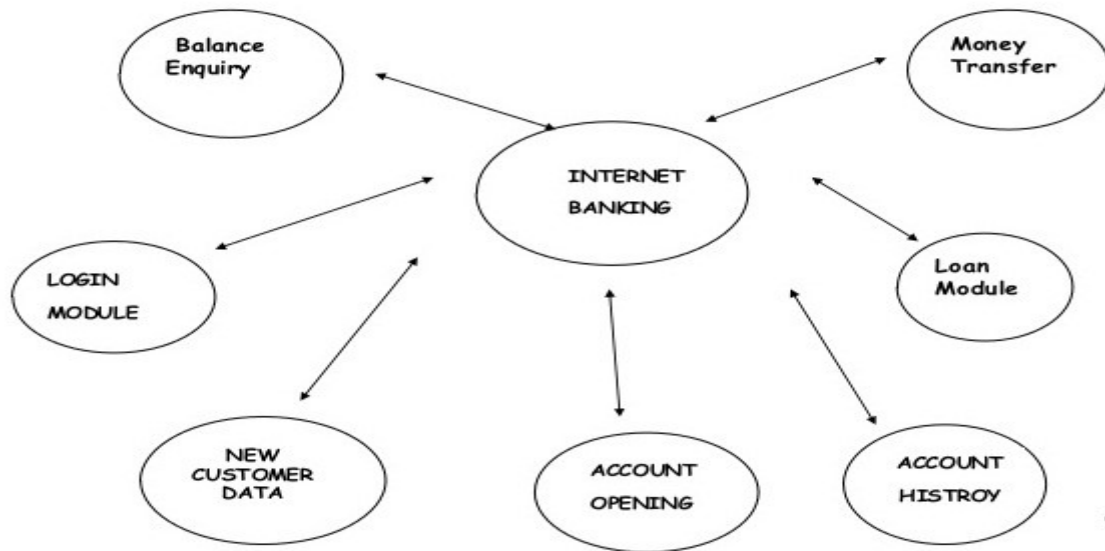


Figure 15

6.8 Use case

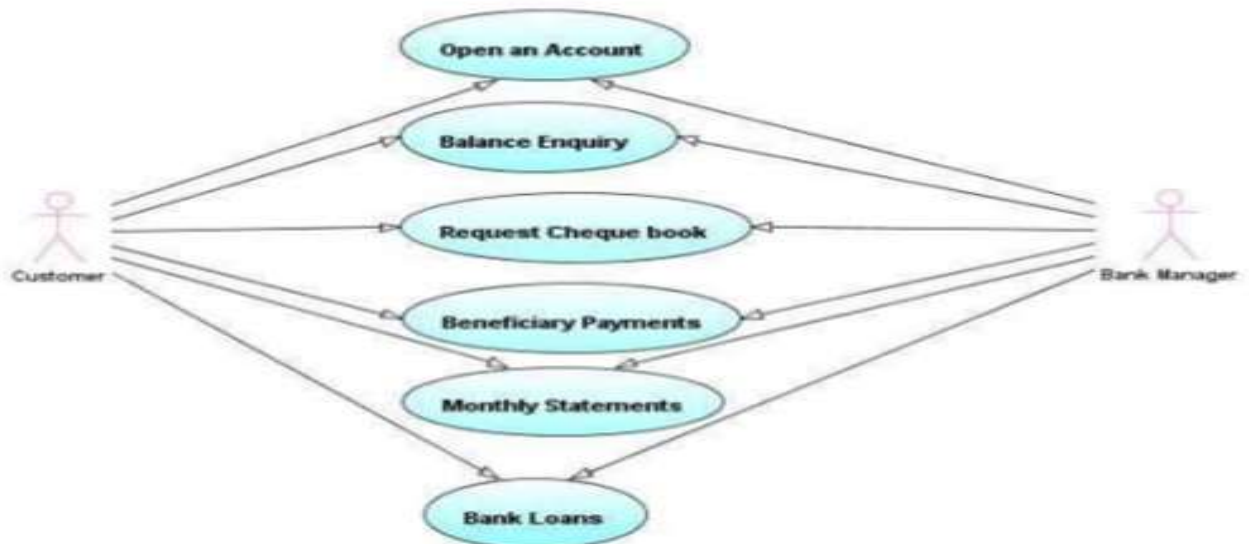


Figure 16

Conclusion

This project developed, incorporated all the activities involved in the browsing centre. It provides all necessary information to the management as well as the customer with the use of this system; the user can simply sit in front of the system and monitor all the activities without any physical movement of the file. Management can service the customer's request best in time.

The system provides quickly and valuable information. These modules have been integrated for effective use of the management for future forecasting and for the current need. Internet banking is changing the banking industry and is having the major effects on banking relationships. The net banking, thus, "now is more of a norm rather than an exception in many developed countries" due to the fact that it is the economical way of providing banking services.

Banking is now no longer confined to the traditional brick and mortar branches, where one has to be at the branch in person, to withdraw cash or deposit a cheque or request a statement of accounts. The application demonstrate the way to develop an online banking system by using interactive web client by using JSP, Servlet with more secure way to access.

This means the application server easily deployable and accessible.

References

- [1] D'Arcy, J., Hovav, A. and Galleta, D. (2009). User Awareness of Security Countermeasures and Its Impact on Information Systems Misuse: A Deterrence Approach. *Information Systems Research*, 20 (1), 79-98.
- [2] Green, I., Raz, T. and Zviran, M. (2007). Analysis Of Active Intrusion Prevention Data For Predicting Hostile Activity In Computer Networks. *Communications of the ACM*, 50 (4), 63-68.
- [3] Newhouse, K. (2007). Six Security Resolutions. *Credit Union Magazine*, 73 (2), 52. Oppliger, R. (2003). *Security Technologies for the World Wide Web*. 2nd ed. Boston: Artech House.
- [4] Whitman, M. (2003). *Enemy at the Gate: Threats to Information Security*. *Communications of the ACM*, 46 (8), 91-95.
- [5] P. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems", *CRYPTO, LNCS 1109*, pp.104-113, Springer, 1996.
- [6] D. J. Bernstein, "Cache-Timing Attacks on AES", <http://cr.yp.to/antiforgery/cachetiming-20050414.pdf>, April 2005.
- [7] J Bonneau, "Cache-Collision Timing Attacks Against AES", *Workshop on Cryptographic Hardware and Embedded Systems*, Yokohama, Japan, Oct. 2006.
- [8] H. Gilbert, T. Peyrin, "Super-Sbox Cryptanalysis, Improved Attacks for AES-like Permutations", *Cryptology ePrint Archive Report 2009/531*, November 2, 2009. <http://eprint.iacr.org/2009/531.pdf>.
- [9] "Online Banking: threats and Countermeasures Revised Version: 1.3", AhnLab, Inc., June, 2010.
- [10] "State of Data Security and Privacy in Indian banking Industry- DSCI-KPMG Survey-2010", *Data Security Council of India*, published in feb,2011.

