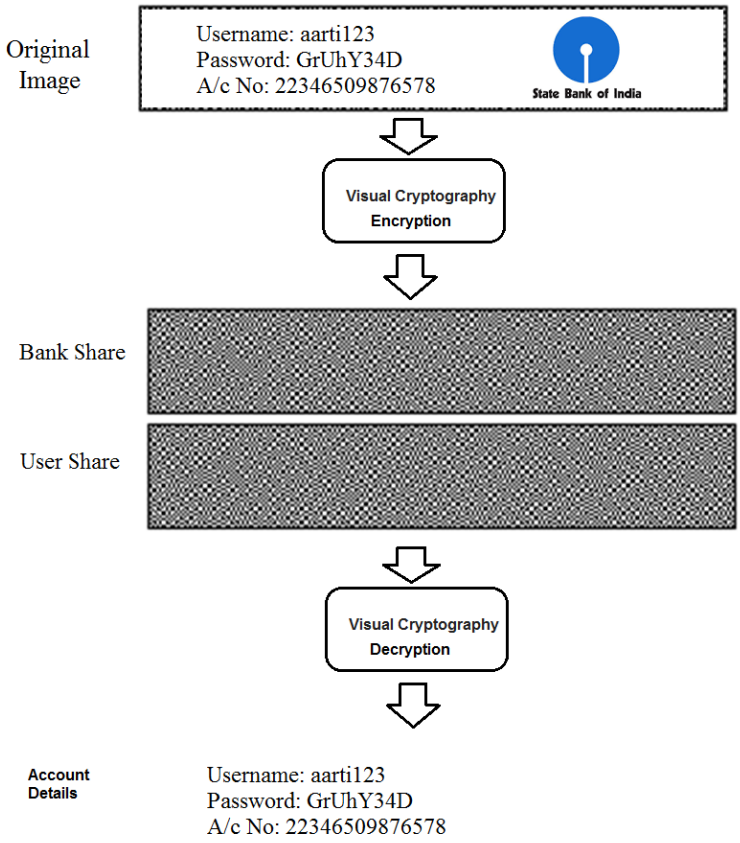
INTRODUCTION

We propose a new method for E-payment that provides unrivaled security by using cryptographic techniques visual cryptography, quantum cryptography and steganography. Visual cryptography hides the authentication details of customer by generating two shares for customer and bank respectively. Quantum cryptography secures the transmission of one time password. Steganography is used to combine the customer's share along with one time password in order to secure the transmission of customer's share to bank .Proposed method for E-shopping can be extended for other bank applications.



1.1 Existing System

E-shopping refers to process of buying commodities through web browser instead of using mortar stores. Due to wide variety of selection and higher convenience customers are magnetized to E shopping. Even though customer focuses on E-shopping than traditional shopping because of its advantage, security problems such as identity theft and phishing are the major concern for both customers and merchant. Identity theft is the practice of stealing another. As phishing refers to process of acquiring sensitive information by masquerading as a reputed entity.

we propose a new method for E-payment that provides unrivaled security by using cryptographic techniques visual cryptography, quantum cryptography and steganography. Visual cryptography hides the authentication details of customer by generating two shares for customer and bank respectively. Quantum cryptography secures the transmission of one time password.

1.2 Literature Survey

A brief survey of related work in the area of payment system is given in this section. An online payment system using visual cryptography and steganography is proposed in but the customer's share generated using visual cryptography is not secured during transmission, As a result of that an eavesdropper can masquerade as customer by hacking customer's share. In an off-line electronic cash scheme using partial blind signature is proposed but it is not meant for online payment. An electronic payment using quantum blind signature is proposed in but it requires account in same bank for both customer and merchant and it is not an online payment method.

1.2.1 Online Payment System using Steganography and Visual Cryptography

Identity theft and phishing are common threats of online shopping. Phishing is a method of stealing personal confidential information such as username, passwords, credit card details from victims. It is a social engineering technique used to deceive users. In this paper new method is proposed that uses text based steganography and visual cryptography. It represents new approach which will provide limited information for fund transfer. This method secures the customer's data and increases customer's confidence and prevents identity theft.

1.2.2 A novel anti phishing framework based on visual cryptography

In this paper we have proposed a new approach named as "A Novel Antiphishing framework based on visual cryptography" to solve the problem of phishing. Here an image based authentication using Visual Cryptography (vc) is used. The use of visual cryptography is explored to preserve the privacy of image captcha by decomposing the original image captcha into two shares that are stored in separate database servers such that the original image captcha can be revealed only when both are simultaneously available; the individual sheet images do not reveal the identity of the original image captcha. Once the original image captcha is revealed to the user it can be used as the password.

1.2.3 Visual CAPTCHAs for document authentication

In order to be applicable, it is essential that the document is represented as an image, recognizable to humans. Representing text documents as images has a series of drawbacks, like high memory and bandwidth requirements. In this paper we present two protocols for electronic exchange of visually authenticable digital documents, one requiring high bandwidth, but easy to implement in current infrastructure, and the other with low bandwidth requirements and preservation of the original document.

1.2.4 Secured Bank Authentication using Image Processing and Visual Cryptography

Because of tremendous realization and growth in the field of hacking it is not safe to rely on internet to store all the information. So in order to overcome this problem we are proposing an efficient algorithm for secured bank authentication. The algorithm mainly deals with Image Processing and Visual Cryptography. In this paper, signature of the applicant is processed in such a way that, signature is taken as input and is divided into different number of shares depending upon the banks scheme. One share is preserved in the bank database and all other shares are given to the applicant. The applicant need to provide his shares during every transaction and those shares are over lapped with the already existing bank shares and a check for authentication will be done by using correlation technique. If a higher correlation coefficient is achieved, then the authentication is succeeded.

1.2.5 An Efficient Iris Authentication Using Chaos Theory-based Cryptography for E-commerce Transactions

Credit card fraud is one of the crimes especially when it is used for web-based transaction. In this paper, a technical solution using Iris authentication technique is proposed for protecting identity theft in e-commerce transactions because Iris patterns are unique to an individual. Further, this research proposes authentication of e-commerce users by using Iris biometric technique as one of the most secure biometric algorithms. Therefore, this research proposes a web-based architecture which uses a combination of Image Processing and secure transmission of customers' Iris templates along with credit card details for decreasing credit card frauds over Internet.

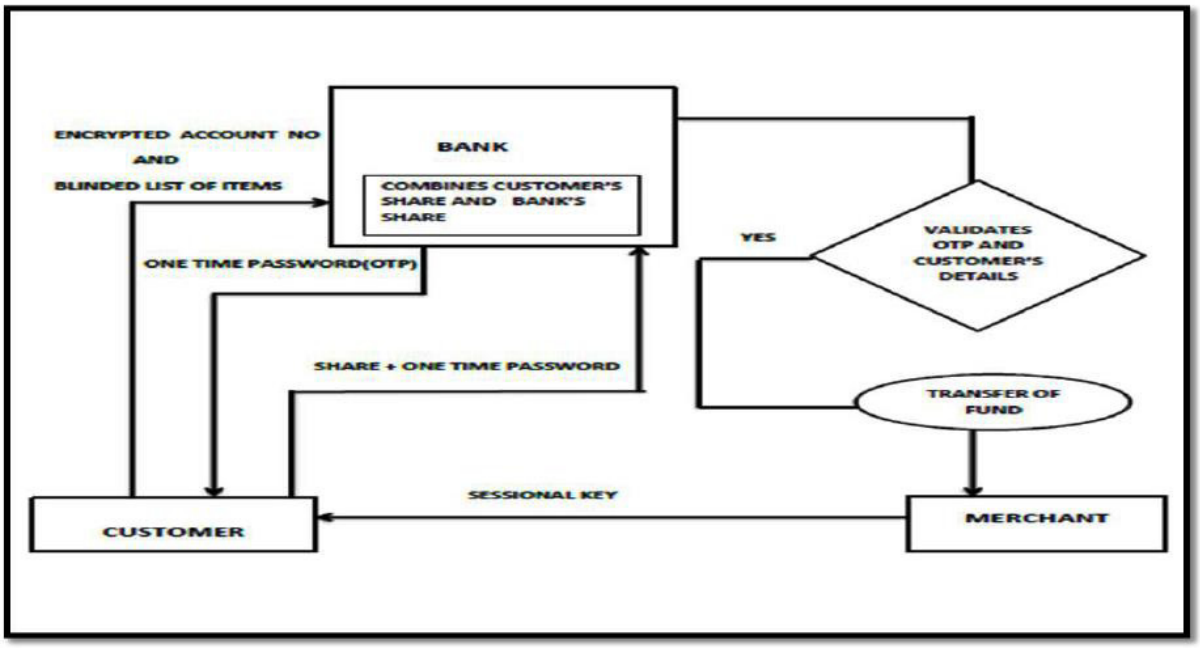
1.3Problem Definition

E-shopping refers to process of buying commodities through web browser instead of using mortar stores. Due to wide variety of selection and higher convenience customers are magnetized to E shopping. Even though customer focuses on E-shopping than traditional shopping because of its advantage, security problems such as identity theft and phishing are the major concern for both customers and merchant. Identity theft is the practice of stealing another person's identity for gaining access to his resource whereas phishing refers to process of acquiring sensitive information by masquerading as a reputed entity.

We propose a new method for E-payment that provides unrivaled security by using cryptographic techniques visual cryptography, quantum cryptography and steganography. Visual cryptography hides the authentication details of customer by generating two shares for customer and bank respectively. Quantum cryptography secures the transmission of one time password. Steganography is used to combine the customer's share along with one time password in order to secure the transmission of customer's

share to bank .Proposed method for E-shopping can be extended for other bank applications.

1.4Proposed System



- In proposed method snapshot of text containing customer's account number and debit and credit card information is taken
- From the snapshot image two shares are generated using visual cryptography
- One share will be in the hand of customer and other one will be in database of bank
- Merchant and customer agrees on a sessional key at the start of E-shopping
- Customer do shopping and start for payment
- Bank generates an one time password and securely transfers it to customer using quantum cryptography
- After receiving one time password ,image steganography is performed by taking customer's share as cover image and hidden information as one time password and stego image is passed to bank

- Bank extracts embedded one time password so that share and one time password gets separated.
- Then Bank combines customer's share with bank's share and obtains account number and credit card details.
- Finally bank validates the one time password and credit card details and if both verification gets right fund is transferred to merchant account number.

MODULES

2.1 Cryptography:

- Our aim is to implement a new method for E-Payment that provides security by using cryptographic techniques:
  - Visual cryptography
  - Quantum cryptography and
  - Steganography
- ***Visual cryptography*** hides the authentication details of customer by generating two shares for customer and bank respectively.
- ***Quantum cryptography*** secures the transmission of one time password.
- ***Steganography*** is used to combine the customer's share along with one time password in order to secure the transmission of customer's share to bank .

SOFTWARE REQUIREMENT SPECIFICATION

3.1 Role Of SRS

The SRS is used for developer and testers of this system who wanted to build and test the system of carpooling.

Developers will be having the following responsibility:

1. To design this system
2. To develop this system
3. To do modular unit testing of this system
4. To provide the details documentation of system
5. To write the code and do bug fixing.
6. To test the system
7. To test the code

3.2 Scope

Visual cryptography hides the details of customer by generating shares whereas Quantum cryptography secures the transmission of one time password .Image steganography embeds the share with one time password which results in secure transmission of share to bank. Proposed approach guarantees unconditional security than traditional E-payment system by using two important cryptographic techniques.

3.3 Software Requirements

Name of system/Application software	Description
Java	JDK 7
Eclipse	3.4
Tomcat 7	
MySql 6	

3.4 Hardware Requirements

Peripheral/Device	Configuration
Processor Intel core2Duo	Pentium-4/i3
RAM	2GB
Hard Disk	100GB

Table 4.1: Hardware Requirements

FEASIBILITY STUDY OF PROJECT

4.1 E-Payment Using Visual Cryptopgraphy:

Given a failure case viz. Q, Invalid image selection & image get distorted; we devise an algorithm for this problem as follows:

For a Problem P1 to be NP-Hard, Satisfiability problem (SAT) must be reducible to P1;

SAT ≤ P ;

Let the propositional formula be: G = X1^X2

Where

X1: True if Invalid image selection

X2: True if image get distorted

Algo sati()

```
{
        For i: 1 to 2
                x_i = Choice(True,False);
  if G(x_1,x_2) then
  Success();
          else
                failure();
}
```

Therefore, since the problem becomes a decision problem, it is **NP.**

4.2 Satisfiability and Reducibility:

3 SAT problem is NP Complete. The system can be reduced to 3SAT problem. A 3SAT problem takes a Boolean formula S that is in CNF in which each clause has exactly three literals. 3SAT is a restricted form of CNF-SAT problem.

x_1 – Image Capturing Module
x_2 - Crate Shares Module
x_3 – OCR

$S = (x_1 \wedge x_2 \wedge x_3)$

```
Algo sat()
{
For i= 1 to 3
 Xi=Choice(true, false)
If(S(x1,x2,x3)= true)
  Success()
Else
  Failure()
}
```

As it is polynomial time. It is NP-Complete.

TECHNOLOGY USED

5.1. Java

Java is well suited to help software developers meet challenges and seize opportunities presented by the emerging computing environment, because Java was designed for networks. Its suitability for networked environments is inherent in its architecture, which enables secure, robust, platform- independent programs to be delivered across networks and run on a great variety of computers and devices.

The Architecture

Java's architecture arises out of four distinct but interrelated technologies:

- the Java programming language
- the Java class file format
- the Java Application Programming Interface
- the Java virtual machine

When you write and run a Java program, you are tapping the power of these four technologies. You express the program in source files written in the Java programming language, compile the source to Java class files, and run the class files on a Java virtual machine. When you write your program, you access system resources (such as I/O, for example) by calling methods in the classes that implement the Java Application Programming Interface, or Java API. As your program runs, it fulfills your program's Java API calls by invoking methods in class files that implement the Java API. You can see the relationship between these four parts in Figure 5-1.

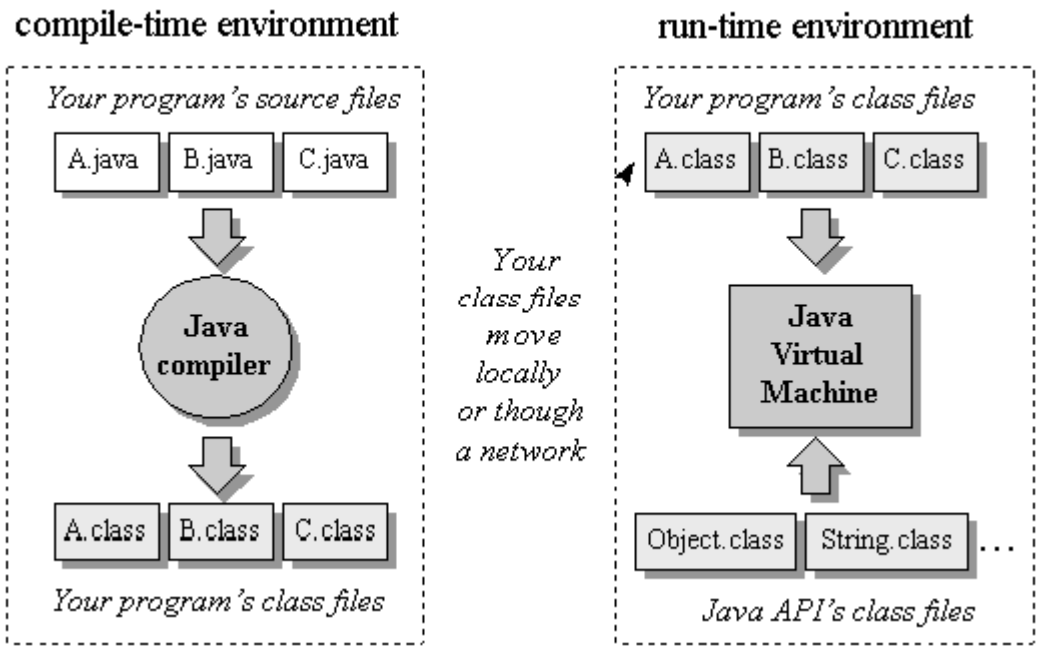


Figure 5-1. The Java programming environment.

Together, the Java virtual machine and Java API form a "platform" for which all Java programs are compiled. In addition to being called the *Java runtime system*, the combination of the Java virtual machine and Java API is called the *Java Platform* (or, starting with version 1.2, the *Java 2 Platform*). Java programs can run on many different kinds of computers because the Java Platform can itself be implemented in software. As you can see in Figure 5-2, a Java program can run anywhere the Java Platform is present.

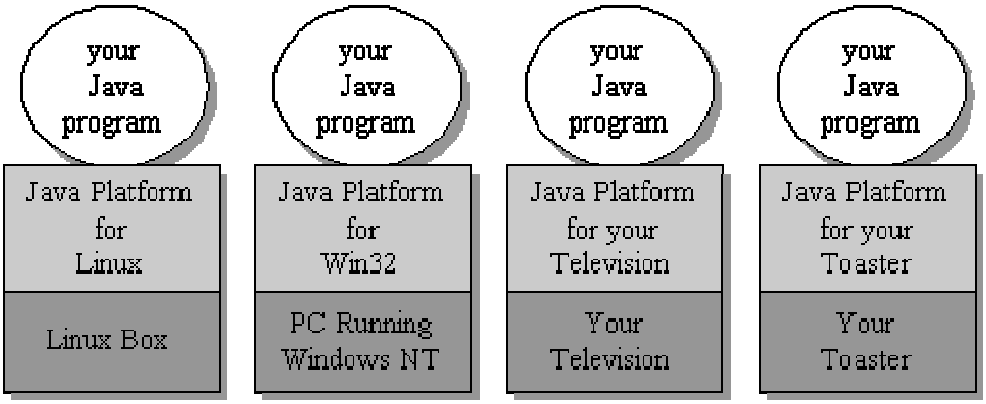


Figure 5-2. Java programs run on top of the Java Platform.

The Java Virtual Machine

At the heart of Java's network-orientation is the Java virtual machine, which supports all three prongs of Java's network-oriented architecture: platform independence, security, and network-mobility.

The Java virtual machine is an abstract computer. Its specification defines certain features every Java virtual machine must have, but leaves many choices to the designers of each implementation. For example, although all Java virtual machines must be able to execute Java bytecodes, they may use any technique to execute them. Also, the specification is flexible enough to allow a Java virtual machine to be implemented either completely in software or to varying degrees in hardware. The flexible nature of the Java virtual machine's specification enables it to be implemented on a wide variety of computers and devices.

A Java virtual machine's main job is to load class files and execute the bytecodes they contain. As you can see in Figure 1-3, the Java virtual machine contains a *class loader*, which loads class files from both the program and the Java API. Only those class files from the Java API that are actually needed by a running program are loaded into the virtual machine. The bytecodes are executed in an *execution engine*.

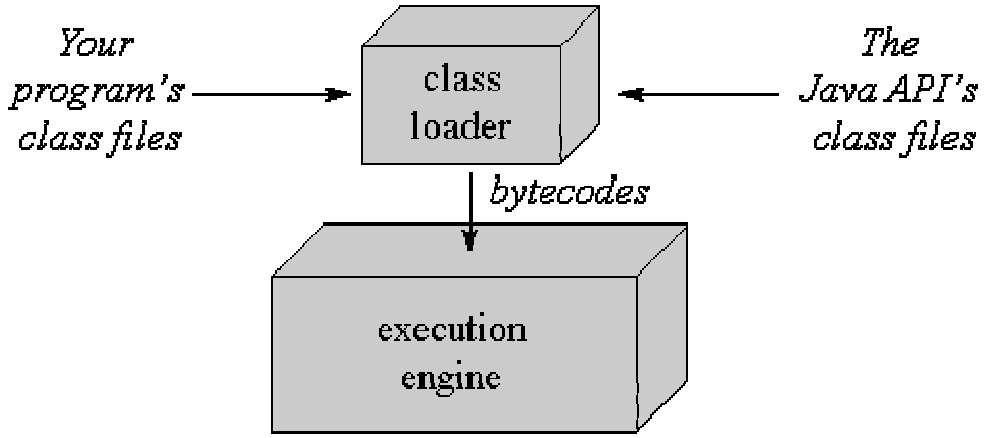


Figure 5-3. A basic block diagram of the Java virtual machine.

The execution engine is one part of the virtual machine that can vary in different implementations. On a Java virtual machine implemented in software, the simplest kind of execution engine just interprets the byte codes one at a time. Another kind of execution engine, one that is faster but requires more memory, is a *just-in-time compiler*. In this scheme, the byte codes of a method are compiled to native machine code the first time the method is invoked. The native machine code for the method is then cached, so it can be re-used the next time that same method is invoked. A third type of execution engine is an *adaptive optimizer*. In this approach, the virtual machine starts by interpreting byte codes, but monitors the activity of the running program and identifies the most heavily used areas of code. As the program runs, the virtual machine compiles to native and optimizes just these heavily used areas. The rest of the of code, which is not heavily used, remain as byte codes which the virtual machine continues to interpret. This adaptive optimization approach enables a Java virtual machine to spend typically 80 to 90% of its time executing highly optimized native code, while requiring it to compile and optimize only the 10 to 20% of the code that really matters to performance. Lastly, on a Java virtual machine built on top of a chip that executes Java byte codes natively, the execution engine is actually embedded in the chip.

Sometimes the Java virtual machine is called the *Java interpreter*; however, given the various ways in which byte codes can be executed, this term can be misleading. While "Java interpreter" is a reasonable name for a Java virtual machine that interprets byte codes, virtual machines also use other techniques (such as just-in-time compiling) to execute bytecodes. Therefore, although all Java interpreters are Java virtual machines, not all Java virtual machines are Java interpreters.

When running on a Java virtual machine that is implemented in software on top of a host operating system, a Java program interacts with the host by invoking *native methods*. In Java, there are two kinds of methods: Java and native. A Java method is written in the Java language, compiled to byte codes, and stored in class files. A native method is written in some other language, such as C, C++, or assembly, and compiled to the native machine code of a particular processor. Native methods are stored in a dynamically linked library whose exact form is platform specific. While Java methods are platform independent, native methods are not. When a running Java program calls a native method, the virtual machine loads the dynamic library that contains the native method and invokes it. As you can see in Figure 1-4, native methods are the connection between a Java program and an underlying host operating system.

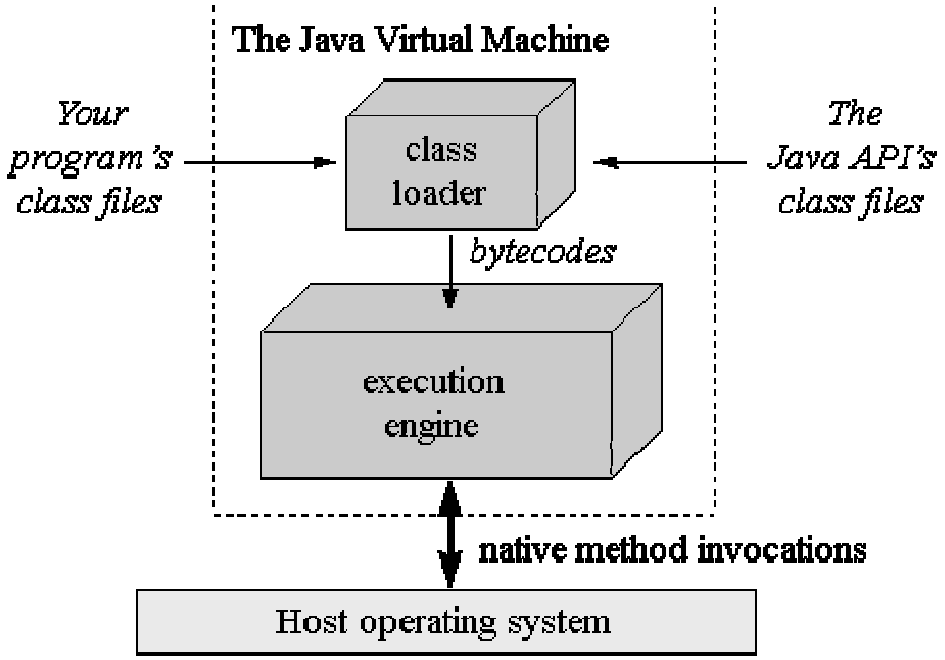


Figure 5-4. A Java virtual machine implemented in software on top of a host operating system.

You can use native methods to give your Java programs direct access to the resources of the underlying operating system. Their use, however, will render your program platform specific, because the dynamic libraries containing the native methods are platform specific. In addition, the use of native methods may render your program specific to a particular implementation of the Java Platform. One native method interface--the *Java Native Interface*, or *JNI*--enables native methods to work with any Java Platform implementation on a particular host computer. Vendors of the Java Platform, however, are not necessarily required to support JNI. They may provide their own proprietary native method interfaces in addition to (or depending on their contract, in place of) JNI.

Java gives you a choice. If you want to access resources of a particular host that are unavailable through the Java API, you can write a platform-specific Java program that calls native methods. If you want to keep your program platform independent, however,

you must access the system resources of the underlying operating system only through the Java API.

The Java API

The Java API helps make Java suitable for networks through its support for platform independence and security. The Java API is set of runtime libraries that give you a standard way to access the system resources of a host computer. When you write a Java program, you assume the class files of the Java API will be available at any Java virtual machine that may ever have the privilege of running your program. This is a relatively safe assumption because the Java virtual machine and the class files for the Java API are the required components of any implementation of the Java Platform. When you run a Java program, the virtual machine loads the Java API class files that are referred to by your program's class files. The combination of all loaded class files (from your program and from the Java API) and any loaded dynamic libraries (containing native methods) constitute the full program executed by the Java virtual machine.

The class files of the Java API are inherently specific to the host platform. The API's functionality must be implemented expressly for a particular platform before that platform can host Java programs. To access the native resources of the host, the Java API calls native methods. As you can see in Figure 5-6, the class files of the Java API invoke native methods so your Java program doesn't have to. In this manner, the Java API's class files provide a Java program with a standard, platform-independent interface to the underlying host. To the Java program, the Java API looks the same and behaves predictably no matter what platform happens to be underneath. Precisely because the Java virtual machine and Java API are implemented specifically for each particular host platform, Java programs themselves can be platform independent.

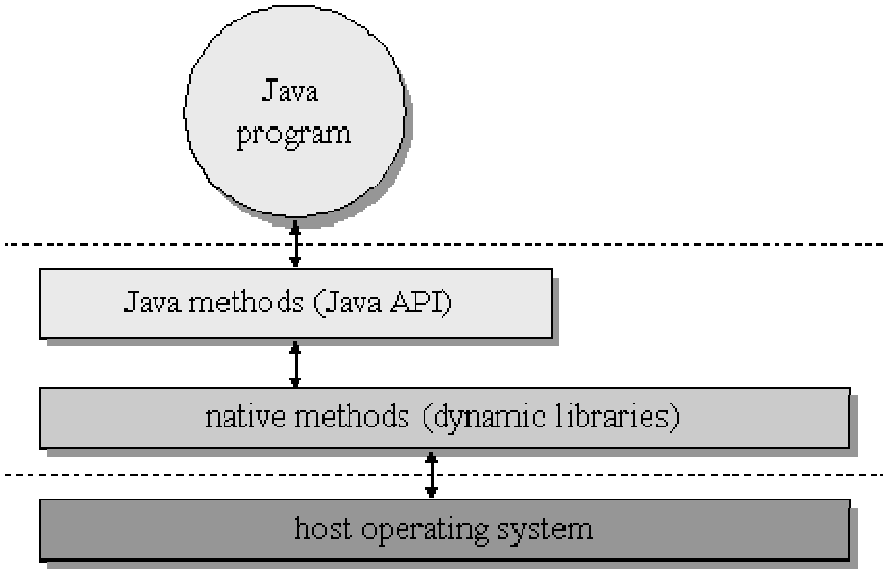


Figure 5-6. A platform-independent Java program.

The internal design of the Java API is also geared towards platform independence. For example, the graphical user interface libraries of the Java API, the Abstract Windows Toolkit (or AWT) and Swing, are designed to facilitate the creation of user interfaces that work on all platforms. Creating platform- independent user interfaces is inherently difficult, given that the native look and feel of user interfaces vary greatly from one platform to another. The AWT library's architecture does not coerce implementations of the Java API to give Java programs a user interface that looks exactly the same everywhere. Instead, it encourages implementations to adopt the look and feel of the underlying platform. The Swing library offers even more flexibility -- enabling the look and feel to be chosen by the programmer. Also, because the size of fonts, buttons, and other user interface components will vary from platform to platform, the AWT and Swing include *layout managers* to position the elements of a window or dialog box at run- time. Rather than forcing you to indicate exact X and Y coordinates for the various elements that constitute, say, a dialog box, the layout manager positions them when your dialog box is displayed. With the aim of making the dialog look its best on each platform, the layout manager will very likely position the dialog box elements slightly differently on

different platforms. In these ways and many others, the internal architecture of the Java API is aimed at facilitating the platform independence of the Java programs that use it.

In addition to facilitating platform independence, the Java API contributes to Java's security model. The methods of the Java API, before they perform any action that could potentially be harmful (such as writing to the local disk), check for permission. In Java releases prior to 1.2, the methods of the Java API checked permission by querying the *security manager*. The security manager is a special object that defines a custom security policy for the application. A security manager could, for example, forbid access to the local disk. If the application requested a local disk write by invoking a method from the pre-1.2 Java API, that method would first check with the security manager. Upon learning from the security manager that disk access is forbidden, the Java API would refuse to perform the write. In Java 1.2, the job of the security manager was taken over by the *access controller*, a class that performs stack inspection to determine whether the operation should be allowed. (For backwards compatibility, the security manager still exists in Java 1.2.) By enforcing the security policy established by the security manager and access controller, the Java API helps to establish a safe environment in which you can run potentially unsafe code.

5.2 Database

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds. Other kinds of data stores can be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those types of systems. So nowadays, we use relational database management systems (RDBMS) to store and manage huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as foreign keys.

5.2.1 Database Connectivity

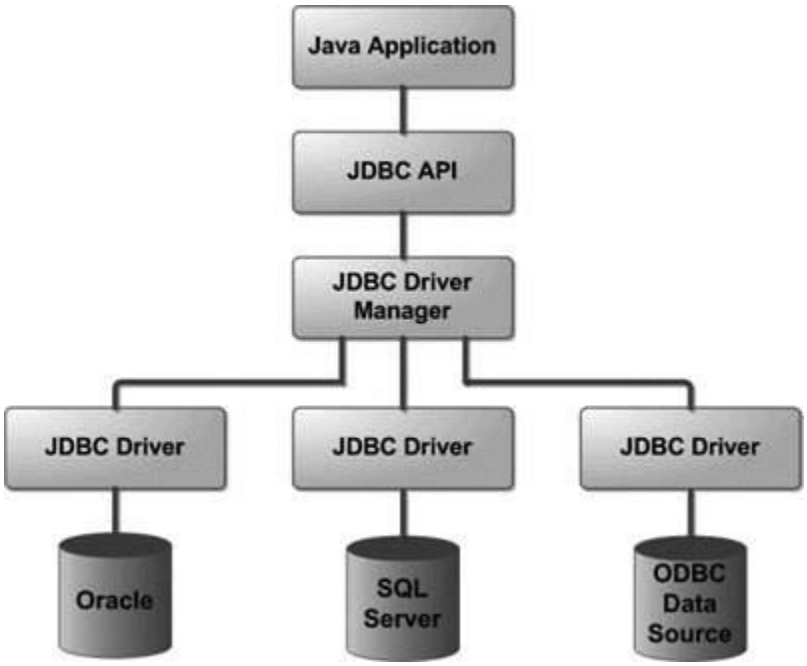
JDBC stands for **J**ava **D**ata**b**ase **C**onnectivity, which is a standard Java API for database-independent connectivity between the Java programming language and a wide range of databases.

The JDBC library includes APIs for each of the tasks mentioned below that are commonly associated with database usage.

- Making a connection to a database.
- Creating SQL or MySQL statements.
- Executing SQL or MySQL queries in the database.
- Viewing & Modifying the resulting records.

JDBC API: This provides the application-to-JDBC Manager connection.

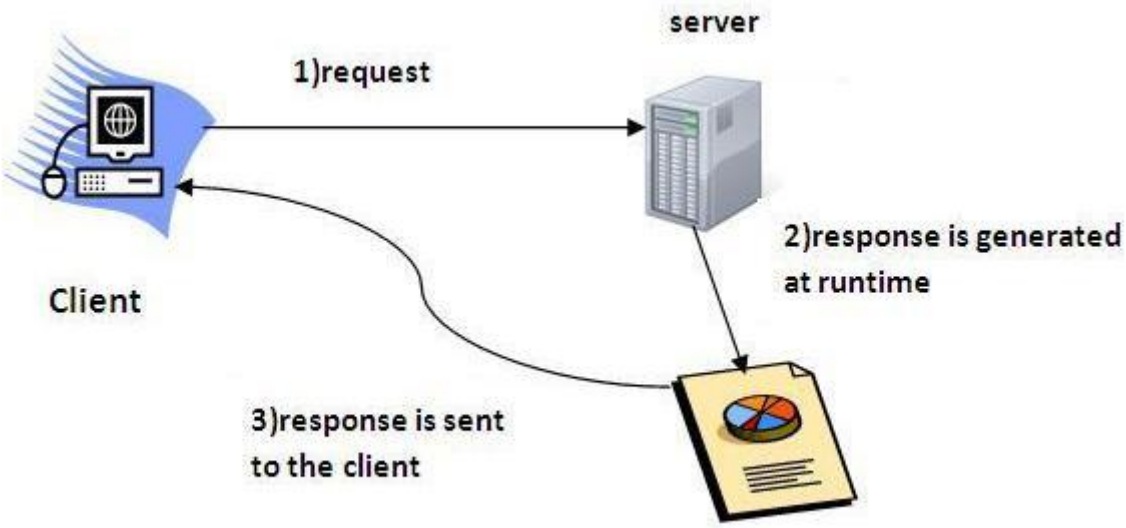
JDBC Driver API: This supports the JDBC Manager-to-Driver Connection.



5.3 Servlet& JSP

Servlet can be described in many ways, depending on the context.

- Servlet is a technology i.e. used to create web application.
- Servlet is an API that provides many interfaces and classes including documentations.
- Servlet is an interface that must be implemented for creating any servlet.
- Servlet is a class that extends the capabilities of the servers and responds to the incoming request. It can respond to any type of requests.
- Servlet is a web component that is deployed on the server to create dynamic web page.



How Servlet works?

When we run our project there are mane web pages like index, home, login, registration etc.But how would server knows which is the first file to run .When we run our project the first file is web.xml. In that we mention in welcome list File which is the first page to run.

The web container maintains the life cycle of a servlet instance. Let's see the life cycle of the servlet:

1. Servlet class is loaded.
2. Servlet instance is created.
3. Init method is invoked.
4. Service method is invoked.
5. Destroy method is invoked.

5.4 JSP

JavaServer Pages (JSP) is a technology for developing web pages that support dynamic content which helps developers insert java code in HTML pages by making use of special JSP tags, most of which start with <% and end with %>.

A JavaServer Pages component is a type of Java servlet that is designed to fulfill the role of a user interface for a Java web application. Web developers write JSPs as text files that combine HTML or XHTML code, XML elements, and embedded JSP actions and commands.Using JSP, you can collect input from users through web page forms, present records from a database or another source, and create web pages dynamically.

JSP tags can be used for a variety of purposes, such as retrieving information from a database or registering user preferences, accessing JavaBeans components, passing control between pages and sharing information between requests, pages etc. JavaServer Pages often serve the same purpose as programs implemented using the Common Gateway Interface (CGI). But JSP offer several advantages in comparison with the CGI. Performance is significantly better because JSP allows embedding Dynamic Elements in HTML Pages itself instead of having a separate CGI files. JSP are always compiled before it's processed by the server unlike CGI/Perl which requires the server to load an interpreter and the target script each time the page is requested. JavaServer Pages are

built on top of the Java Servlets API, so like Servlets; JSP also has access to all the powerful Enterprise Java APIs, including JDBC, JNDI, EJB, JAXP etc. JSP pages can be used in combination with servlets that handle the business logic, the model supported by Java servlet template engines.

OUR JSP Pages:

- index.jsp
- admin.jsp
- signup.jsp
- showReport.jsp
- logout.jsp

5.5 HTML

HTML5 is the latest and most enhanced version of HTML. Technically, HTML is not a programming language, but rather a markup language. HTML5 is the next major revision of the HTML standard superseding HTML 4.01, XHTML 1.0, and XHTML 1.1. HTML5 is a standard for structuring and presenting content on the World Wide Web.

HTML5 is cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).

The new standard incorporates features like video playback and drag-and-drop that have been previously dependent on third-party browser plug-ins such as Adobe Flash, Microsoft Silverlight, and Google Gears.

New Features

HTML5 introduces a number of new elements and attributes that helps in building a modern website. Following are great features introduced in HTML5.

- **New Semantic Elements** − These are like <header>, <footer>, and <section>.
- **Forms 2.0** − Improvements to HTML web forms where new attributes have been introduced for <input> tag.
- **Persistent Local Storage** − To achieve without resorting to third-party plugins.
- **WebSocket** − A a next-generation bidirectional communication technology for web applications.
- **Server-Sent Events** − HTML5 introduces events which flow from web server to the web browsers and they are called Server-Sent Events (SSE).
- **Canvas** − This supports a two-dimensional drawing surface that you can program with JavaScript.
- **Audio & Video** − You can embed audio or video on your web pages without resorting to third-party plugins.
- **Geolocation** − Now visitors can choose to share their physical location with your web application.
- **Microdata** − This lets you create your own vocabularies beyond HTML5 and extend your web pages with custom semantics.
- **Drag and drop** − Drag and drop the items from one location to another location on a the same webpage.

The most interesting new API's in HTML5 are:

- HTML Geolocation
- HTML Drag and Drop
- HTML Local Storage
- HTML Application Cache
- HTML Web Workers
- HTML SSE

Below are the most interesting new HTML5 elements are:

- New **semantic elements** like <header>, <footer>, <article>, and <section>.
- New **attributes of form elements** like number, date, time, calendar, and range.
- New **graphic elements**: <svg> and <canvas>.
- New **multimedia elements**: <audio> and <video>.

SYSTEM DESIGN

6.1Design Methodology

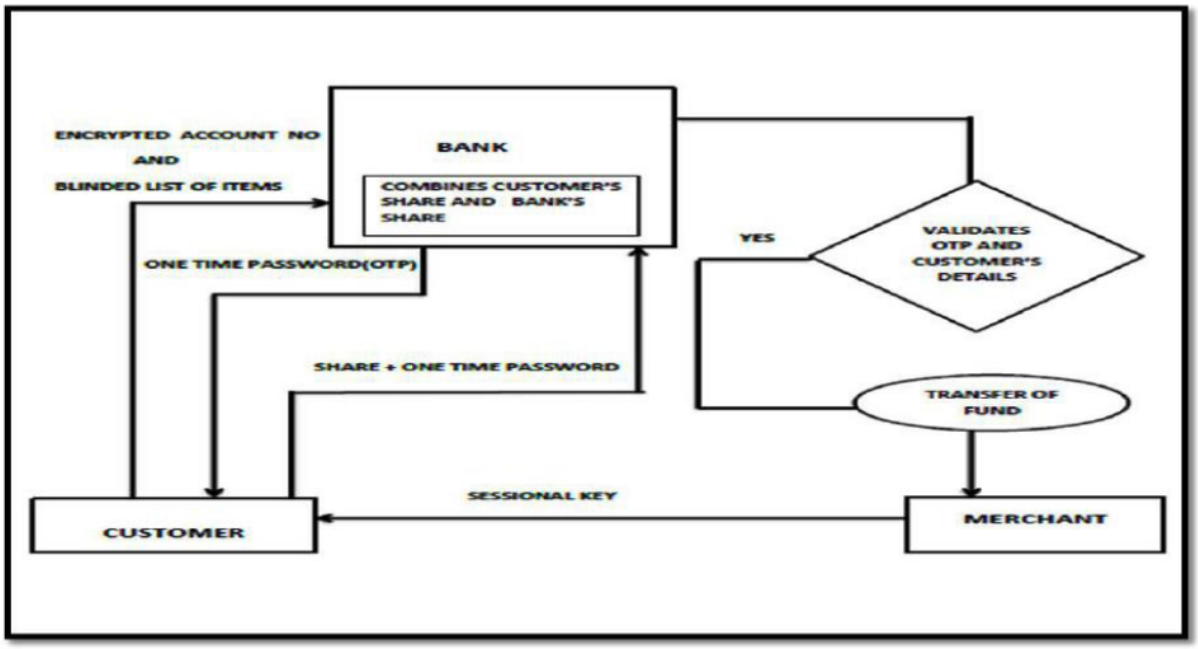


Figure 6.1: Architecture Diagram

Server will generate snapshot of text containing customer's account number and debit and credit card information is taken. From the snapshot image two shares are generated using visual cryptography. One share will be in the hand of customer and other one will be in database of bank. Merchant and customer agrees on a sessional key at the start of E-shopping. After that customer select the desired items and transfer blinded list of items along with encrypted account number to bank. This blinded list is generated by encrypting list of items with sessional key between customer and merchant. On receiving blinded list of items along with encrypted account number bank generates an one time password and securely transfers it to customer using quantum cryptography .After receiving one time password ,image steganography is performed by taking customer's share as cover image and hidden information as one time password and stego image is

passed to bank. Bank extracts embedded one time password so that share and one time password gets separated. Then Bank combines customer's share with bank's share and obtains account number and credit card details. Finally bank validates the one time password and credit card details and if both verification gets right fund is transferred to merchant account number.

UML DIAGRAM

7.1 Use Case Diagram:

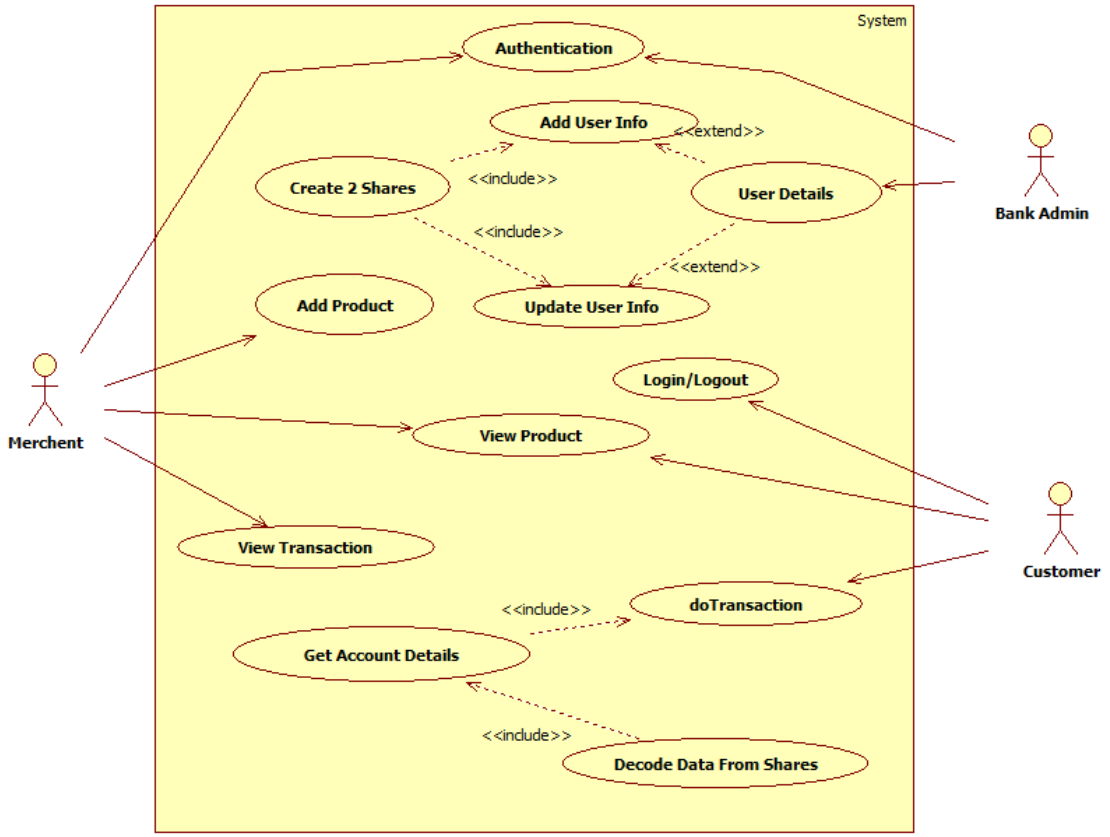


Figure 7.1 - Usecase Diagram

7.2 Data Flow Diagram:

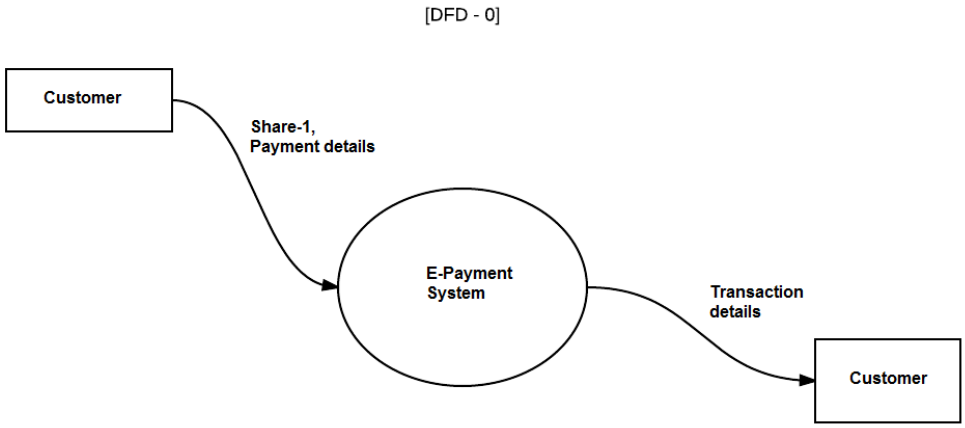


Figure 7.2.1 - DFD0 Diagram

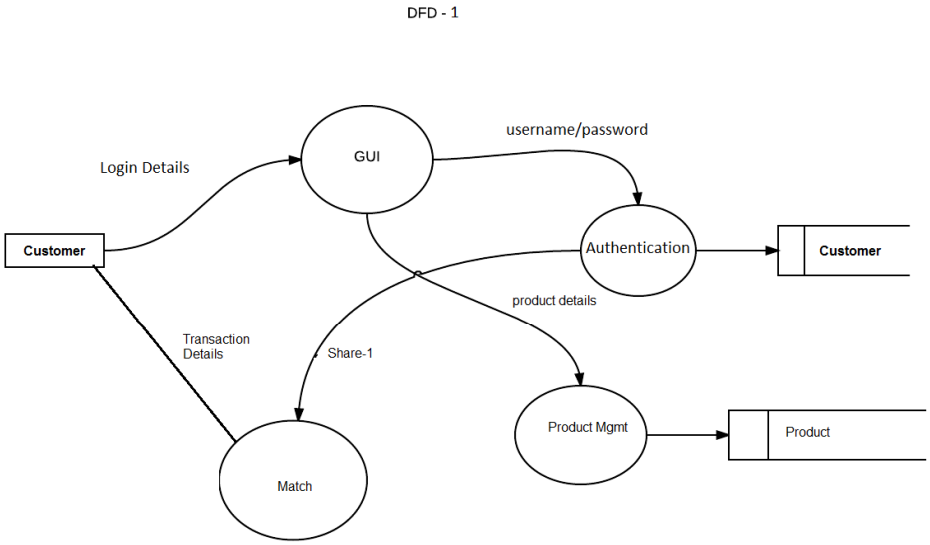


Figure 7.2.2 - DFD1 Diagram

7.3 Class Diagram:

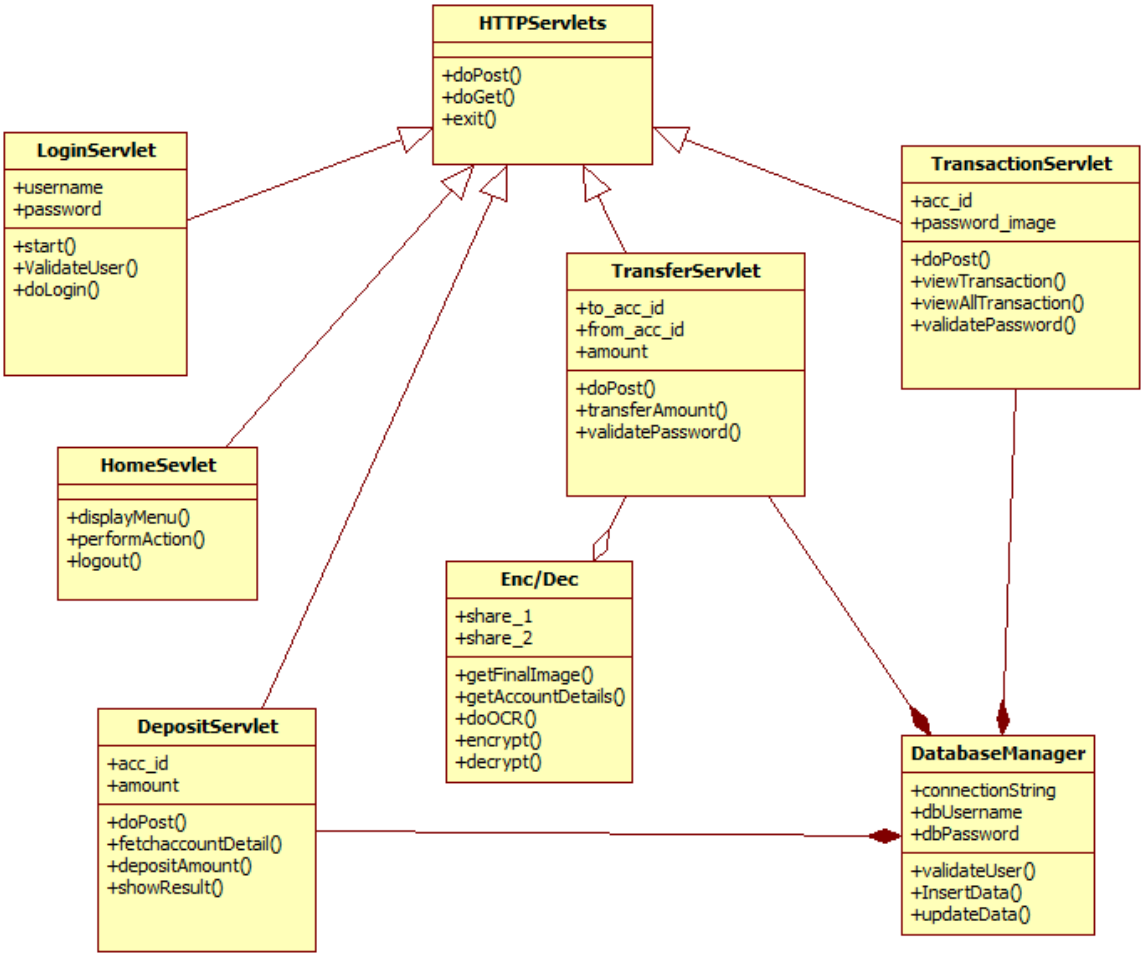


Figure 7.3 - Class Diagram

7.4 Component Diagram:

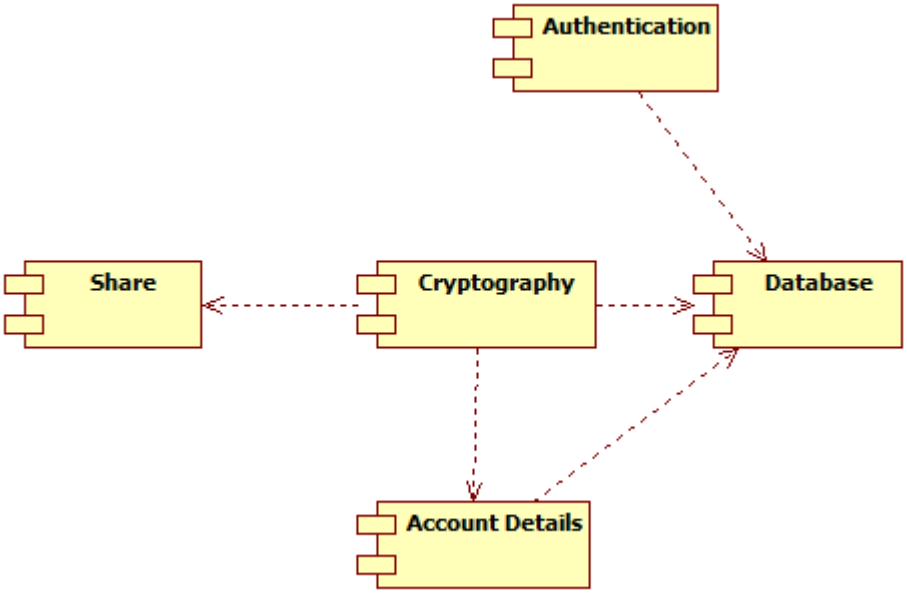


Figure 7.4 - Component Diagram

7.5 Deployment Diagram:

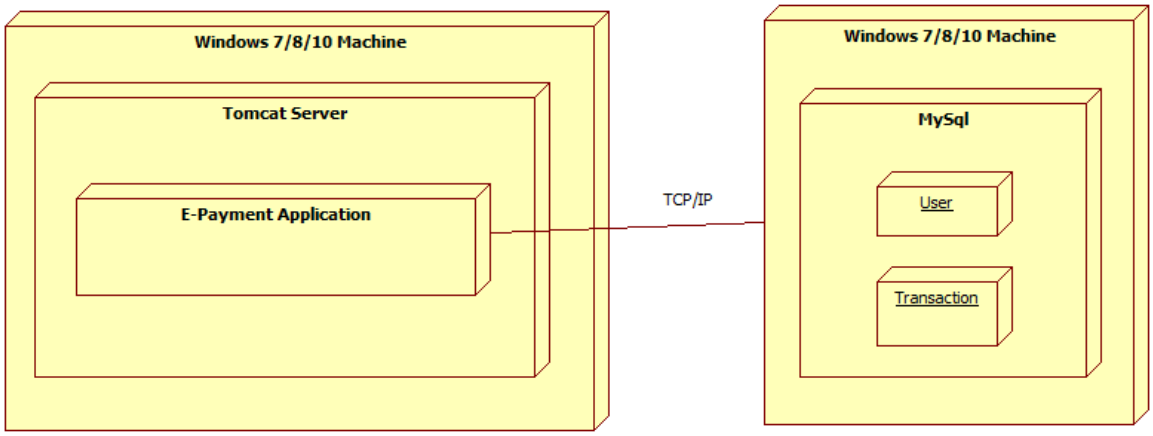


Figure 7.5 - Deployment diagram

7.6 Sequence Diagram:

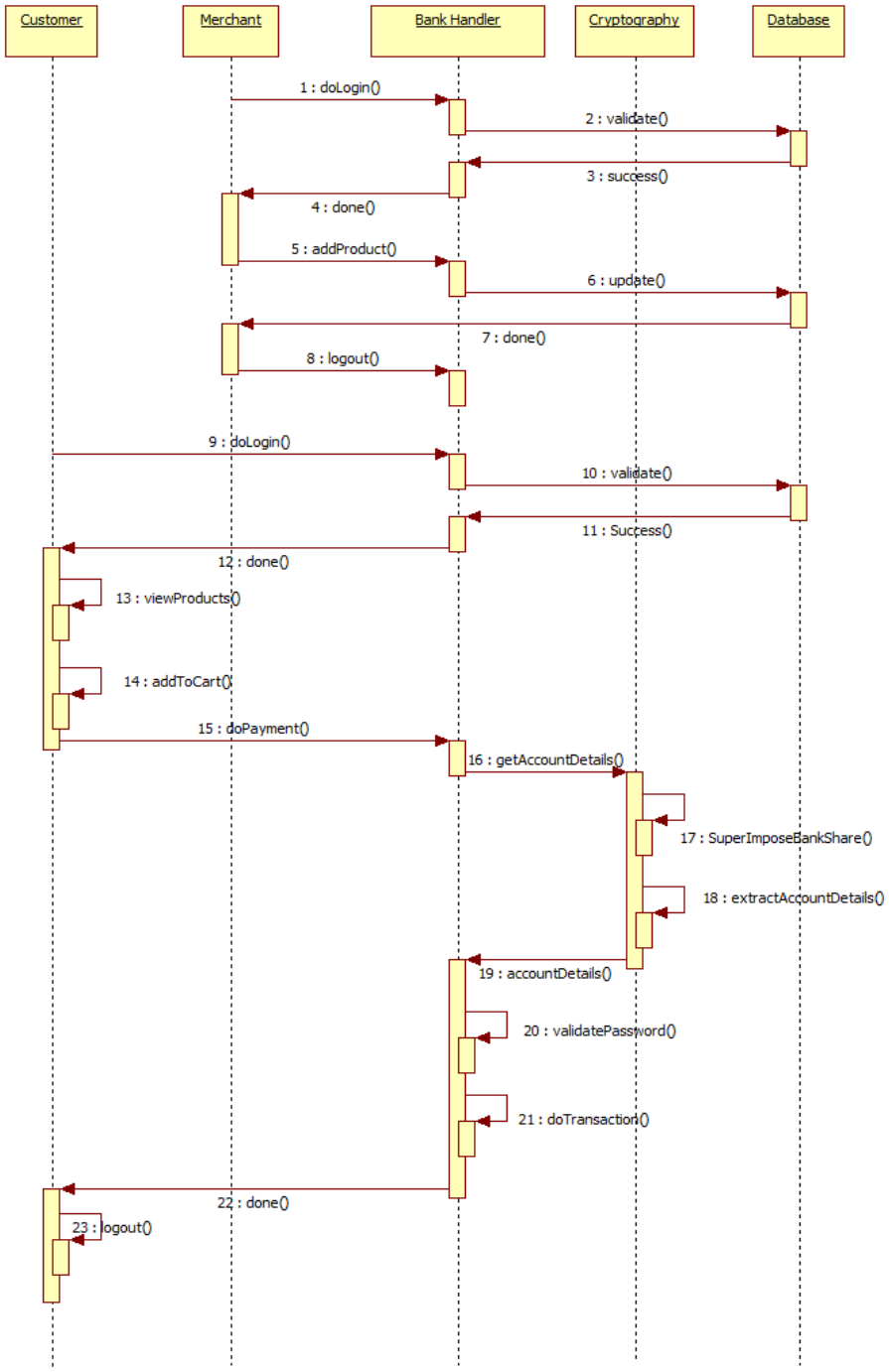


Figure 7.1Sequence diagram

7.7 Activity Diagram:

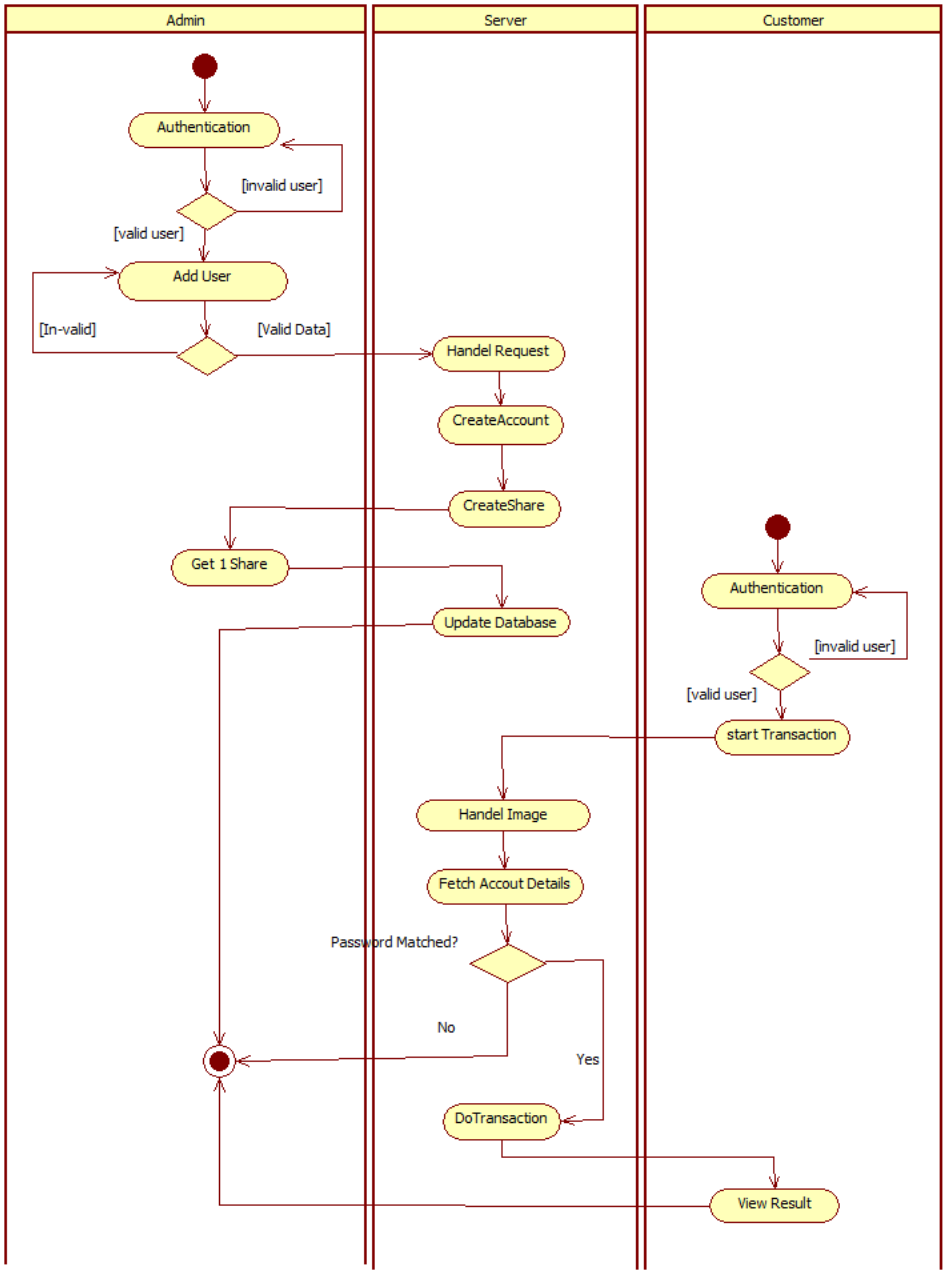
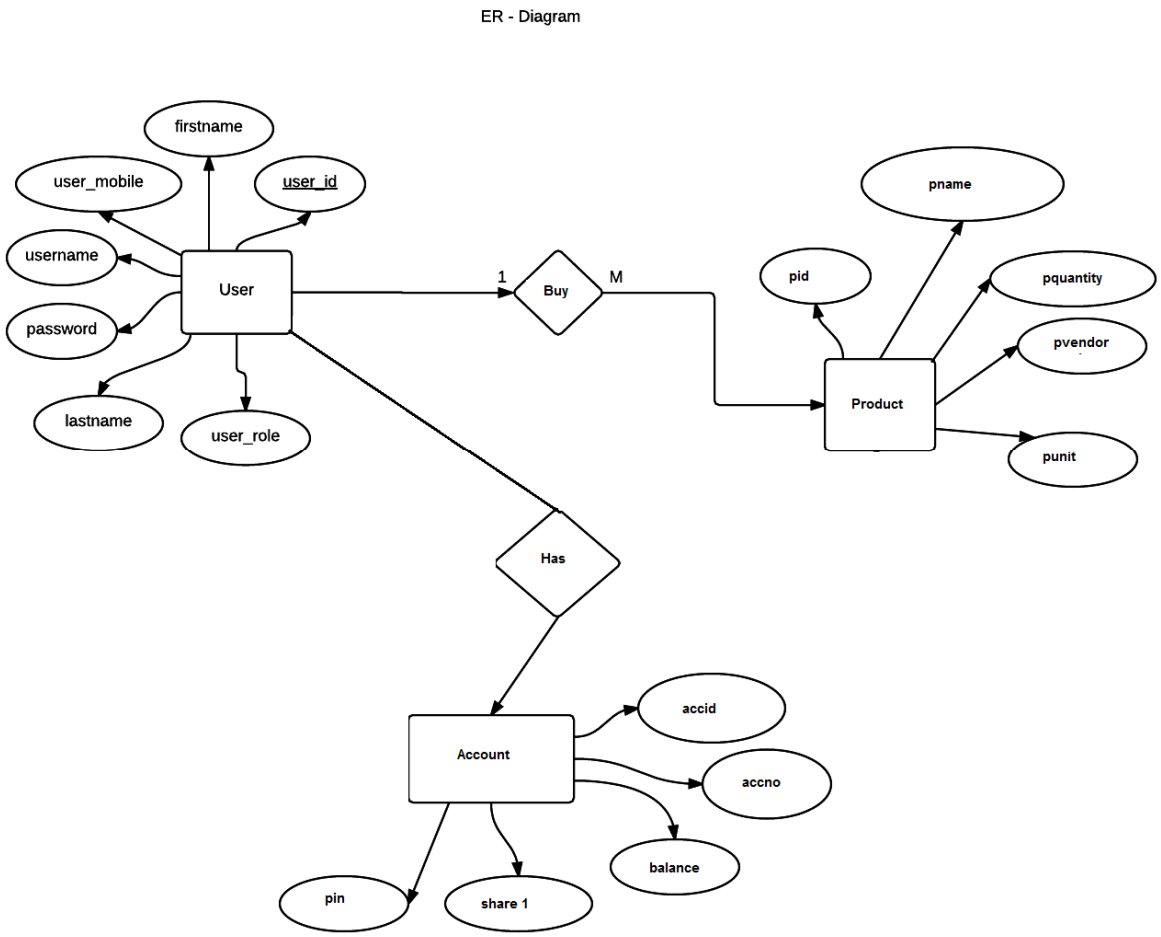


Figure 7.1Activity diagram

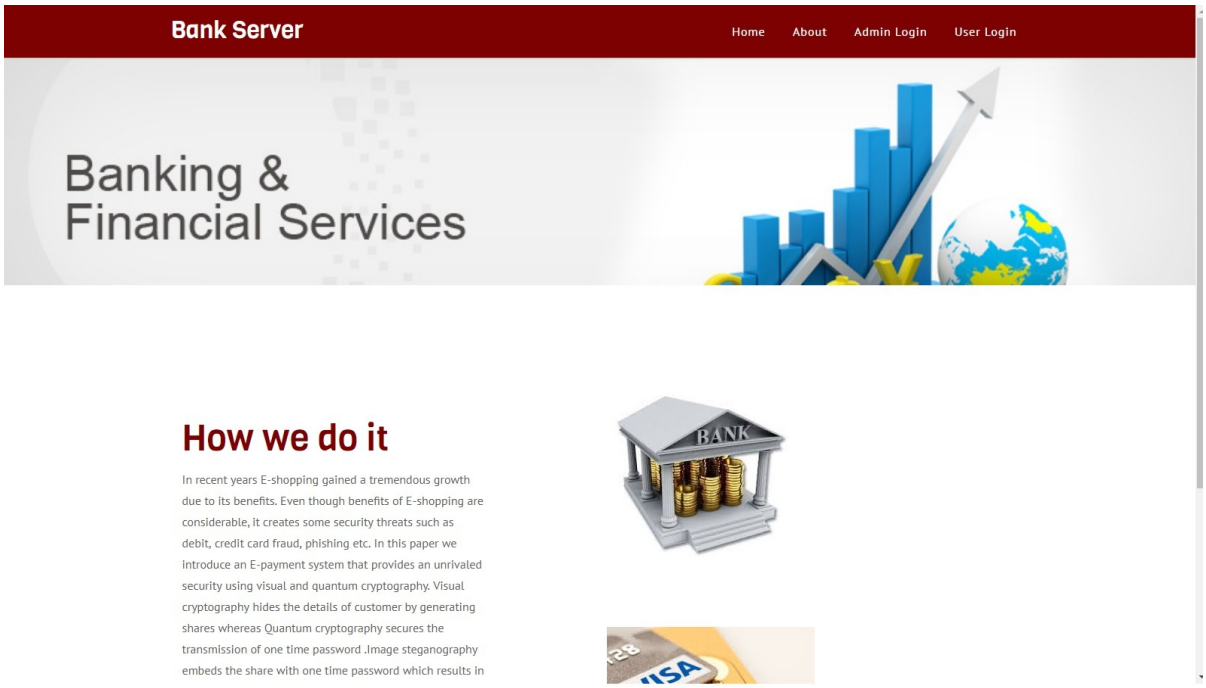
7.8 ER Diagram:



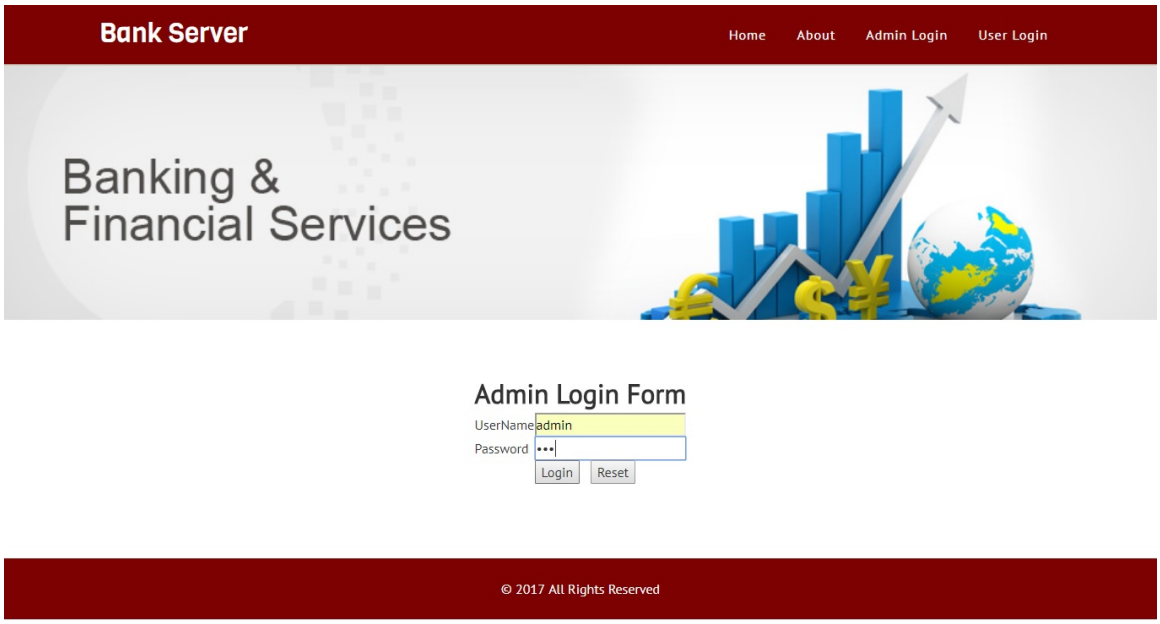
SCREENSHOTS

8.1 BANK SERVER

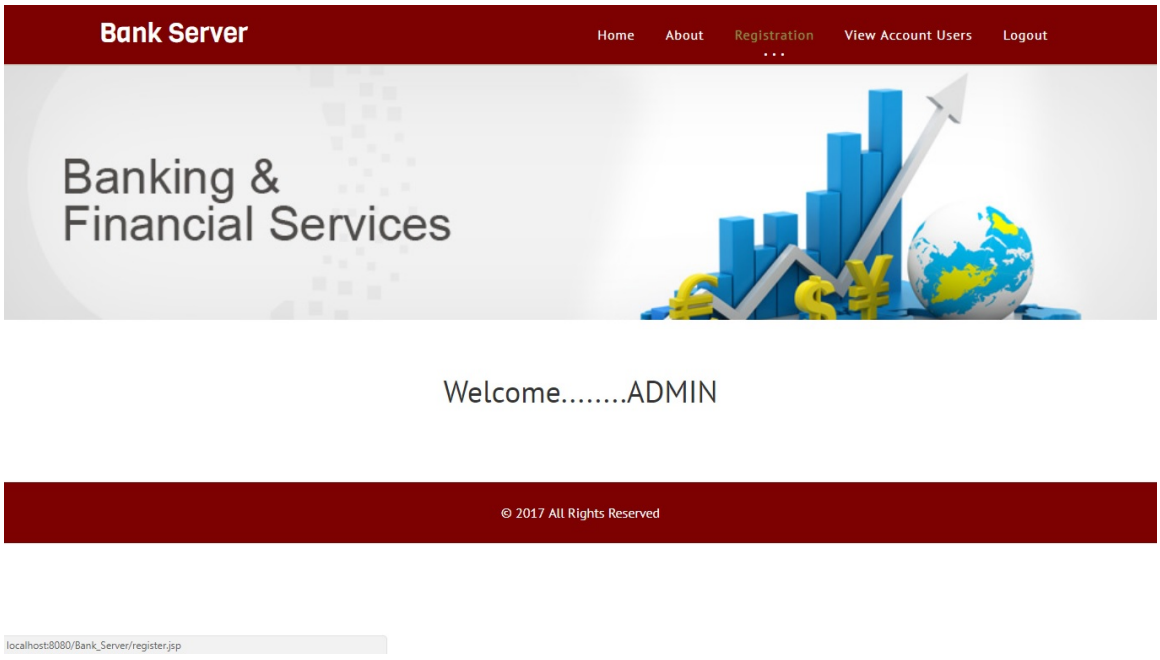
8.1.1 HOME PAGE



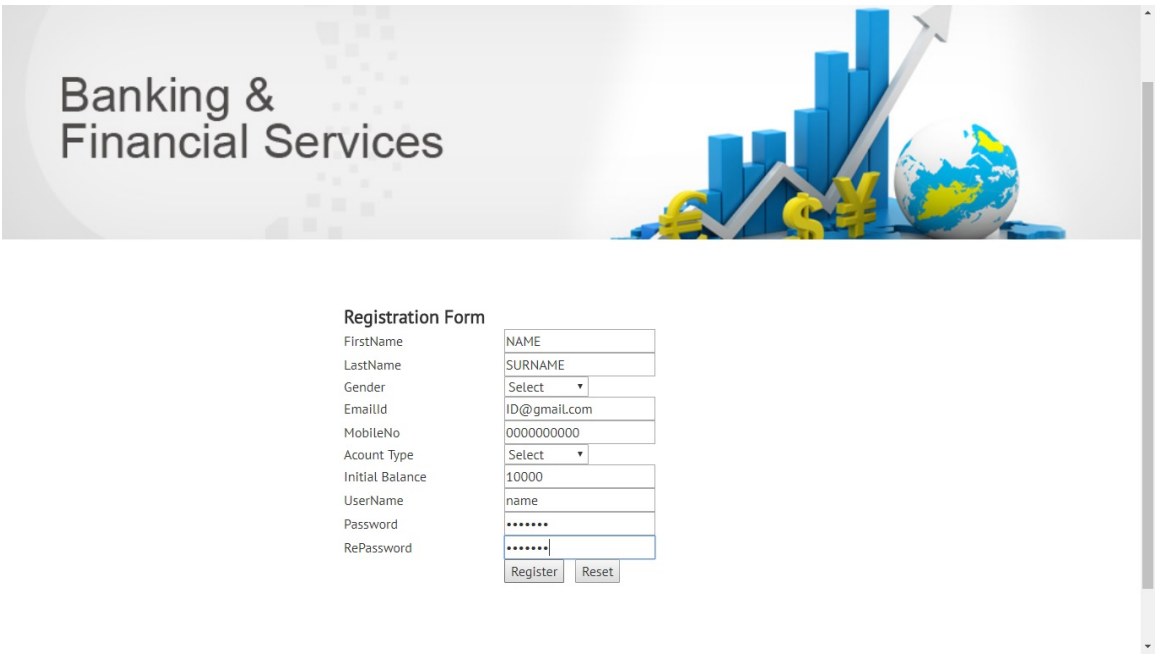
8.1.2 ADMIN LOGIN



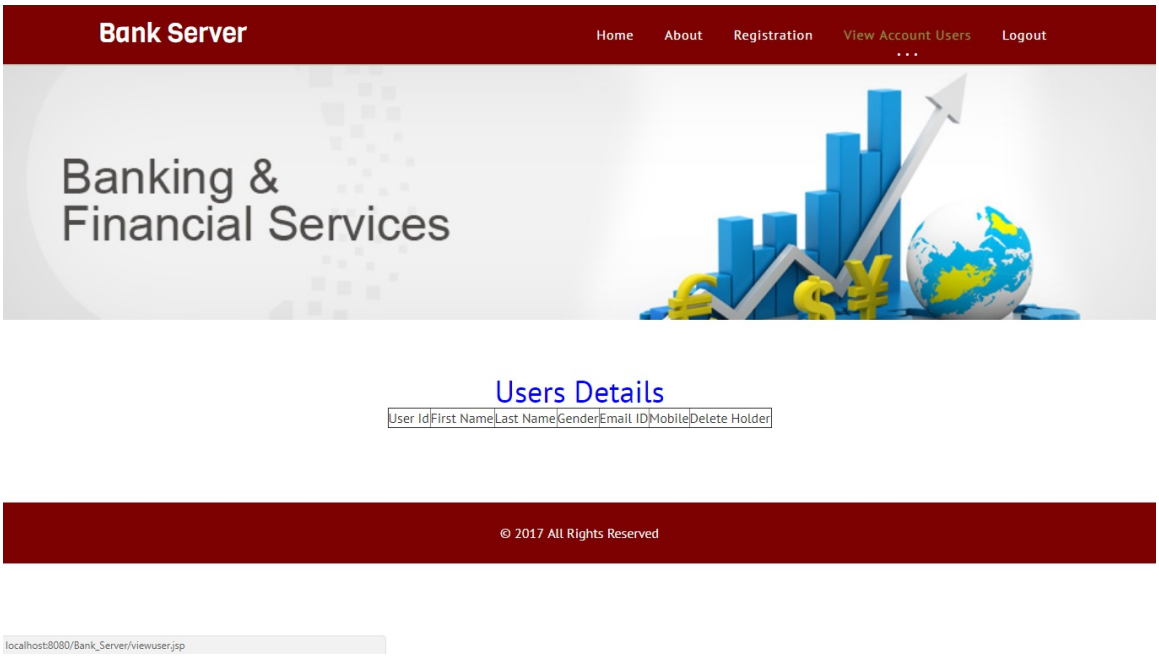
8.1.3 WELCOME PAGE



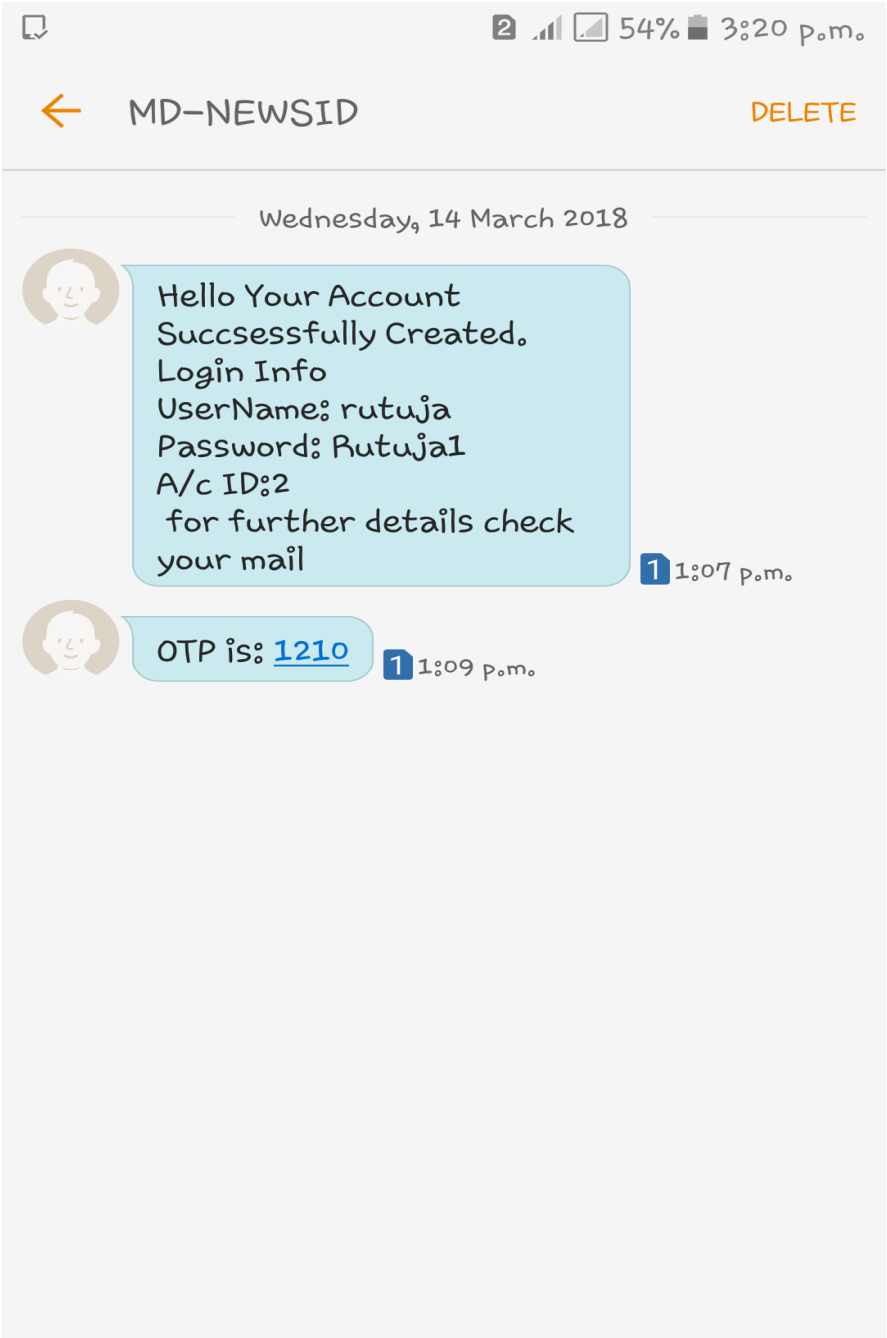
8.1.4 USER REGISTRATION



8.1.5 USER DETAILS

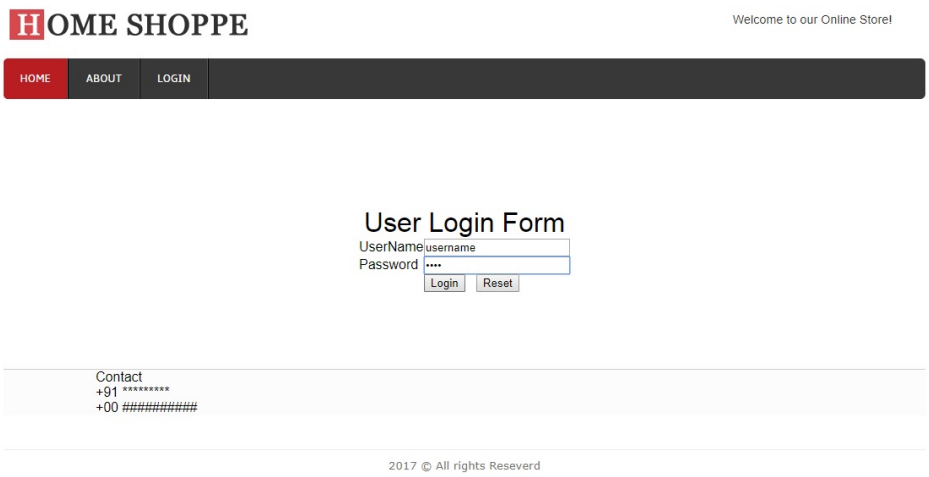


8.1.6 ACCOUNT DETAILS

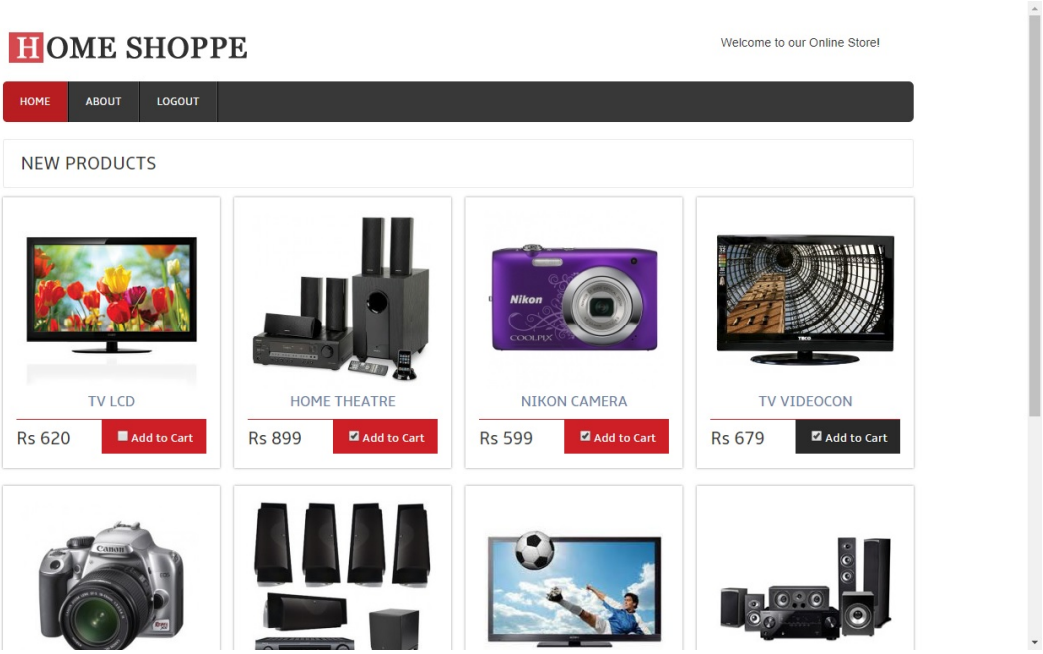


8.2 SHOPPING SITE

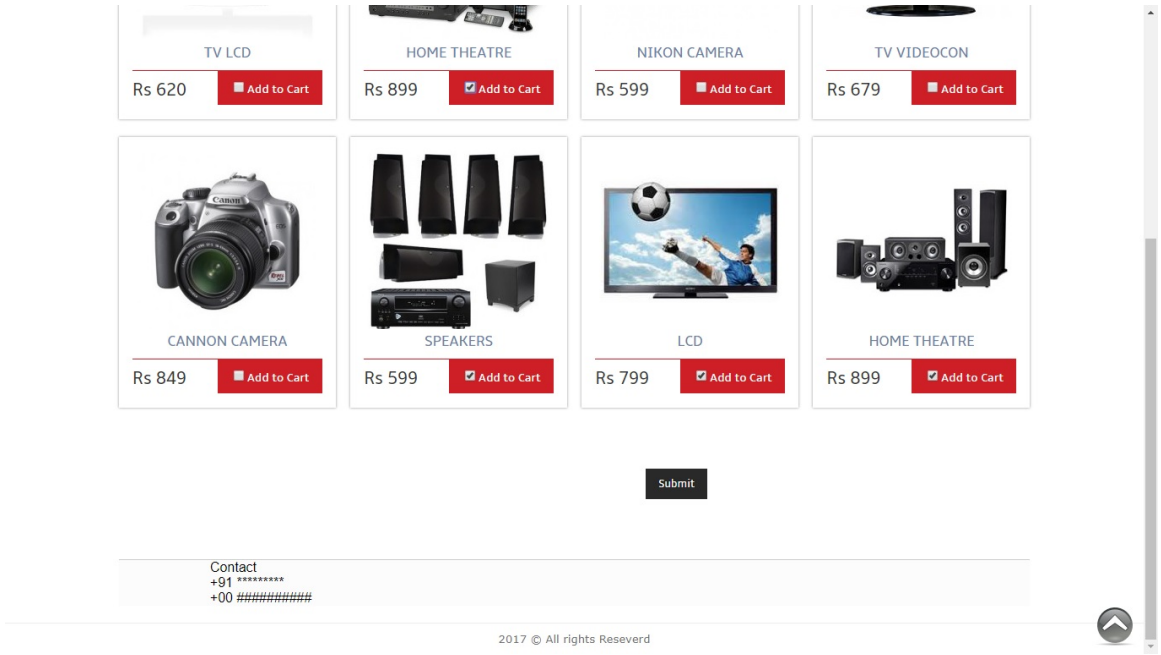
8.2.1 HOME LOGIN



8.2.2 ADD TO CART

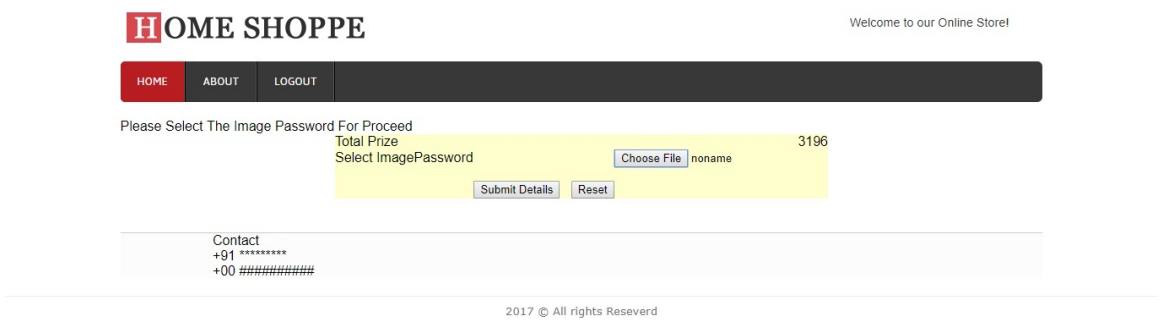


8.2.3 SUBMIT CART

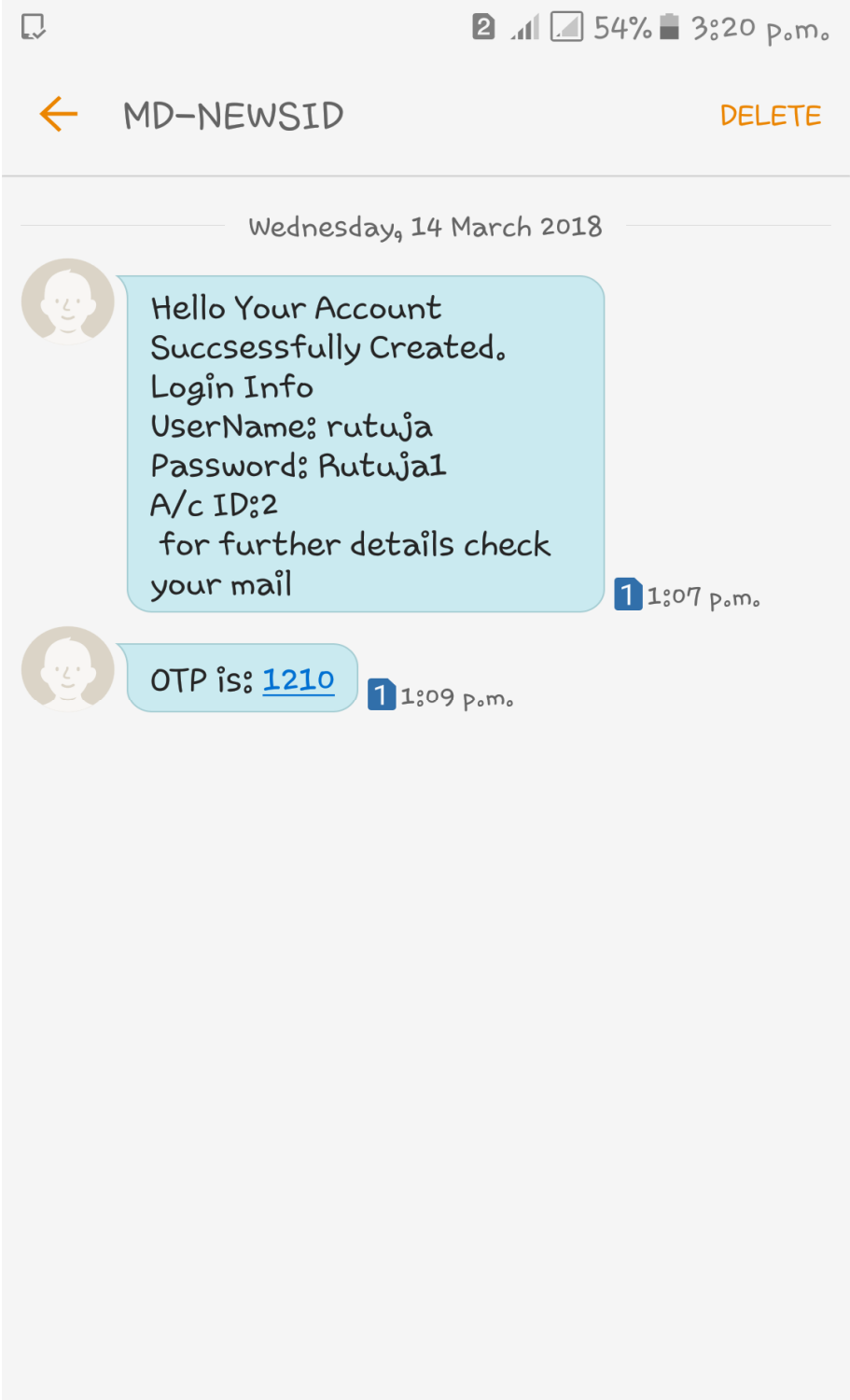


8.2.4 SUBMIT ACCOUNT ID

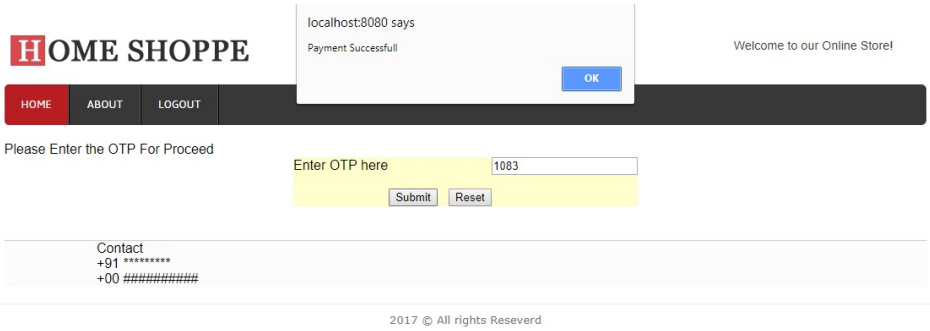
8.2.5 SUBMIT IMAGE



8.2.6 OTP



8.2.7 TRANSACTION SUCCESSFUL


HOME SHOPPE
localhost:8080 says
Payment Successfull
OK
Welcome to our Online Store!
HOME
ABOUT
LOGOUT
Please Enter the OTP For Proceed
Enter OTP here
1083
Submit
Reset
Contact
+91 *********
+00 ##########
2017 © All rights Reseverd

TESTING

9.1 Testing Information

Software Testing:

Software testing is the process of evaluation a software item to detect differences between given input and expected output. Also to assess the feature of A software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words software testing is a verification and validation process.

Verification:

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

Validation:

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

Basics of software testing:

There are two basics of software testing: black box testing and white box testing.

Black box Testing:

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

White box Testing :

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing.

Black box testing is often used for validation and white box testing is often used for verification.

Types of testing:

There are many types of testing like

- Unit Testing
- Integration Testing
- Functional Testing
- System Testing
- Stress Testing
- Performance Testing
- Usability Testing
- Acceptance Testing
- Regression Testing
- Beta Testing

Unit Testing:

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

Integration Testing:

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing.

Functional Testing:

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.

System Testing:

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

Stress Testing:

Stress testing is the testing to evaluate how system behaves under unfavorable conditions. Testing is conducted at beyond limits of the specifications. It falls under the class of black box testing.

Performance Testing:

Performance testing is the testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in performance requirements. It falls under the class of black box testing.

Usability Testing:

Usability testing is performed to the perspective of the client, to evaluate how the GUI is user-friendly? How easily can the client learn? After learning how to use, how

proficiently can the client perform? How pleasing is it to use its design? This falls under the class of black box testing.

Acceptance Testing:

Acceptance testing is often done by the customer to ensure that the delivered product meets the requirements and works as the customer expected. It falls under the class of black box testing.

Regression Testing:

Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly and is not damaging or imposing other modules to produce unexpected results. It falls under the class of black box testing.

Beta Testing:

Beta testing is the testing which is done by end users, a team outside development, or publicly releasing full pre-version of the product which is known as beta version. The aim of beta testing is to cover unexpected errors. It falls under the class of black box testing.

9.2 Test cases

Test Case ID	1
Test Case Description	Website should start successfully
Steps	1.Open website in browser.
Test Case Result	Website should be successful run and open first window.
Action Result	Website starts successfully
Status	Pass

Test Case ID	2
Test Case Description	System should have welcome page.
Steps	1. Run the application 2. Login window will get opened. 3. Main page.
Test Case Result	Website should display welcome page
Action Result	Welcome page displays
Status	Pass

Test Case ID	3
Test Case Description	Website should logged in for admin only
Steps	1. Open website. 2. Insert username and password 3. Click on submit
Test Case Result	When insert correct username and password, then logged in successfully
Action Result	Admin login successful
Status	Pass

Test Case ID	4
Test Case Description	Website should not logged in for un authenticate user
Steps	1. Open website. 2. Insert wrong username and password 3. Click on submit
Test Case Result	When insert wrong username and password, then it stay on same page means not logged in.
Action Result	Error message displays
Status	Pass

Test Case ID	5
Test Case Description	Website should get show proper error message
Steps	1. Open website. 2. Insert wrong username and password 3. Click on submit
Test Case Result	When insert wrong username and password, then it stay on same page and give proper error message
Action Result	Error message displays
Status	Pass

Test Case ID	6
Test Case Description	Admin will create users account
Steps	1. Install the application. 2. Insert username and password 3. Click on add new user
Test Case Result	Application will create users account
Action Result	Account create successfully
Status	Pass

Test Case ID	7
Test Case Description	Online shopping site should login successfully
Steps	1. Enter username 2. Enter password 3. Click pn submit
Test Case Result	After clicking on submit button; Home page will displays
Action Result	Website starts successfully
Status	Pass

Test Case ID	8
Test Case Description	Choose products and add to cart
Steps	1.Click on add to cart 2. Click on submit
Test Case Result	Product will added to cart
Action Result	Product will added to cart
Status	Pass

Test Case ID	9
Test Case Description	Enter account id and submit
Steps	1.Enter Account id 2.Click on Submit
Test Case Result	Successfully accept the account id
Action Result	ID will accepted
Status	Pass

Test Case ID	10
Test Case Description	Upload the encrypted image for authentication
Steps	1. Select the image 2.Click on submit
Test Case Results	Image successfully accepted
Action Result	Authentication will accepted
Status	Pass

Test ID	11
Test Case Description	Entering OTP
Steps	1.Enter OTP 2. Click on submit
Test Case Result	OTP will successfully accpeted
Action Result	OTP
Status	Pass

CONCLUSION

The proposed method preserves secret information of users, Verifies whether the website is a genuine/secure website or a phishing website. In this way project helps user to pay securly.

REFERENCES

1. Frdric Dupuis, Omar Fawzi, and Stephanie Wehner Entanglement Sampling and Applications ,IEEE TRANSACTIONS ON INFORMATION THEORY,VOL. 61, NO. 2, FEBRUARY 2015
2. Souvik Roy a, *, P.Venkateswaran b, Online Payment System using Steganography and Visual Cryptography ,2014 IEEE Students Conference on Electrical, Electronics and Computer Science
3. Siavash Khodambashi, Ali ZakerolhosseiniA Quantum Blind Signature Scheme for Electronic Payments The 22nd Iranian Conference on Electrical Engineering (ICEE 2014), May 20-22, 2014, Sahid Beheshti University
4. Monika Agarwal Department of Computer Science and Engineering, PDPM-IIITDM, Jabalpur, India, TEXT
5. STEGANOGRAPHIC APPROACHES A COMPARISON , International Journal of Network Security Its Applications (IJNSA), Vol.5, No.1, January 2013
6. Souvik Roy a, *, P.Venkateswaran b, A Text based Steganography Technique with Indian Root ,International Conference on Computational Intelligence:Modeling Techniques and Applications (CIMTA) 2013.