

ABSTRACT

ABSTRACT

Security is a broad topic and covers a multitude of sins. In its simplest form, it is concerned with making sure that nosy people cannot read, or worse yet, secretly modify messages intended for other recipients. It is concerned with people trying to access remote services that they are not authorized to use. Most security problems are intentionally caused by malicious people trying to gain some benefit, get attention, or to harm someone. Network security problems can be divided roughly into four closely intertwined areas: secrecy, authentication, non-repudiation, and integrity control. Secrecy, also called confidentiality, has to do with keeping information out of the hands of unauthorized users. This is what usually comes to mind when people think about network security. Authentication deals with determining whom you are talking to before revealing sensitive information or entering into a business deal. Non-repudiation deals with signatures.

Here by this method when we share images to any of our friends their secret key also encrypted using 3DES with the image. After sharing data the sharing notification will be visible for the user, but to retrieve the data user has to enter the secret key. Using this method sharing data can be protected even if the account is hacked. Also while sharing if the key for sharing is not correct the share will be rejected, this will help to prevent fake messages and to restrict cheating by hacking the account.

INTRODUCTION

1. INTRODUCTION

The Project is about Social networking site hence it is a website called Seqshare. As the name suggests the website is designed specifically for secure share of images in a social network. We used this project as an extension for facebook. This website include features that are included in common social networks and along with this features we can share images in a most secure way.

The website provide user friendly environment and it provide all detail required for a native user. Seqshare is just like Facebook. A User first creates a "Profile" in which the user provides their social and personal details. They can add friends, chat with them etc. The main feature of this project is the security during image sharing. Here when we are uploading an image we can do the settings and can cancel the unauthorized sharing of the image and a notification will be sent, if any attempt is done.

Social Networking Website project itself is a huge project comprising various features like profile updating, friend's list organization and various other applications to enhance the overall look and feel of the website. However, in this project I am basically working on two essential features (profile management & friends' organization).

Profile management: module maintain the profile of a user like name, gender, dob, email, status etc. Friends organization module maintains the friend list, handles request and sends request to the other user. Profiles and Friends lists are two key features on social network.

To overcome this problem we are introducing a new feature in Social Networking Websites. This can be used as an extension to Facebook. Here there is also features like "Adding Friends", sending them "Friend Requests", "Chatting", "Image Uploading" and "Sharing" etc. But here we can share an image by setting security details. Here we can enable or disable share, save and download. When we disable the share feature and share the image, then even if the image is saved or downloaded to another location the image cannot be shared. And also a notification will be received by the sender about the unauthorized attempt of sharing image. Through this we can ensure the security of the images from unauthorized users.

ORGANIZATION PROFILE

2. ORGANIZATION PROFILE

Syntax Soft-Tech India (P) Ltd is an emerging Web and Software developing company with strong and qualified staff in respective field. Their exquisite design skills and internet oriented marketing strategies helps our customers to up-lift their business through us. The 'USP' of Syntax Soft-Tech India (P) Ltd is our 100% Responsibility and Devotion towards customer satisfaction. This dedication helps Syntax Soft-Tech India (P) Ltd to compete with the best in the business in this highly competitive field. With the help of generation next technologies, we provide latest trends in applications on various platforms of Computers, Tablets, Mobile Phones and Social networking apps with remarkable designs. Our pleased client lists can testimony for Syntax Soft-Tech India (P) Ltd.

SYSTEM ANALYSIS

3. ANALYSIS

System analysis also referred to as system study is the primary phase of the software development. It is the study of existing operation to learn what they accomplish, why they work as they do and what role they may have in future processing activities. In studying the current system, it is necessary to identify the essential data and procedure required for a new approach. It is also necessary to pinpoint current weakness and problem so that will not be carried over to the new system.

3.1. Existing System

All the social media websites mainly aim at searching friends easily on entire network and sending friend request adding them as friends. After this all most of the users use this websites for exposing their new faces and styles in front of friends. They mainly do this using “Image uploading” and “Sharing” option.

Security of such images that we are sharing will be at a great risk, because when we are sharing an image, the image can be viewed, can be saved and also can be downloaded. When hackers or other type of unauthorized users got our username and password, they can easily share the image for unwanted users.

Limitations of existing system:-

- The multimedia content, is being attacked, and used illegally by individuals
- Users are not completely aware of who has access to their multimedia content
- Also a user cannot prevent further sharing of the image.
- User cannot share a secure image with another user.
- Less data confidentiality.

3.2. Proposed System

To overcome these problems we are introducing a new feature in Social Networking Websites. This can be used as an extension to Facebook. Here there is also features like “Adding Friends”, sending them “Friend Requests”, ”Chatting”, ”Image Uploading” and “Sharing”.

But here we can share an image by setting security details. Here we can enable or disable share, save and download. When we disable the share feature and share the image, then even if the image is saved or downloaded to another location the image cannot be shared. And also a notification will be received by the sender about the unauthorized attempt of sharing image. Through this we can ensure the security of the images from unauthorized users.

Advantages of proposed system:-

- This system allows a user to share the secure image with another user.
- The system provides a better way to share the image with another even if the image is renamed.
- We can protect the image by applying security settings to prevent the further share, downloading and saving of the image.
- The image is secured even if the account is hacked.

3.3. Feasibility Analysis

Feasibility analysis is the procedure for identifying the candidate system, evaluating and electing the most feasible system. This is done by investigating the existing system and generating ideas about a new system. It is a test of system proposal according to its work ability, impact on the organization, ability to meet user needs, and effective use of resource. The object of feasibility study is not to solve the problem but to acquire a sense of its scope.

The key considerations are involved in the feasibility analysis.

- Economical Feasibility
- Technical Feasibility
- Behavioral Feasibility
- Operational Feasibility
- Legal Feasibility

3.3.1. Economic Feasibility

Economic analysis system is the most frequently method for evaluating the effectiveness of the system. In Economic Feasibility the system can be judged to be economically feasible. By evaluating the project 'Seqshare' we can understand the

economic feasibility of the project. Through this evaluation we can know the cost/benefit of the project 'Seqshare'.

3.3.2. Technical Feasibility

Technical feasibility of the system checks whether the required technology, hardware and software, the sources are available or not. Here, in the project 'Seqshare' the required technologies are Java and 3DES. And these are available to develop the project. The 3DES algorithm is used for steganography purpose. The required hardware and software are also available for the project. The sources that are needed to develop the project like Netbeans, WAMP and also the data from the existing system are also available. So that the project 'Seqshare' is technically feasible.

3.3.3. Behavioral Feasibility

Behavioral feasibility of the system evaluates the behavior of the system. The project entitled 'Seqshare' is behaviorally feasible. Because it can be run on any system.

3.3.4. Operational Feasibility

Operational feasibility of the system evaluates the operations of the system works properly.

3.3.5. Legal Feasibility

Legal feasibility of the system ensures the legal contracts and

3.4. Software Overview

3.4.1. Java Technology

Initially the language was called as oak. But it was renamed as Java in 1995. The primary motivation of this language was the need for a platform-independent (i.e., architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices.

- Java is a programming language.
- Java is cohesive and consistent.
- Except for those constraints imposed by the Internet environment, Java gives the programmer, full control.
- Finally, Java is to Internet programming where C was to system programming.

❖ **Importance of Java to the Internet**

Java has had a profound effect on the Internet. This is because; Java expands the Universe of objects that can move about freely in Cyberspace. In a network, two categories of objects are transmitted between the Server and the Personal computer.

They are: Passive information and Dynamic active programs.

The Dynamic, Self-executing programs cause serious problems in the areas of Security and probability. But, Java addresses those concerns and by doing so, has opened the door to an exciting new form of program called the Applet. Java can be used to create two types of programs:

Applications and Applets: An application is a program that runs on our Computer under the operating system of that computer. It is more or less like one creating using C or C++. Java's ability to create Applets makes it important. An Applet is an application designed to be transmitted over the Internet and executed by a Java – compatible web browser. An applet is actually a tiny Java program, dynamically downloaded across the network, just like an image. But the difference is, it is an intelligent program, not just a media file. It can react to the user input and dynamically change.

❖ **Features of Java Security**

Every time you that you download a normal program; you are risking a viral infection. Prior to Java, most users did not download executable programs frequently, and those who did scan them for viruses prior to execution. Most users still worried about the possibility of infecting their systems with a virus. In addition, another type of malicious program exists that must be guarded against. This type of program can gather private information, such as credit card numbers, bank account balances, and passwords. Java answers both these concerns by providing a –firewall between a network application and your computer.

When you use a Java-compatible Web browser, you can safely download Java applets without fear of virus infection or malicious intent.

- **Portability**

For programs to be dynamically downloaded to all the various types of platforms connected to the Internet, some means of generating portable executable code is needed. As you will see, the same mechanism that helps ensure security also helps create portability. Indeed, Java's solution to these two problems is both elegant and efficient.

- **The Byte code**

The key that allows the Java to solve the security and portability problems is that the output of Java compiler is Bytecode. Bytecode is a highly optimized set of instructions designed to be executed by the Java run-time system, which is called the Java Virtual Machine (JVM). That is, in its standard form, the JVM is an interpreter for bytecode.

Translating a Java program into byte code helps makes it much easier to run a program in a wide variety of environments. There as on is, once the run- time package exists for a given system, any Java program can run on it.

Although Java was designed for interpretation, there is technically nothing about Java that prevents on-the-fly compilation of byte code into native code.

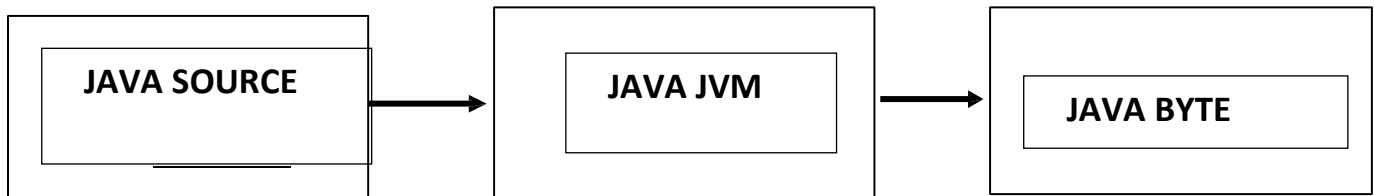
Sun has just completed its JustInTime(JIT) compiler for bytecode. When the JIT compiler is a part of JVM, it compiles byte code into executable code in realtime, on a piece-by-piece, demand basis. It is not possible to compile an entire Java program into executable code all at once, because Java performs various run-time checks that can be done only at run time. The JIT compiles code, as it is needed, during execution.

- **Java Virtual Machine (JVM)**

Beyond the language, there is the Java virtual machine. The Java virtual machine is an important element of the Java technology. The virtual machine can be embedded within a web browser or an operating system. Once a piece of Java code is loaded onto a machine, it is verified. As part of the loading process, a class loader is invoked and does bytecode verification makes sure that the code that's has been generated by the compiler will not corrupt the machine that it's loaded on. Byte code verification takes place at the end of the

compilation process to make sure that is all accurate and correct. So bytecode verification is integral to the compiling and executing of Java code.

- **Overall Description**



- **Java Architecture**

Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each platform by the run-time environment. Java is a dynamic system, able to load code when needed from a machine in the same room or across the planet.

- **Compilation of code**

When you compile the code, the Java compiler creates machine code (called byte code) for a hypothetical machine called Java Virtual Machine (JVM). The JVM is supposed to execute the byte code. The JVM is created for overcoming the issue of portability. The code is written and compiled for one machine and interpreted on all machines. This machine is called Java Virtual Machine.

- **Simple**

Java was designed to be easy for the Professional programmer to learn and to use effectively. If you are an experienced C++ programmer, learning Java will be even easier. Because Java inherits the C/C++ syntax and many of the object oriented features of C++. Most of the confusing concepts from C++ are either left out of Java or implemented in a cleaner, more approachable manner. In Java there are a small number of clearly defined ways to accomplish a given task.

- **Object Oriented**

Java was not designed to be source-code compatible with any other language. This allowed the Java team the freedom to design with a blank slate. One outcome of this was a clean usable, pragmatic approach to objects. The object model in Java is simple and easy to extend, while simple types, such as integers, are kept as high-performance non-objects.

- **Robust**

The multi-platform environment of the Web places extra ordinary demands on a program, because the program must execute reliably in a variety of systems. The ability to create robust programs was given a high priority in the design of Java. Java is strictly typed language; it checks your code at compile time and run time.

Java virtually eliminates the problems of memory management and de-allocation, which is completely automatic. In a well-written Java program, all runtime errors can and should be managed by your program.

3.4.2. Java Database Connectivity

JDBC is a Java API for executing SQL statements. (As a point of interest, JDBC is a trademarked name and is not an acronym; nevertheless, JDBC is often thought of as standing for Java Database Connectivity. It consists of a set of classes and interfaces written in the Java programming language. JDBC provides a standard API for tool/database developers and makes it possible to write database applications using a pure Java API.

Using JDBC, it is easy to send SQL statements to virtually any relational database. One can write a single program using the JDBC API, and the program will be able to send SQL statements to the appropriate database. The combinations of Java and JDBC lets a programmer write it once and run it anywhere.

- ❖ **What Does JDBC do?**

Simply put, JDBC makes it possible to do three things

- Establish a connection with a database
- Send SQL statements
- Process the results.

- ❖ **JDBC versus ODBC and other APIs**

At this point, Microsoft's ODBC (Open Database Connectivity) API is that probably the most widely used programming interface for accessing relational databases. It offers the ability to connect to almost all databases on almost all platforms.

So why not just use ODBC from Java? The answer is that you can use ODBC from Java, but this is best done with the help of JDBC in the form of the JDBC-ODBC Bridge, which we will cover shortly. The question now becomes "Why do you need JDBC?" There are several answers to this question:

1. ODBC is not appropriate for direct use from Java because it uses a C interface.

Calls from Java to native C code have a number of drawbacks in the security, implementation, robustness, and automatic portability of applications.

2. A literal translation of the ODBC C API into a Java API would not be desirable.

For example, Java has no pointers, and ODBC makes copious use of them, including the notoriously error-prone generic pointer "void*". You can think of JDBC as ODBC translated into an object-oriented interface that is natural for Java programmers.

3. ODBC is hard to learn. It mixes simple and advanced features together, and it has complex options even for simple queries. JDBC, on the otherhand, was designed to keep simple things simple while allowing more advanced capabilities where required.

4. A Java API like JDBC is needed inorder to enable a "pure Java" solution. When ODBC is used, the ODBC driver manager and drivers must be manually installed on every client machine. When the JDBC driver is written completely in Java, however, JDBC code is automatically installable, portable, and secure on all Java platforms from network computers to mainframes.

❖ Two-tier and Three-tier Models

The JDBC API supports both two-tier and three-tier models for database access. In the two-tier model, a Java applet or application talks directly to the database. This requires a JDBC driver that can communicate with the particular database management system being accessed. A user's SQL statements are delivered to the database, and the results of those statements are sent back to the user. The database may be located on another machine to which the user is connected via a network.

❖ **JDBC Driver Types**

The JDBC drivers that we are aware of at this time fit into one of four categories:

- JDBC-ODBC bridge plus ODBC driver
- Native-API partly-Java driver
- JDBC-Net pure Javadrivers
- Native-protocol pure Javadrivers

❖ **JDBC-ODBC Bridge**

If possible, use a Pure Java JDBC driver instead of the Bridge and an ODBC driver. This completely eliminates the client configuration required by ODBC. It also eliminates the potential that the Java VM could be corrupted by an error in the native code brought in by the Bridge (that is, the Bridge native library, the ODBC driver manager library, the ODBC driver library, and the database client library).

❖ **What Is the JDBC-ODBC Bridge?**

The JDBC-ODBC Bridge is a JDBC driver, which implements JDBC operations by translating them into ODBC operations. To ODBC it appears as a normal application program. The Bridge implements JDBC for any database for which an ODBC driver is available. The Bridge is implemented as the `Sun.jdbc.odbc` Javapackage and contains a native library used to access ODBC. The Bridge is a joint development of Innersole and JavaSoft.

❖ **JDBC connectivity**

The JDBC provides database-independent connectivity between the J2EE platform and a wide range of tabular data sources. JDBC technology allows an Application Component Provider to:

- Perform connection and authentication to a database server
- Manage transactions
- Move SQL statements to a database engine for preprocessing and execution
- Execute stored procedures
- Inspect and modify the results from Select statements

3.4.3. Java Server Pages(JSP)

Java server Pages is a simple, yet powerful technology for creating and maintaining dynamic-content web pages. Based on the Java programming language, Java Server Pages offers proven portability, open standards, and a mature re-usable component model. The Java Server Pages architecture enables the separation of content generation from content presentation. This separation not eases maintenance headaches; it also allows web team members to focus on their areas of expertise. Now, web page designer can concentrate on layout, and web application designers on programming, with minimal concern about impacting each other's work.

Features of JSP:-

- **Portability:**

Java Server Pages files can be run on any web server or web-enabled application server that provides support for them. Dubbed the JSP engine, this support involves recognition, translation, and management of the Java Server Page life cycle and its interaction components.

- **Components:**

It was mentioned earlier that the Java Server Pages architecture can include reusable Java components. The architecture also allows for the embedding of a scripting language directly into the Java Server Pages file. The components current supported include JavaBeans, and Servlets.

- **Processing:**

A Java Server Pages file is essentially an HTML document with JSP scripting or tags. The Java Server Pages file has a JSP extension to the server as a Java Server Pages file. Before the page is served, the Java Server Pages syntax is parsed and processed into a Servlet on the server side. The Servlet that is generated outputs real content in straight HTML for responding to the client.

- **Access Models:**

A Java Server Pages file may be accessed in at least two different ways. A client's request comes directly into a JavaServerPage. In this scenario, suppose the page accesses

reusable Java Bean components that perform particular well-defined computations like accessing a database. The result of the Beans computations, called result sets is stored within the Bean as properties. The page uses such Beans to generate dynamic content and present it back to the client.

❖ **Steps in the execution of a JSP Application:**

1. The client sends a request to the web server for a JSP file by giving the name of the JSP file within the form tag of a HTML page.
2. This request is transferred to the Java Web Server. At the server side Java Web Server receives the request and if it is a request for a jsp file server gives this request to the JSP engine.
3. JSP engine is program which can understand the tags of the jsp and then it converts those tags into a Servlet program and it is stored at the server side. This Servlet is loaded in the memory and then it is executed and the result is given back to the Java Web Server and then it is transferred back to the result is given back to the Java Web Server and then it is transferred back to the client.

3.4.4. AJAX

Asynchronous JavaScript XML. It is not a new programming language, but a new way to use existing standards. AJAX is the art of exchanging data with a server and updating parts of a webpage without reloading the whole page.

AJAX is a group of interrelated web development technique used on the client side to create asynchronous web application. It is not a single technology but a group of technologies. HTML, CSS can be used in combination to mark up style information.

3.4.5. HTML

HTML, an initialism of Hypertext Markup Language, is the predominant markup language for web pages. It provides a means to describe the structure of text-based information in a document—by denoting certain text as headings, paragraphs, lists, and soon—and to supplement that text with interactive forms, embedded images, and other objects. HTML is written in the form of labels (known as tags), surrounded by angle brackets. HTML can also describe, to some degree, the appearance and semantics of a

document, and can include embedded scripting language code which can affect the behavior of web browsers and other HTML processors.

HTML is also often used to refer to content of the MIME type text/html or even more broadly as a generic term for HTML whether in its XML-descended form (such as XHTML1.0 and later) or its form descended directly from SGML

Hypertext Markup Language (HTML), the languages of the World Wide Web (WWW), allows users to produce Web pages that include text, graphics and pointer to other Web pages (Hyperlinks).

HTML is not a programming language but it is an application of ISO Standard 8879, SGML (Standard Generalized Markup Language), but specialized to hypertext and adapted to the Web. The idea behind Hypertext is that instead of reading text in rigid linear structure, we can easily jump from one point to another point. We can navigate through the information based on our interest and preference. A markup language is simply a series of elements, each delimited with special characters that define how text or other items enclosed within the elements should be displayed. Hyperlinks are underlined or emphasized words that lead to other documents or some portions of the same document.

HTML can be used to display any type of document on the host computer, which can be geographically at a different location. It is a versatile language and can be used on any platform or desktop. HTML provides tags (special codes) to make the document look attractive. HTML tags are not case-sensitive. Using graphics, fonts, different sizes, color, etc., can enhance the presentation of the document. Anything that is not a tag is part of the document itself.

3.4.6. Back End: MYSQL

MySQL database has become the world's most popular open source database because of its consistency, fast performance, high reliability and ease of use. It has also become the database of choice for a new generation of applications built on the LAMP stack (Linux, Apache, MySQL, PHP/Perl/Python). MySQL runs on more than 20 platforms including Linux, Windows, OS/2, HP-UX, AIX, Netware, giving you the kind of flexibility that puts you in control. MySQL offers a comprehensive range of certified software, support, training and consulting.

MySQL is a multithreaded, multi-user SQL Database Management System. MySQL's implementation of a relational database is an abstraction on top of a computer's file system. The relational database abstraction allows collection of data items to be organized as a set of formally described tables. Data can be accessed or reassembled from these tables in many different ways, which do not require any organization of the database tables themselves.

Relational database speak SQL. SQL is a standard interactive programming language for getting information from and updating a relational database. Although SQL itself is both an ANSI and an ISO standard, many database products support SQL with proprietary extensions to the standard language. MySQL's extensions to SQL are not proprietary, since MySQL's code is kept free by the GPL. SQL queries take the form of a command language that lets you select, insert, update, find out the location of data, and so forth.

❖ MySQL features:

- Very fast and much reliable for any type of application.
- Very lightweight application.
- Command line tool is very powerful and can be used to run SQL queries against database.

3.4.7. Developing Tool: Netbeans

3.4.7.1 Netbeans IDE

The NetBeans Platform allows applications to be developed from a set of modular software components called modules. A module is a Java archive file that contains Java classes written to interact with the NetBeans Open APIs and a manifest file that identifies it as a module. Applications built on modules can be extended by adding new modules. Since modules can be developed independently, applications based on the NetBeans platform can be easily and powerfully extended by third party developers. JDK APIs, and understanding them is not only crucial, but can also make programming with them a lot of fun. APIs, such as JavaBeans, JFC, RMI, JDBC, and so on.

3.4.7.2 Netbeans Platform

The NetBeans Platform is a reusable framework for simplifying the development of other desktop applications. When an application based on the NetBeans Platform is run, the platform's Main class is executed. Available modules are located, placed in an in-memory registry, and the modules' startup tasks are executed. Generally, a module's code is loaded into memory only as it is needed.

Applications can install modules dynamically. Any application can include the Update Center module to allow users of the application to download digitally-signed upgrades and new features directly into the running application. Installing an upgrade or a new release does not force users to download the entire application again.

The platform offers services common to desktop applications, allowing the developer to focus on the logic specific to his application. Among the features of the platform are:

- User interface management (e.g. menus and toolbars)
- User settings management
- Storage management (saving and loading any kind of data)
- Window management
- Wizard framework (supports step-by-step dialogs)

The Net Beans Mobility Pack is a tool for developing applications that run on mobile phones.

3.5. Data Flow Diagrams

The data flow diagram (DFD) is one of the most important tools used by the system analysts. Data flow diagrams are made up of number of symbols, which represent system components. Most data flow modeling methods use four kinds of symbols. These symbols are used to represent four kinds of system components. Processes, data store, data flows, external entities. Circles in DFD represent processes. Data flow represents by a thin line in the DFD and each store has a unique name and square or rectangle represents external entities. A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about

the timing of process or information about whether processes will operate in sequence or in parallel (which is shown on a flowcharts).

Basic DFD Symbols

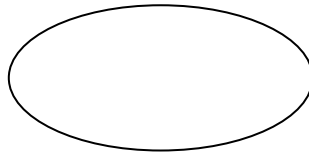
There are different notations to draw data flow diagrams (Yourdon & Coad and Gane&Sarson), defining different visual representations for processes, data stores, data flow, and external entities.

Arrows



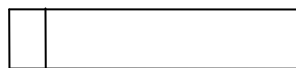
A data flow is a route, which enables packets of data to travel from one point to another. An arrow identifies data flow-data in motion. It is a pipeline through which information flows.

Process



A process represents transformation where incoming data flows are changed into outgoing data flows. A circle or a “bubble” represents a process that transforms incoming data flows into outgoing data flows.

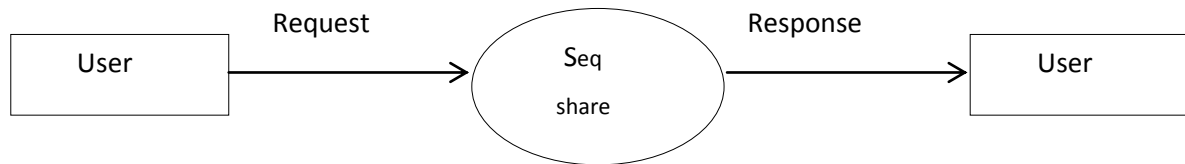
Data Store



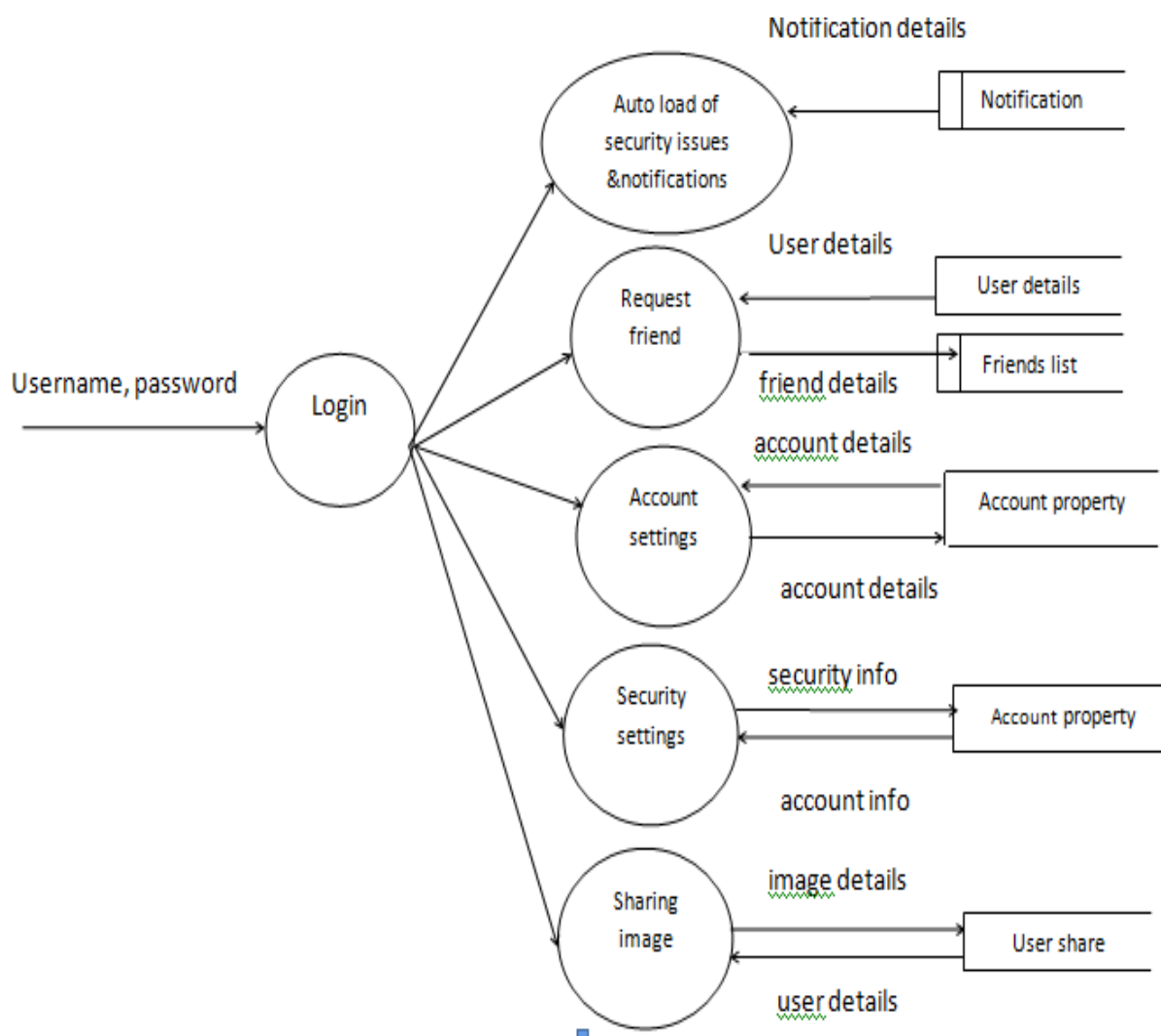
The DFD methodology is quite effective; especially when the required design is unclear and the user analyst need a notational language for communication. The DFD is easy to understand after a brief orientation. Several rules of thumb are used during drawing DFD's;

- Process should be named and numbered for easy reference.
- The direction of flow is from top to bottom and from left to right.
- When a process is exploded into lower-level details, they are numbered.
- The names of data stores, sources, and destinations are written in capital letter.

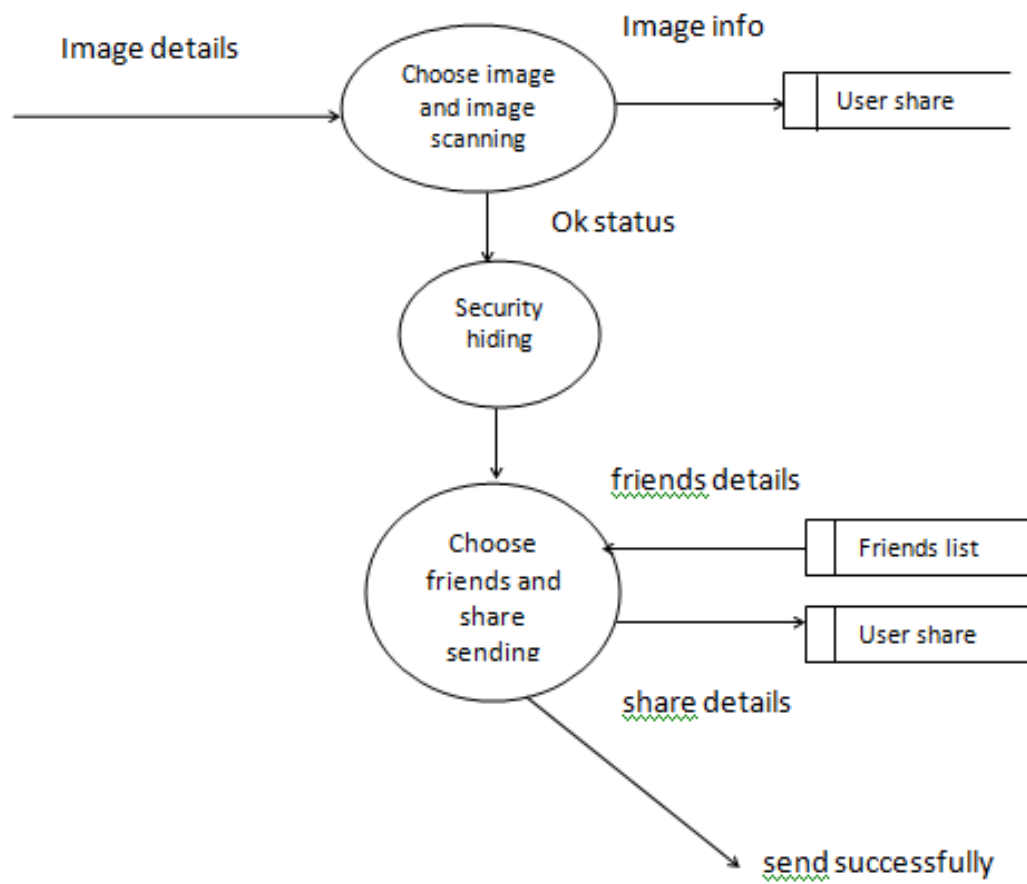
Level- 0



Level-1



LEVEL- 2 of sharing image



3.6. Structure Chart

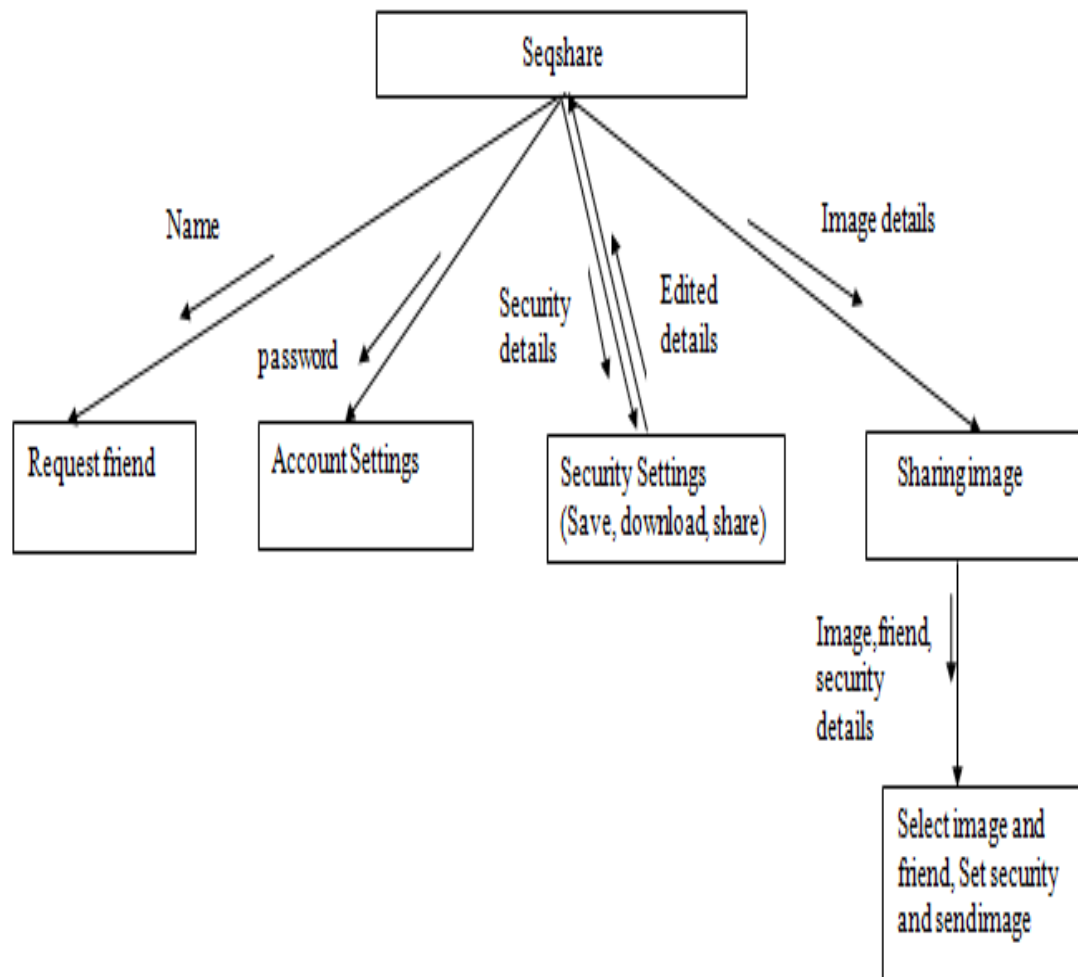
A Structure Chart (SC) in software engineering and organizational theory is a chart, which shows the breakdown of the configuration system to the lowest manageable levels. This chart is used in structured programming to arrange the program modules in a tree structure. Each module is represented by a box, which contains the module's name. The tree structure visualizes the relationships between the modules. Structure charts show the relation of processing modules in computer software. It is a design tool that visually displays the relationships between program modules. It shows which module within a system interacts and graphically depicts the data that are communicated between various modules.

A structure chart is a top-down modular design tool, constructed of squares representing the different modules in the system, and lines that connect them. The lines represent the connection or ownership between activities and sub activities as they are used in organization charts.

A Structure Chart Depicts

- The size and complexity of the system.
- The number of readily identifiable functions and modules within each function
- Whether each identifiable function is a manageable entity or should be broken down into smaller components.


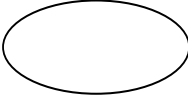
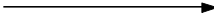


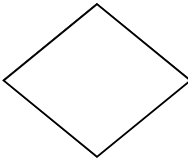
3.6.1. Structured Chart



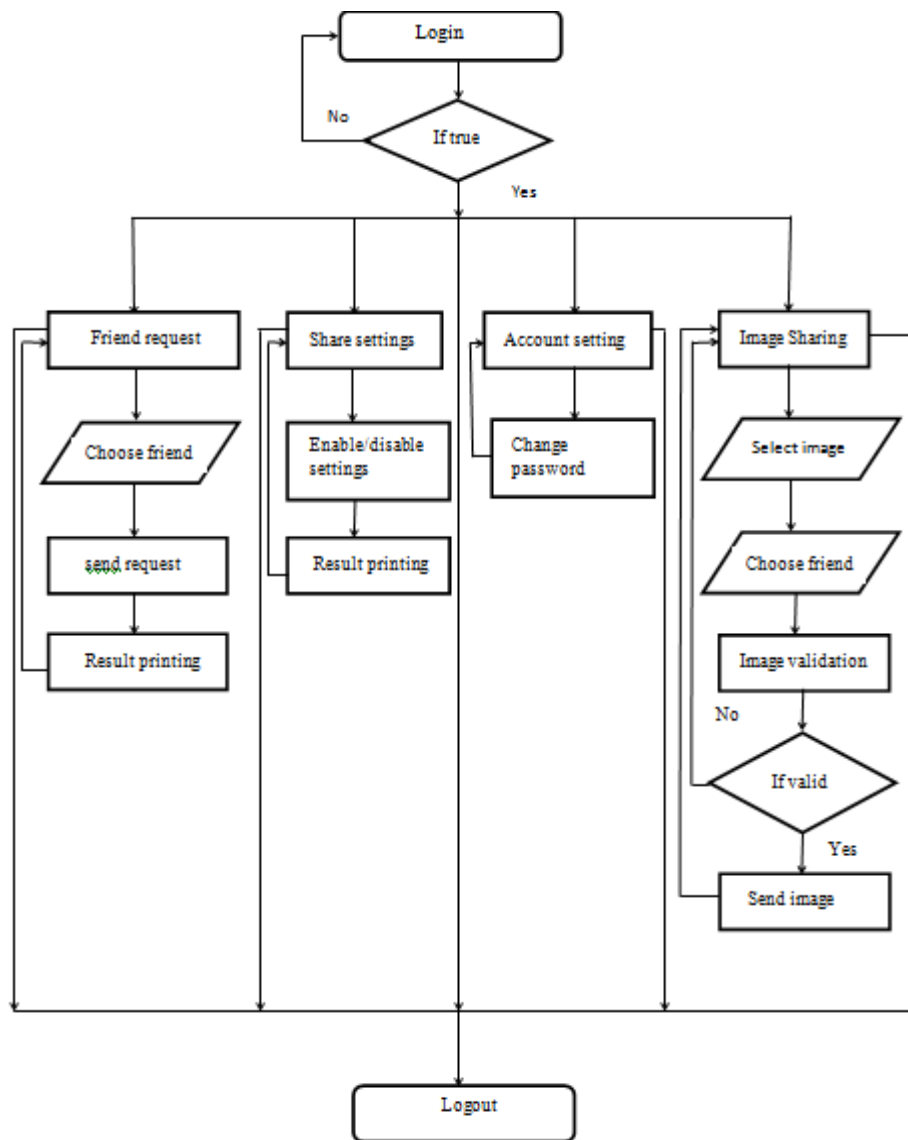
3.7. System Flow Chart

A flow chart is a graphical or symbolic representation of a process. Each step in the process is represented by a different symbol and contains a short description of the process step. The flow chart symbols are linked together with arrows showing the process flow direction. The flowchart is a means of visually presenting the flow of data through an information processing systems, the operations performed within the system and the sequence in which they are performed.

- The symbols uses in flow charts are:

Name	Symbol	Use in flow chart
Rounded Rectangle		It is the First and Last Symbol in the Flow Chart
Oval		Denotes the start and end of a program
Flow Line		Denotes the direction of logic flow in a program.
Parallelogram		Denotes either an input operation or an output operation.
Rectangle		Denotes a process to be carried out.
Diamond		Denotes a decision to be made.

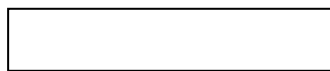
3.7.1. System Flow Chart



3.8. Menu Tree

Menu tree is also helpful for representing the simplified version of the system. Which is in the form of a tree structure? Here firstly the entire system taken as the main part, and then each modules coming in that system can be represented as the branches and their functions of each modules can be represented as same as the leaf in the tree. So menu tree is the hierarchical representation of entire system, so it is very helpful for identifying the system easily.

Symbols used in Menu tree

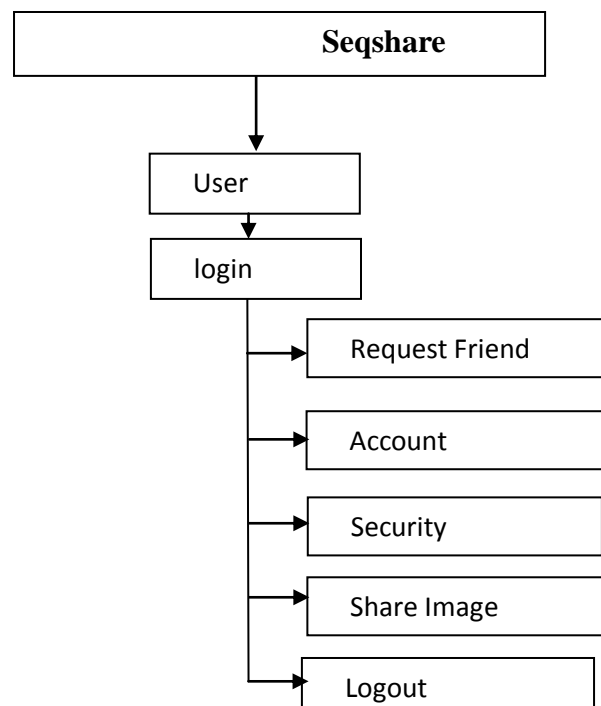


A rectangle represents the menus.



A line represents the connection.

3.8.1. Menu Tree



SYSTEM DESIGN

4. SYSTEM DESIGN

4.1 Module Description

A modular design reduces complexity, facilitates change and results in easier implementation by encouraging parallel development of different parts of a system. Modularity is the single attribute of software that allows a program to be intellectually manageable.

The proposed system is mainly having the following modules:

4.1.1. Login Module:

The registered user is required to login inorder to access the sites and it's contents using username and password.

4.1.2. User:

Here User can be also considered as an Administrator. The User has to provide accurate details in the registration form. The information will be helpful for communicating. The User can add Friends, view friend list, chat with friends and also can upload and share Image.

4.1.2.1. Adding Friends:

We can add friends by searching using username. There will be filter facility to make the searching easy for the user. We can search and request to the desired friend.

4.1.2.2. Image Uploading:

User can also upload image to the home page and can share the Image.

4.1.2.3. Image Sharing:

Image can be shared by the User to any friend he wish. Here we can change the security settings of the Image for preventing the Image from unauthorized access.

4.2 Input Output Design

4.2.1 Input Design

MEMBER LOGIN

Username	<input type="text"/>
Password	<input type="password"/>
	<input type="button" value="Login"/>

SIGN UP HERE

Name	<input type="text"/>
Birth day	<input type="text"/>
Email	<input type="text"/>
Contact	<input type="text"/>
Address	<input type="text"/>
Place	<input type="text"/>
Username	<input type="text"/>
Password	<input type="password"/>
Security question	--select--
Answer	<input type="text"/>
	<input type="button" value="Sign up"/>

Find your friends

Send request

USER PROFILE

Name	<input type="text"/>
Date of birth	<input type="text"/>
Contact	<input type="text"/>
Email	<input type="text"/>
Address	<input type="text"/>
Place	<input type="text"/>
Working at	<input type="text"/>

Update

4.2.2 Output Design

MEMBER LOGIN

Username	<input type="text" value="neethu@gmail.com"/>
Password	<input type="password" value="*****"/>

Login

Find your friends

Send request

ee

neethu
sreejith
geetha

USER PROFILE

Name	neethu
Date of birth	1992-11-11
Contact	9895475262
Email	neethu@gmail.com
Address	Kollam
Place	Kollam
Working at	Ernakulam
<div>Update</div>	

4.3. Database Design

The most important aspect of building an is the design of tables or database schema. The overall objective in the process of table design has been to treat data as an organization of data in the database.

4.3.1. Table Name : account_property

Sl.No.	Field Name	Data Type	Description
1	userID(Primarykey)	Bigint	User Id
2	Share	Int	Share Status
3	Download	Int	Download status
4	Save	Int	Save Status
5	secret key	Varchar(10)	Secret Key

4.3.2. Table Name : friends_list

Sl.No.	Field Name	Data Type	Description
1	uid(Primary key)	Bigint	User Id
2	req_id	Int	Request Id
3	req_status	Int	Request Status

4.3.3. Table Name : notifications

Sl.No.	Field Name	Data Type	Description
1	sln(Primarykey)	Bigint	Serial Number
2	Date	Date	Date
3	userID	Bigint	User ID
4	Message	Varchar(500)	MessageContent
5	Type	Varchar(10)	Type of Message

4.3.4. Table Name : share_id_record

Sl.No.	Field Name	Data Type	Description
1	shareID (Primary key)	Bigint	Share Id
2	userID	Int	User Id
3	view_status	Int	View Status
4	Type	Varchar(10)	Type of share

4.3.5. Table Name : user_details

Sl.No.	Field Name	Data Type	Description
1	userID(Primarykey)	Bigint	User Id
2	Name	Varchar(30)	Name of User
3	Address	Varchar(50)	Address of User
4	Place	Varchar	Place of User
5	Contact	Varchar(12)	Contact Number
6	Email	Varchar(50)	Email ID
7	Dob	Varchar(15)	Date of Birth
8	Username	Varchar(10)	Username
9	Password	Varchar(10)	Password
10	secQstn	Varchar(50)	Security Question
11	secAns	Varchar(20)	Security Answer
12	Online	Int	Online Stauts

4.3.6 Table Name : user_shares

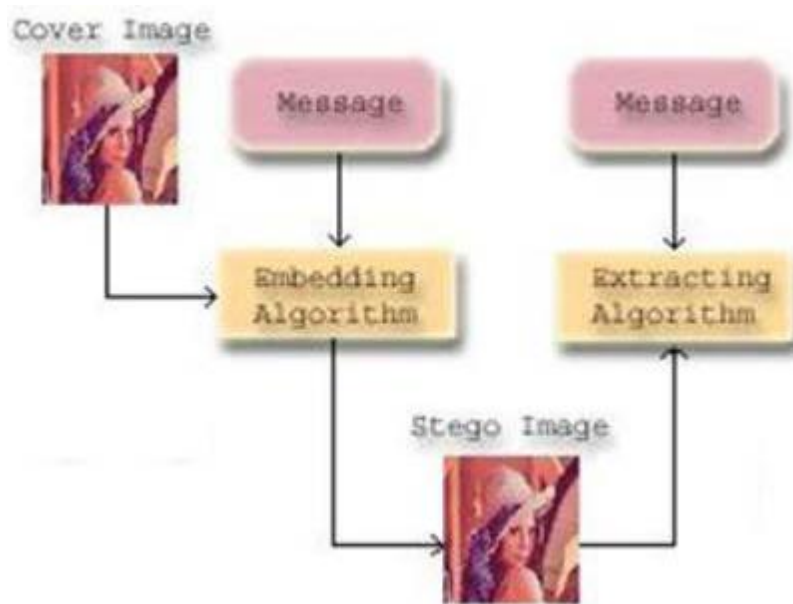
Sl.No.	Field Name	Data Type	Description
1	shareID (Primary key)	Bigint	Share Id
2	share Date	Varchar(20)	Share Date
3	shareTo	Varchar(30)	User to share
4	Path	Varrchar(50)	Path of share
5	shareBy	Int	User who share
6	share	Int	Share Status
7	Download	Int	Download Status
8	Save	Int	Save Status
9	accessType	Varchar(10)	Type of Share

OVERVIEW

5. OVERVIEW

5.1. Image Steganography Terminologies:

- **Cover-Image:** Original image which is used as a carrier for hidden information.
- **Message:** Actual information which is used to hide into images. Message could be a plain text or some other image.
- **Stego-Image:** After embedding message into cover image is known as stego-image.
- **Stego-Key:** A key is used for embedding or extracting the messages from cover-images and stego-images.



Generally image steganography is method of information hiding into cover-image and generates a stego-image. This stego-image then sent to the other party by known medium, where the third party does not know that this stego-image has hidden message. After receiving stego-image hidden message can simply be extracted with or without stego-key (depending on embedding algorithm) by the receiving end [21]. Basic diagram of image steganography is shown in Figure 2 without stego-key, where embedding algorithm required a cover image with message for embedding procedure. Output of embedding algorithm is a stego-image which simply sent to extracting algorithm, where extracted algorithm unhides the message from stego-image.

5.2. Steganographic Measures:

- **High Capacity:** Maximum size of information can be embedded into image.
- **Perceptual Transparency:** After hiding process into cover image, perceptual quality will be degraded into stego-image as compare to cover-image.

- **Robustness:** After embedding, data should stay intact if stego-image goes into some transformation such as cropping, scaling, filtering and addition of noise.
- **Temper Resistance:** It should be difficult to alter the message once it has been embedded into stego-image.
- **Computation Complexity:** How much expensive it is computationally for embedding and extracting a hidden message?

SYSTEM ARCHITECTURE

6. SYSTEM ARCHITECTURE

6.2.1 Triple Data Encryption Standard (Triple-DES)

Triple DES (aka 3DES, 3-DES, TDES) is based on the DES (Data Encryption Standard) algorithm; therefore it is very easy to modify existing software to use Triple DES.

Triple DES (3DES) is the common name for the Triple Data Encryption Algorithm (TDEA or Triple DEA) symmetric-key block cipher, which applies the Data Encryption Standard (DES) cipher algorithm three times to each data block.

The original DES cipher's key size of 56 bits was generally sufficient when that algorithm was designed, but the availability of increasing computational power made brute-force attacks feasible.

Triple DES provides a relatively simple method of increasing the key size of DES to protect against such attacks, without the need to design a completely new block cipher algorithm.

It also has the advantage of proven reliability and a longer key length that eliminates many of the attacks that can be used to reduce the amount of time it takes to break DES. However, even this more powerful version of DES may not be strong enough to protect data for very much longer. As such, the DES algorithm itself has become obsolete and is in need of replacement.

The National Institute of Standards and Technology (NIST) ratified the Advanced Encryption Standard (AES) as a replacement for DES. NIST endorsed Triple DES as an interim standard to be used until AES was finished. Although AES is at least as strong as Triple DES, it is significantly faster. Many security systems support both Triple DES and AES. AES is the default algorithm, while Triple DES is often maintained for backward compatibility.

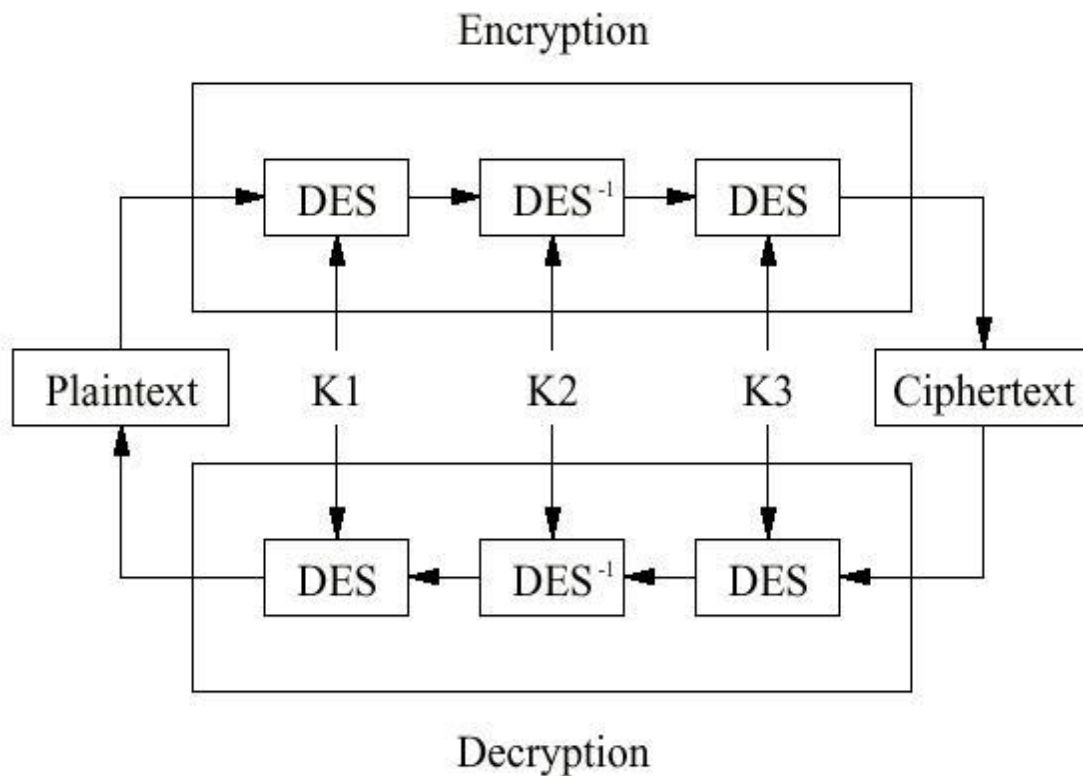
6.2.2 Triple DES Algorithm

Triple DES is another mode of DES operation. It takes three 64-bit keys, for an overall key length of 192 bits. In Stealth, you simply type in the entire 192-bit (24 character) key rather than entering each of the three keys individually. The Triple DES DLL then breaks the user-provided key into three subkeys, padding the keys if necessary so they are each 64 bits long. The procedure for encryption is exactly the same as regular DES, but it is repeated three times, hence the name Triple DES. The

data is encrypted with the first key, decrypted with the second key, and finally encrypted again with the third key.

Triple DES runs three times slower than DES, but is much more secure if used properly. The procedure for decrypting something is the same as the procedure for encryption, except it is executed in reverse. Like DES, data is encrypted and decrypted in 64-bit chunks. Although the input key for DES is 64 bits long, the actual key used by DES is only 56 bits in length. The least significant (right-most) bit in each byte is a parity bit, and should be set so that there are always an odd number of 1s in every byte. These parity bits are ignored, so only the seven most significant bits of each byte are used, resulting in a key length of 56 bits. This means that the effective key strength for Triple DES is actually 168 bits because each of the three keys contains 8 parity bits that are not used during the encryption process.

6.2.3 Triple DES Algorithm Structure



6.2.4 Triple DES Modes

❖ Triple ECB (Electronic Code Book)

- This variant of Triple DES works exactly the same way as the ECB mode of DES.
- This is the most commonly used mode of operation.

❖ Triple CBC (Cipher Block Chaining)

- This method is very similar to the standard DES CBC mode.
- As with Triple ECB, the effective key length is 168 bits and keys are used in the same manner, as described above, but the chaining features of CBC mode are also employed.
- The first 64-bit key acts as the Initialization Vector to DES.
- Triple ECB is then executed for a single 64-bit block of plaintext.
- The resulting ciphertext is then XORed with the next plaintext block to be encrypted, and the procedure is repeated.
- This method adds an extra layer of security to Triple DES and is therefore more secure than Triple ECB, although it is not used as widely as Triple ECB.

6.2.5. Algorithm

Triple DES uses a "key bundle" that comprises three DES keys, K_1 , K_2 and K_3 , each of 56 bits (excluding parity bits). The encryption algorithm is:

$$\text{ciphertext} = E_{K_3}(D_{K_2}(E_{K_1}(\text{plaintext})))$$

I.e., DES encrypt with K_1 , DES *decrypt* with K_2 , then DES encrypt with K_3 .

Decryption is the reverse:

$$\text{plaintext} = D_{K_1}(E_{K_2}(D_{K_3}(\text{ciphertext})))$$

I.e., decrypt with K_3 , *encrypt* with K_2 , then decrypt with K_1 .

Each triple encryption encrypts one block of 64 bits of data.

In each case the middle operation is the reverse of the first and last. This improves the strength of the algorithm when using keying option 2, and provides backward compatibility with DES with keying option 3.

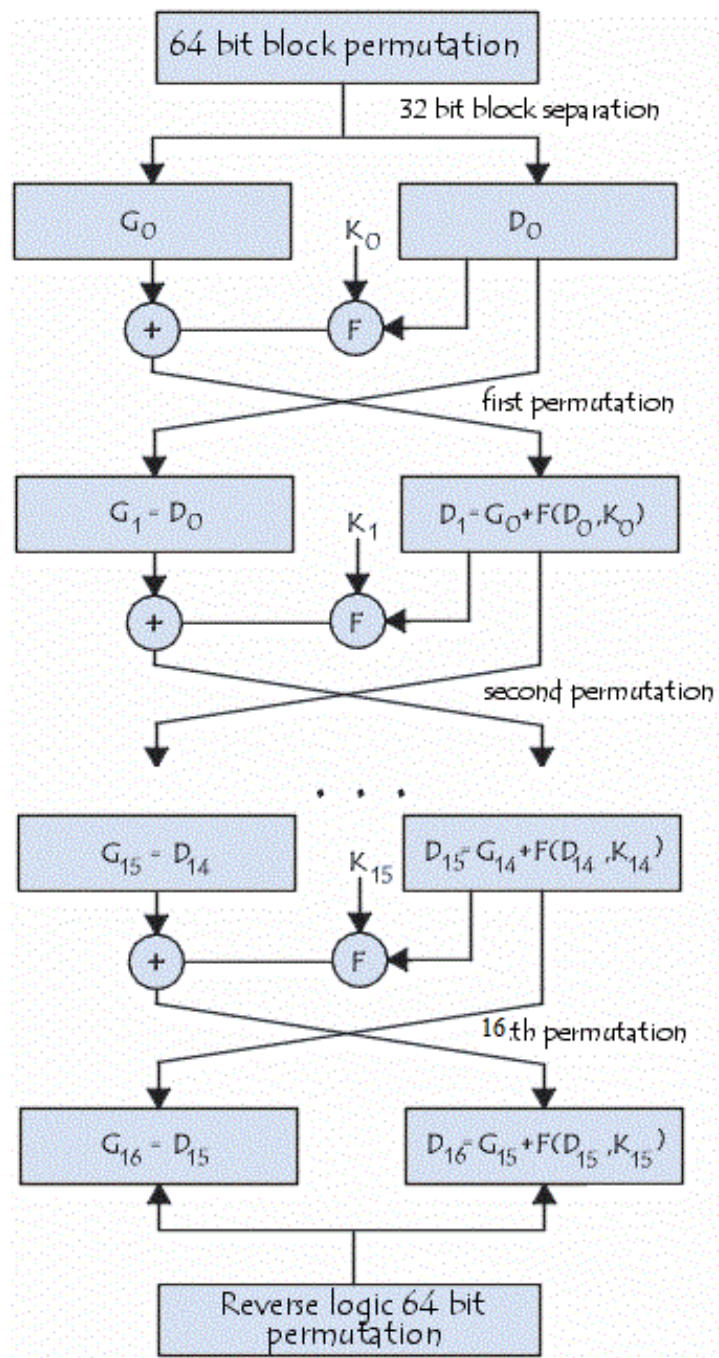
❖ Algorithm Overview

Triple Data Encryption Algorithm	
General	
First published	1998 (ANS X9.52)
Derived from	DES
Cipher detail	
Key sizes	168, 112 or 56 bits (keying option 1, 2, 3 respectively)
Block sizes	64 bits
Structure	Feistel network
Rounds	48 DES-equivalent rounds
Best public cryptanalysis	
Lucks: 2^{32} known plaintexts, 2^{113} operations including 2^{90} DES encryptions, 2^{88} memory;	
Biham: find one of 2^{28} target keys with a handful of chosen plaintexts per key and 2^{54} encryptions	

❖ DES Modes of Operation

The DES algorithm turns a 64-bit message block M into a 64-bit cipher block C . If each 64-bit block is encrypted individually, then the mode of encryption is called Electronic Code Book (ECB) mode. There are two other modes of DES encryption, namely Chain Block Coding (CBC) and Cipher Feedback (CFB), which make each cipher block dependent on all the previous messages blocks through an initial XOR operation.

6.2.6. DES Algorithm Illustration

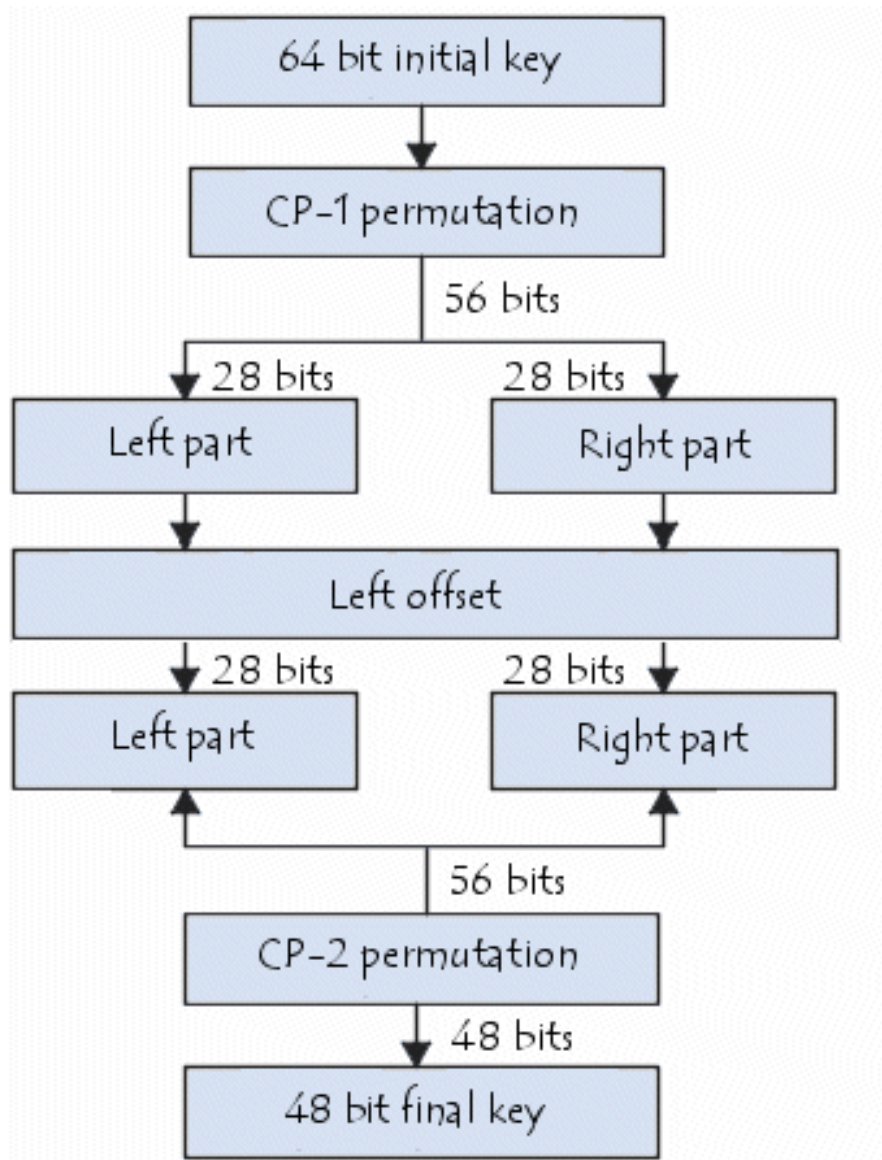


In each round the left half is XORed with the left half data and the result of 'f' function. The resulted data is the new right half. And the left half is the same data as previous right half.

For finding the result of 'f' function, the 32 bit data is converted into 48 bits. For converting the data 32 bit data into 48 bit the 'E' table is used. Bit positions are arranged according to the table and we get the 48 bit data. This 48 bit data is XORed with 48 bit key. The resulting data is 48 bit of 8 groups having 6 bits. These 8 boxes are considered as 8 'S' boxes. From this 8 'S' boxes of 6 bits produces 4 bit output. The first and last bits of this 'S' box is used for finding the row of the 'S' table. And the remaining 4 bits are used for finding the column for finding a particular value. Then this value is converted into binary, have for digit. That is why we say that 4 bit output from 6 bit input. The 4 bit output of each 'S' boxes are combined. Then the resulted 32 bit output is permuted according to the table. The permuted output is XORed with the previous left half and produces the right half. Then this process is continued till the 16 rounds are completed.

6.2.7 Key Generation

The randomly generated 56bit key is divided into two 28bit halves. Each having 28bit sub keys. These 28 bit sub keys are bit shifted. The first, second, ninth and sixteenth rounds have only one shift and others have two shift. Each half is shifted and this shifted sub keys are combined to produce the keys for each round. From this combined key, it is reduced to 48 bit key and is produced according to the table by arranging the bits in 56 bit key. Then this reduced 48 bit key is used in each rounds for encryption. For decryption, it is (keys) reversed in order.



After encryption, the encrypted data is embedded within the cover image. Data hiding is a method of hiding secret messages into a cover-media such that an unintended observer will not be aware of the existence of the hidden messages.

The following formula provides a very generic description of the pieces of the steganographic process:

$$\text{cover_medium} + \text{hidden_data} + \text{stego_key} = \text{stego_medium}$$

In this context, the `cover_medium` is the file in which we will hide the `hidden_data`, which may also be encrypted using the `stego_key`. The resultant file is the `stego_medium` (which will, of course, be the same type of file as the `cover_medium`). The `cover_medium` (and, thus, the `stego_medium`) are typically image or audio files. In

this article, I will focus on image files and will, therefore, refer to the `cover_image` and `stego_image`.

An image file is merely a binary file containing a binary representation of the color or light intensity of each picture element (pixel) comprising the image. Images typically use either 8-bit or 24-bit color. When using 8-bit color, there is a definition of up to 256 colors forming a palette for this image, each color denoted by an 8-bit value. A 24-bit color scheme, as the term suggests, uses 24 bits per pixel and provides a much better set of colors. In this case, each pixel is represented by three bytes, each byte representing the intensity of the three primary colors red, green, and blue (RGB), respectively. The size of an image file, then, is directly related to the number of pixels and the granularity of the color definition. A typical 640x480 pixel image using a palette of 256 colors would require a file about 307 KB in size ($640 \cdot 480$ bytes), whereas a 1024x768 pixel high-resolution 24-bit color image would result in a 2.36 MB file ($1024 \cdot 768 \cdot 3$ bytes).

The simplest approach to hiding data within an image file is called least significant bit (LSB) insertion. In this method, we can take the binary representation of the `hidden_data` and overwrite the LSB of each byte within the `cover_image`. If we are using 24-bit color, the amount of change will be minimal and indiscernible to the human eye. As an example, suppose that we have three adjacent pixels (nine bytes) with the following RGB encoding:

```
10010101 00001101 11001001
10010110 00001111 11001010
10011111 00010000 11001011
```

Now suppose we want to "hide" the following 9 bits of data (the hidden data is usually compressed prior to being hidden): 101101101. If we overlay these 9 bits over the LSB of the 9 bytes above, we get the following (where bits in **bold** have been changed):

```
10010101 0000110 11001001
1001011 0000111 1100101
10011111 00010000 11001011
```

Note that we have successfully hidden 9 bits but at a cost of only changing 4, or roughly 50%, of the LSBs.

This description is meant only as a high-level overview. Similar methods can be applied to 8-bit color but the changes, as the reader might imagine, are more dramatic. Gray-scale images, too, are very useful for steganographic purposes. One potential problem with any of these methods is that they can be found by an adversary who is looking. In addition, there are other methods besides LSB insertion with which to insert hidden information.

3DES

Key Generation

The randomly generated 56bit key is divided into two 28bit halves. Each having 28bit sub keys. These 28 bit sub keys are bit shifted. The first, second, ninth and sixteenth rounds have only one shift and others have two shift. Each half is shifted and this shifted sub keys are combined to produce the keys for each round. From this combined key, it is reduced to 48 bit key and is produced according to the table by arranging the bits in 56 bit key. Then this reduced 48 bit key is used in each rounds for encryption. For decryption, it is (keys) reversed in order.

DES

The message that we want to encrypt is converted into hexadecimal and then is to binary values having 64 bit long. If the message is not long as 64 bit, then padded zeros at the end. The message is then permuted in IP(Initial Permutation) is based on the table and arranging the bit positions according to the table. Then this permuted data is divided into two 32 bit halves. These halves are used in DES.

In each round the left half is XORed with the left half data and the result of 'f' function. The resulted data is the new right half. And the left half is the same data as previous right half.

For finding the result of 'f' function, the 32 bit data is converted into 48 bits. For converting the data 32 bit data into 48 bit the 'E' table is used. Bit positions are arranged according to the table and we get the 48 bit data. This 48 bit data is XORed with 48 bit key. The resulting data is 48 bit of 8 groups having 6 bits. These 8 boxes are considered as 8 'S' boxes. From this 8 'S' boxes of 6 bits produces 4 bit output. The first and last bits of this 'S' box is used for finding the row of the 'S' table. And the remaining 4 bits are used for finding the column for finding a particular value. Then this

value is converted into binary, have for digit. That is why we say that 4 bit output from 6 bit input. The 4 bit output of each 'S' boxes are combined. Then the resulted 32 bit output is permuted according to the table. The permuted output is XORed with the previous left half and produces the right half. Then this process is continued till the 16 rounds are completed.

CODING

7. CODING

7.1. reqPr.jsp

```
<% @page import="dbconnection.QueryGenerator"%>
<jsp:useBean id="con" class="dbconnection.DbConnection"/>
<%
    String opt=request.getParameter("opt"); //out.println(opt);
    String uid=request.getParameter("uid");

    if(opt.equals("REJ")){
        if(con.putData(QueryGenerator.REJECT_REQUEST+uid)>0){
            %>
            <script>
                alert("Friend request rejected..");
                window.location=" ../user/home.jsp";
            </script>
            <%
        }else{
            %>
            <script>
                alert("Friend request rejection failed..");
                window.location=" ../user/home.jsp";
            </script>
            <%
        }
    }

    if(opt.equals("ACP")){
        if(con.putData(QueryGenerator.ACCEPT_REQUEST+uid)>0){
            %>
            <script>
                alert("Friend request Accepted..");
                window.location=" ../user/home.jsp";
            </script>
            <%
        }else{
```

```

    %>
    <script>
        alert("Action failed..");
        window.location="../user/home.jsp";
    </script>
    <%
    }
    }
    %>

```

7.2. share_settings_pr.jsp

```

<% @page import="java.util.Vector"%>
<% @page import="java.util.Iterator"%>
<jsp:useBean id="settings" class="beanfiles.Settings"/>
<jsp:setProperty name="settings" property="*" />

<%
String category=request.getParameter("cat");
settings.setUid(session.getAttribute("uid").toString());

if(category.equals("1")){
    if(settings.putData(1)>0){
        %>
        <script>
            alert("Setting has been saved..")
            window.location="../user/home.jsp";
        </script>
        <%
    }else{
        %>
        <script>
            alert("Updation failed.. try again later..")
            window.location="../user/home.jsp";
        </script>
        <%
    }
}

```

```

    }

    if(category.equals("2")){
        Iterator it=settings.getData(2).iterator();
        if(it.hasNext()){
            Vector v=(Vector)it.next();

            if(v.get(0).toString().equals(settings.getCurpwd())           &&
settings.getNewpwd().equals(settings.getConpwd())){

                if(settings.putData(3)>0){
                    %>
                    <script>
                        alert("Password has been changed..");
                        window.location=" ../home.jsp";
                    </script>
                    <%
                    }else{
                    %>
                    <script>
                        alert("Updation failed.. try again later..");
                        window.location=" ../home.jsp";
                    </script>
                    <%
                    }
                }
            }
        }
    }
    %>

```

7.3. shareprocessing.jsp

```

<% @page import="java.net.NetworkInterface"%>
<% @page import="java.net.InetAddress"%>
<% @page import="security.RetrieveData"%>
<% @page import="dbconnection.QueryGenerator"%>
<% @page import="java.util.Date"%>
<% @page import="common.Constants"%>

```



```

<% @page import="java.io.File"%>
<% @page import="java.util.Vector"%>
<% @page import="java.util.Iterator"%>
<jsp:useBean id="settings" class="beanfiles.Settings"/>
<jsp:setProperty name="settings" property="*" />
<jsp:useBean id="embed" class="security.EmbedData"/>
<jsp:useBean id="con" class="dbconnection.DbConnection"/>

<%
    Date dt=new Date();
    String date_time="_"+dt.toGMTString();
    date_time=(date_time.substring(0, date_time.length()-4)).replace(" ", "").replace(":",
    "");
    settings.setUid(session.getAttribute("uid").toString());
    String []shareIDS=settings.getIds().split(" ");
    File orig=new
File(Constants.PROJECT_PATH+"tempfiles/"+session.getAttribute("uid").toString()+".j
pg");
    File dest=new
File(Constants.PROJECT_PATH+"image_shares/"+session.getAttribute("uid").toString()
+date_time+".jpg");
    Vector v=null;
    String uploadConfig="",shares="";
    String sharedBy=session.getAttribute("uid").toString();

    //checking for data in the image to be shared
    RetrieveData data=new RetrieveData();
    String existingData=data.RetrieveData(orig);
    if(existingData.substring(0, 1).equals("1") || existingData.equals("#NODATA#")){

        //setting configuration data
        Iterator it=settings.getData(4).iterator();
        if(it.hasNext()){
            v=(Vector)it.next();

```

```
uploadConfig=v.get(0).toString()+":"+v.get(1).toString()+":"+v.get(2).toString()+":"+shareBy+ "::";
```

```
    settings.setShare(v.get(0).toString());
```

```
    settings.setDownload(v.get(1).toString());
```

```
    settings.setSave(v.get(2).toString());
```

```
    for(int i=0;i<shareIDS.length;i++){
```

```
        shares=shareIDS[i];
```

```
        uploadConfig=uploadConfig+","+shareIDS[i];
```

```
    }uploadConfig=uploadConfig+ "::";
```

```
}
```

```
//embedding data to image
```

```
embed.embedMessage(orig,dest,uploadConfig,3);
```

```
//database manipulation
```

```
settings.setPath("image_shares/"+session.getAttribute("uid").toString()+date_time+".jpg" );
```

```
    if(settings.putData(5)>0){
```

```
        String qry="";
```

```
        int status=0;
```

```
        for(int i=0;i<shareIDS.length;i++){
```

```
            qry="insert          into          share_id_record          (shareID,userID)
```

```
values((" +QueryGenerator.MAX_SHARE_ID+"),"+shareIDS[i]+")";
```

```
            System.out.println(qry);
```

```
            if(con.putData(qry)>0){
```

```
                status=1;continue;
```

```
            }else{
```

```
                status=0;break;
```

```
        }
```

```
    }
```

```
    if(status==1){
```

```

%>
<script>
    alert("Your share has been send..");
    window.location="../user/home.jsp";
</script>
<%
}else{
    qry="delete          from          user_shares          where
shareID=(" +QueryGenerator.MAX_SHARE_ID+"));
    con.putData(qry);
%>
<script>
    alert("Sending failed..");
    window.location="../user/home.jsp";
</script>
<%
}
}
}else{
    String informTo=existingData.substring(7,8);
    String message="";
//    Iterator it=con.getData(QueryGenerator.GET_SHARE_DATE).iterator();
//    if(it.hasNext()){
//        //getting ip address and mac address of the host
//        InetAddress ip = InetAddress.getLocalHost();
//        NetworkInterface network = NetworkInterface.getByInetAddress(ip);
//        byte[] mac = network.getHardwareAddress();
//        StringBuilder sb = new StringBuilder();
//
//        for (int i = 0; i < mac.length; i++) {
//            sb.append(String.format("%02X%s", mac[i], (i < mac.length - 1) ? "-" : ""));
//
//        }
//
//        message="Attempt to share the private message send to yor friend was blocked
successfully..<br/><br/> "

```

```

        + "Details:"
        + "<h5 style='color:red'> System
IP:&nbsp;&nbsp;&nbsp;&nbsp;<i>"+ip.toString()+"</i></h5>"
        + "<h5 style='color:red'> MAC
Address:&nbsp;&nbsp;&nbsp;&nbsp;<i>"+sb.toString()+"</i></h5>"
        + "<h5 style='color:red'> Date and
time:&nbsp;&nbsp;&nbsp;&nbsp;<i>"+dt.toGMTString()+"</i></h5>";
// out.println(message);
String shareNotification="insert into notifications(date,userid,message,type) "
        + "values((SELECT
NOW()),"+informTo+", '"+message+"','SEQ')";
con.putData(shareNotification);

%>
<script>
    alert("This image cannot be shared..");
    window.location="../user/home.jsp";
</script>
<%
// }
}
%>

```

7.4. findnewfriends.jsp

```

<% @page import="java.util.Vector"%>
<% @page import="java.util.Iterator"%>
<% @page import="dbconnection.QueryGenerator"%>
<!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=utf-8" />
    <title>Bird Store Online, CSS Template, Free Download</title>
    <meta name="keywords" content="Bird Store Online, CSS Template, Free Download"
/>

```

```

<meta name="description" content="Bird Store Online - Download Free CSS
Template" />
<link href="../css/templatemo_style.css" rel="stylesheet" type="text/css" />
<link rel="stylesheet" href="../css/reveal.css"/>
<script type="text/javascript" src="../css/jquery-1.6.min.js"></script>
<script type="text/javascript" src="../css/jquery.reveal.js"></script>
<jsp:useBean id="con" class="dbconnection.DbConnection"/>
<link rel="stylesheet" href="../js/jquery.mobile-1.2.0.min.css">
<script src="../js/jquery-1.8.2.min.js"></script>
<script src="../js/jquery.mobile-1.2.0.min.js"></script>

<script>
    function setValue(uid){
        document.getElementById("uid").value=uid;
    }
    function processRequest(){

window.location="../process/frndRqstPr.jsp?uid="+document.getElementById("uid").value;
    }
</script>
</head>
<body>
<div id="templatemo_container">
    <div id="templatemo_header"></div>
        <!-- end of header -->

    <div id="templatemo_menu">
        <ul>
            <li><a href="home.jsp" class="current">Home</a></li>
            <li><a href="#">Notifications</a></li>
            <li><a href="#">Messages</a></li>
            <li><a href="#">Friends</a></li>
            <li><a href="#">My Zone</a></li>
            <li><a href="#" class="last">Contact Us</a></li>

```

```

    </ul>
</div> <!-- end of menu -->

<div id="templatemo_content" style="color: #000">

    <div id="content_right">
        <div class="margin_bottom_20">&nbsp;</div>

        <div class="right_col_section_w420">
            <center><br/><h3>Find your friends </h3>
<!--      <div data-role="page">          -->
            <div data-role="content">
                <div class="outer_new">
                    <form name="fndrqst" onsubmit="processRequest();">
                        <input type="hidden" name="uid" id="uid" />
                        <input type="submit" value="Send request"/>
                    </form>
                    <ul data-role="listview" data-inset="true" data-filter="true">
                        <%
                            Iterator
it=con.getData(QueryGenerator.VIEW_ALL_MEMBERS).iterator();
                            while(it.hasNext()){
                                Vector v=(Vector)it.next();
                                out.println("<li><a      href=\"#{\"      onClick=\"setvalue(\"'+v.get(0)
+\"');\">"+v.get(1)+\"</a></li>");
                                    }
                                %>
                            </ul>
                        </div>
                    </div>

                </center>

                <div class="cleaner">&nbsp;</div>
            </div> <!-- end of right_col_section_w420 -->

```

```

</div> <!-- end of content right -->

<div class="cleaner">&nbsp;</div>
</div> <!-- end of container -->
<div id="templatemo_content_bottom">&nbsp;</div>
<!-- Free CSS Templates by www.TemplateMo.com -->
<div id="templatemo_footer">
    Copyright 2014 <a href="#">LCC Computer Education</a>
</div> <!-- end of footer -->
<!-- end of container -->
</body>
</html>

```

7.5. Share_image.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=utf-8" />
    <title>Bird Store Online, CSS Template, Free Download</title>
    <meta name="keywords" content="Bird Store Online, CSS Template, Free Download"
/>
    <meta name="description" content="Bird Store Online - Download Free CSS
Template" />
    <link href="css/templatemo_style.css" rel="stylesheet" type="text/css" />
    <link rel="stylesheet" href="css/reveal.css"/>
    <script type="text/javascript" src="css/jquery-1.6.min.js"></script>
    <script type="text/javascript" src="css/jquery.reveal.js"></script>

<script>
    function loadMenu(opt){

```

```

obj=(window.XMLHttpRequest)?                                new
XMLHttpRequest():((window.ActiveXObject)?                  new
ActiveXObject(Microoft.XMLHTTP):null);
    if(obj!=null){
        obj.onreadystatechange=function(){
            if(obj.readyState==4&obj.status==200){ //alert(obj.responseText);
                document.getElementById('statusdiv').innerHTML=obj.responseText;
            }
        };
        if(opt==1){
            obj.open('post','user/shareImage.jsp',true);
        }else if(opt==2){
            obj.open('post','#',true);
        }else if(opt==3){
            obj.open('post','#',true);
        }else if(opt==4){
            obj.open('post','#',true);
        }
        obj.send(null);
    }
}
</script>

```

```

<script type="text/javascript">
    function PreviewImage() {
        var oFReader = new FileReader();
        oFReader.readAsDataURL(document.getElementById("uploadImage").files[0]);

        oFReader.onload = function (oFREvent) {
            document.getElementById("uploadPreview").src = oFREvent.target.result;
        };
    };
</script>
</head>
<body>

```



```
<div id="templatemo_container">
  <div id="templatemo_header"></div>
    <!-- end of header -->

  <div id="templatemo_menu">
    <ul>
      <li><a href="#" class="current">Home</a></li>
      <li><a href="#">Notifications</a></li>
      <li><a href="#">Messages</a></li>
      <li><a href="#">Friends</a></li>
      <li><a href="#">My Zone</a></li>
      <li><a href="#" class="last">Contact Us</a></li>
    </ul>
  </div> <!-- end of menu -->

  <div id="templatemo_content">

    <div id="content_left">

      <div class="content_left_section">
        <div class="content_left_section_title">Control Panel</div>
        <div class="content_left_section_content">
          <ul>
            <li><a href="#">Profile</a></li>
            <li><a href="#">Friends</a></li>
            <li><a href="#">Chats</a></li>
          </ul>
          <div class="cleaner">&nbsp;</div>
        </div>
        <div class="cleaner">&nbsp;</div>
        <div class="content_left_section_bottom">&nbsp;</div>
      </div>

      <div class="margin_bottom_20">&nbsp;</div>
      <div class="content_left_section">
        <div class="content_left_section_title">Documents</div>
```



```

        <td><textarea name="shareto" id="shareto" rows="10"></textarea>
        <br/><input type="button" value="Continue"/></td>
    </tr>
</table> <br/>
<form name="uploadimage" action="share_image.jsp" method="post">
    <table border="3">
        <tr>
            <td>Browse File</td>
            <td><input type="file" name="file" id="uploadImage"
onchange="PreviewImage();" /></td>
        </tr>
        <tr>
            <td colspan="2"><center><img id="uploadPreview" width="220"
height="180"/></center></td>
        </tr>
        <tr>
            <td colspan="2"><input type="submit" name="uploadimg"
value="Share Image"></td>
        </tr>
    </table>
</form>
</div> <!-- end of content right -->

<div class="cleaner">&nbsp;</div>
</div> <!-- end of container -->

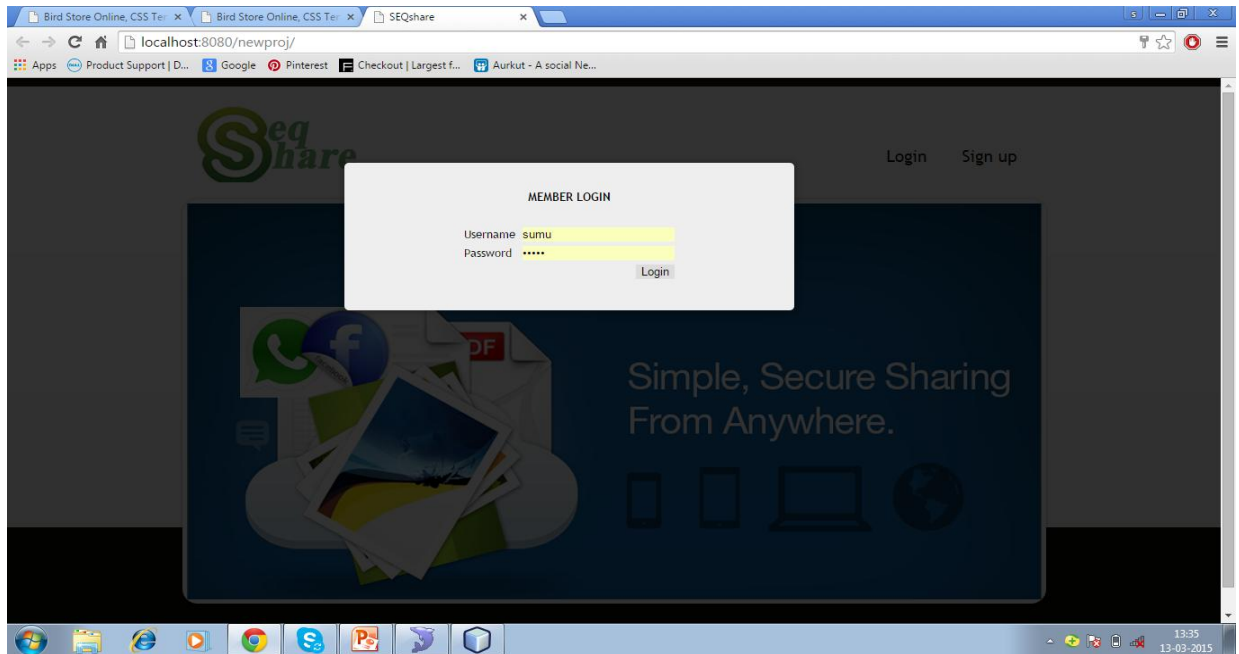
<div id="templatemo_content_bottom">&nbsp;</div>
<!-- Free CSS Templates by www.TemplateMo.com -->
<div id="templatemo_footer">
    Copyright 2014 <a href="#">LCC Computer Education</a>
</div> <!-- end of footer -->
</div> <!-- end of container -->
</body>
</html>

```

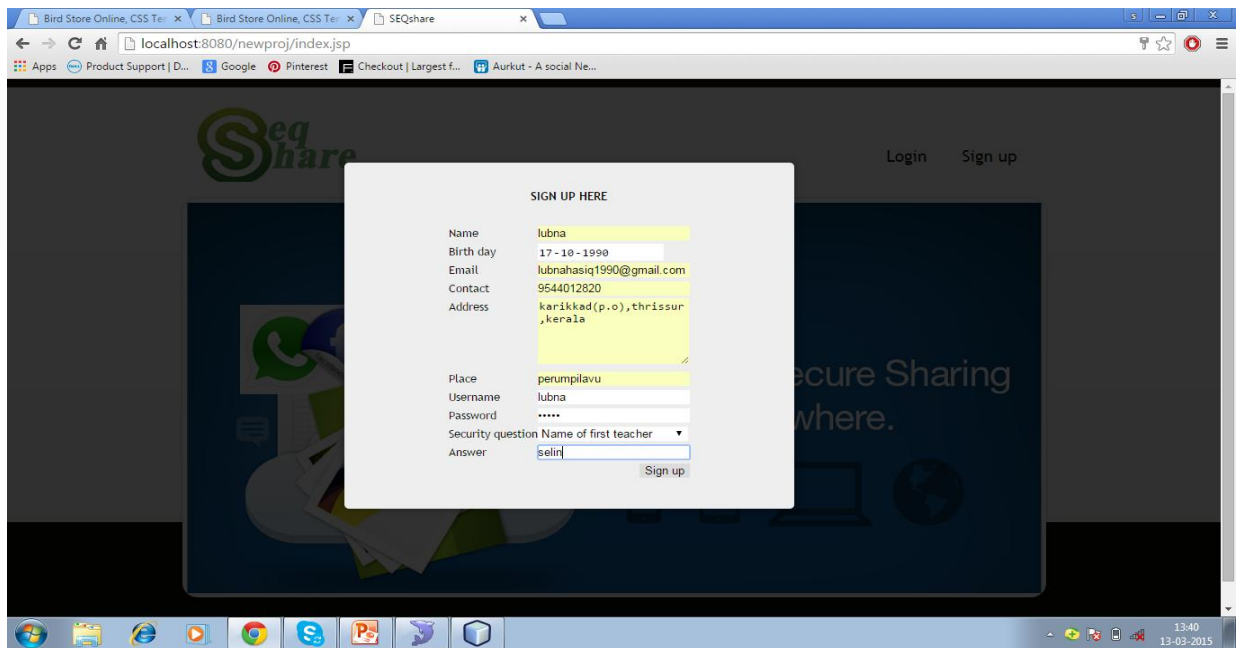
SCREEN SHOTS

8. Screen Shots

8.1. Login Page



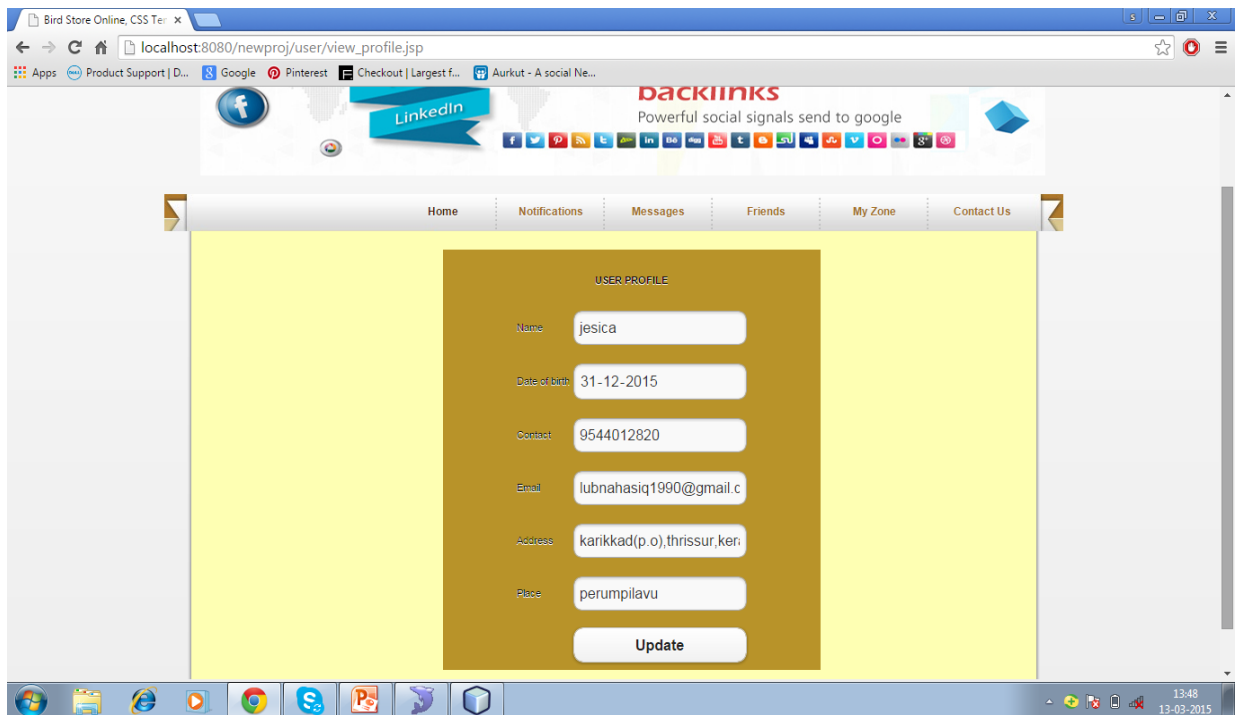
8.2. Registration Page



8.3. Friend Request



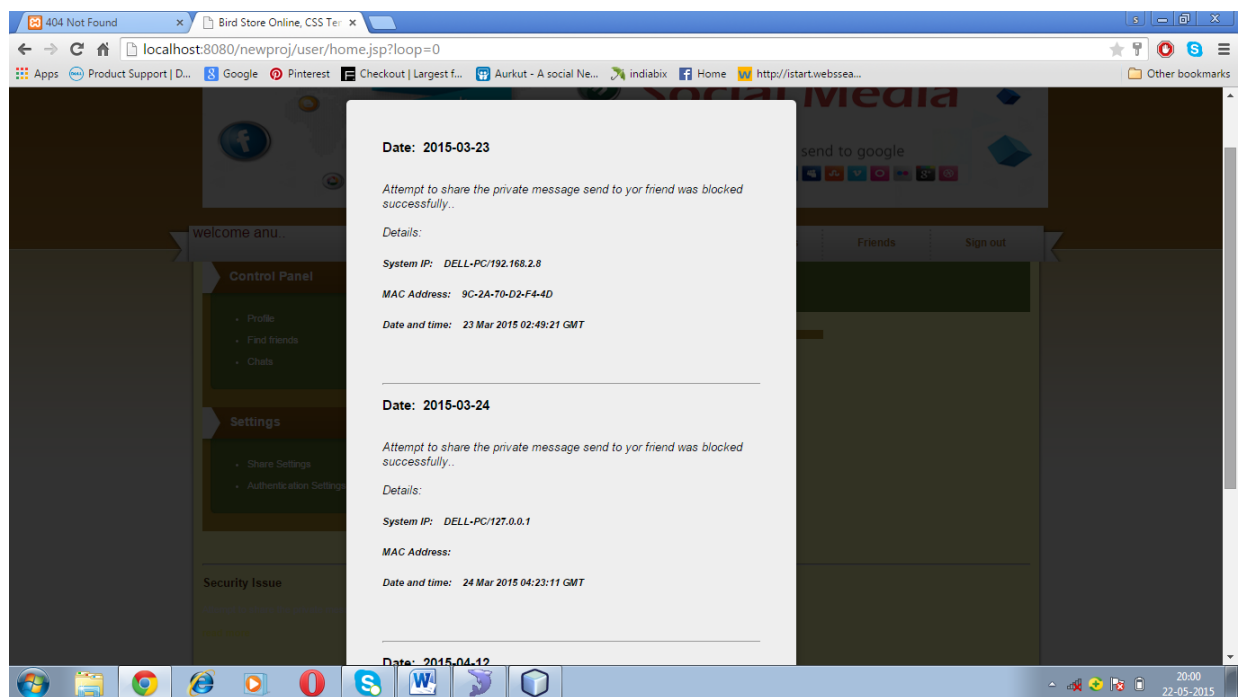
8.4. User Profile



8.5. Security Settings for Image



8.6. Security Notification



TESTING

9. TESTING

9.1. Test Case

TEST CASE					
TEST CASE	TEST DESCRIPTION	TEST PROCEDURE	TEST DATA	EXPECTED RESULT	ACTUAL RESULT
Login	To test whether Usemame and Password are valid	Enter the Usemame and Password	Admin Admindef	Valid user Invalid user	Valid user Valid user
Registration	Joining the social media site	Personal Data Entry	Data Entry Fields	Successful Registration	Successful Registration
Image Upload	Test whether the uploading is right or not	Upload Image	Image	Successfully Uploaded.	Successfully Uploaded
Security Settings	Check whether the uploaded image is in secure mode or not	Image Upload	Image	Security checking is Successful	Security checking is Successful
Friend Request	Send request for necessary friends	Search by names	Name	Request is Accepted/ Request is Rejected	Request is Accepted

9.2. Testing

The three testing strategies are:

- Unit Testing
- Integration Testing and
- System Testing

9.2.1. Unit Testing

In unit testing, tests each module individually. The first level of testing is unit testing. This test focuses on each module separately assuming that it functions properly as a unit. Unit testing focuses verification efforts on the smallest unit of software design, the module. This is also known as “Module Testing”. The modules are tested separately. This testing is carried out during programming stage itself. In this testing step each module is found to be working satisfactorily as regard to the expected output from the module. Developer and supervisor do the testing in order to trace out the bugs in the minimal part of the code. The goal is to test the integral logic of the module. Unit testing is typically done by the developers and not by end-users.

The unit testing comprises of two types of testing: White Box Testing and Black Box Testing.

9.2.1.1. Login module

9.2.1.1.1. Black Box Testing: In the login module the black box testing involves the testing of username and password. When we enter the username and password, and if it is valid, then we are logged in into the web site 'Seqshare'. Otherwise, if the username and password are wrong error message is displayed like 'Invalid Username or Password'.

9.2.1.1.2. White Box Testing: In this module the white box testing involves the testing of username and password. Here, in white box testing it includes the testing of the username and password that we enter with the username and password that we stored in the database. When we enter the username and password, it is compared with the username and password in the database. It is done for checking whether the username and password are valid. If the username and password are matching with the username and password in the database, the user with the entered username and password are allowed to logged in into the website 'Seqshare'.

9.2.1.2. Adding Friends Module

9.2.1.2.1. Black Box Testing: In the Adding Friends Module, the black box testing involves the searching and send friend request to the searched friend. If we want to add a friend to our friends list, then we can search them with name and can send friend request to them. When they accept our friend request the notification is provided to inform us that the request is accepted.

9.2.1.2.2. White Box Testing: In the Adding Friends Module, the White Box Testing is done by searching the friend by name. When we enter a name, it is searched in the database and retrieves the names related with the entered name. From the retrieved names, we can select the particular friend and can send friend request to the friend. When the request is send, the status is waiting, i.e., 0 and when the user accepts the request the status is changed to accept, i.e., 1. We can also do the white box testing by evaluating the status change of the database. This helps to identify whether the request is accepted or not.

9.2.1.3. Image Uploading

9.2.1.3.1. Black Box Testing: In the image uploading module, the black box testing is done by testing whether the image is uploaded to a particular friend or friends successfully or not.

9.2.1.3.2. White Box Testing: In the Image Uploading Module, white box testing is done to ensure the internal working of the image upload. Here, we check the database to evaluate whether the selected image for uploading is saved or stored correctly in the database. And the user id of the user to whom it is send, who send the image and date etc are stored in the database.

9.2.1.4. Image Sharing

9.2.1.4.1. Black Box Testing: In the image sharing module, the black box testing is done by testing whether the image is shared according to the security settings. It ensures that whether the image is shared is based on the security settings.

9.2.1.4.2. White Box Testing: In the image sharing module, the white box testing is done by testing the image to be shared that whether it follows the security settings. The security settings can be set by the user. The user can set the security of image by enable or disable share, save and download of the image. The security settings are stored in the database.

9.2.2. Integration Testing

Integration testing focuses on the design and the construction of the software architecture. The data can be lost across the interface or one module can pose an adverse effect on another. The sub functions when combined may not produce the major functions. Integration testing is a systematic technique for the program structure, while at the same conducting test to uncover errors associated with the interface. In integration testing here follows the top down approach in which the top modules are tested first and the basic modules are tested last. The modules are combined into one module and checks whether these modules are work together. Here a user must first sign up and then the user can log in to the site. Then the user can send friend request to another user, can share image, and can upload the images. The top down testing starts with login and tests the following modules work properly.

9.3.3. System Testing

System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. For user acceptance testing, the system was given to the end user to use. The errors found are rectified. In system testing, the whole system is tested. In system testing, we test the Graphical User Interface (GUI) of the website and ensure that the GUI is designed and worked properly. The flow of the entire system is tested to evaluate that the website work properly. Here also check the security and performance of the website. The system is tested to know whether it is susceptible for updation. The full website is executed to test the system runs correctly without any problem.

IMPLEMENTATION

10. IMPLEMENTATION

All the social media websites mainly aim at searching friends easily on entire network and sending friend request adding them as friends. Security of the images that we are sharing will be at a great risk, because when we are sharing an image, the image can be viewed, can be saved and also can be downloaded. When hackers or other type of unauthorized users got our username and password, they can easily share the image for unwanted users. The multimedia content, is being attacked, and used illegally by individuals, Users are not completely aware of who has access to their multimedia content, Also a user cannot prevent further sharing of the image, User cannot share a secure image with another user, Less data confidentiality are its disadvantages.

Purpose of Implementation

The main aim of the project is to overcome the disadvantages and provide a better system. Here we can share an image by setting security details. Here we can enable or disable share, save and download. When we disable the share feature and share the image, then even if the image is saved or downloaded to another location the image cannot be shared. And also a notification will be received by the sender about the unauthorized attempt of sharing image. Through this we can ensure the security of the images from unauthorized users.

CONCLUSION

11. CONCLUSION

While developing the system a conscious effort has been made to create and develop a software package, making use of available tools, techniques and resources – that would generate a proper system for Seqshare. While making the system, an eye has been kept on making it as user friendly. As such one may hope that the system will be acceptable to any user and will adequately meet his/her needs.

This site ‘Seqshare’ can be used as an extension to Facebook. Here there is also features like “Adding Friends”, sending them “Friend Requests”, ”Image Uploading” and “Sharing”. But here we can share an image by setting security details. Here we can enable or disable share, save and download. When we disable the share feature and share the image, then even if the image is saved or downloaded to another location the image cannot be shared. And also a notification will be received by the sender about the unauthorized attempt of sharing image. Through this we can ensure the security of the images from unauthorized users.

FUTURE ENHANCEMENT

12. FUTURE ENHANCEMENT

In the web site 'Seqshare', we can share an image by setting security details. Here we can enable or disable share, save and download. When we disable the share feature and share the image, then even if the image is saved or downloaded to another location the image cannot be shared. There are some of the areas of improvement which couldn't be implemented due to time constraints. One such feature was online chat where members can chat with his friends through this website.

BIBLIOGRAPHY

13. BIBLIOGRAPHY

APPENDIX I

14. APPENDIX I

14.1 GANTT CHART

