SECURE BROWSING USING ANTI-PHISHING

Submitted in partial fulfillment of the requirements
for the degree of

B. E. Computer Engineering

By

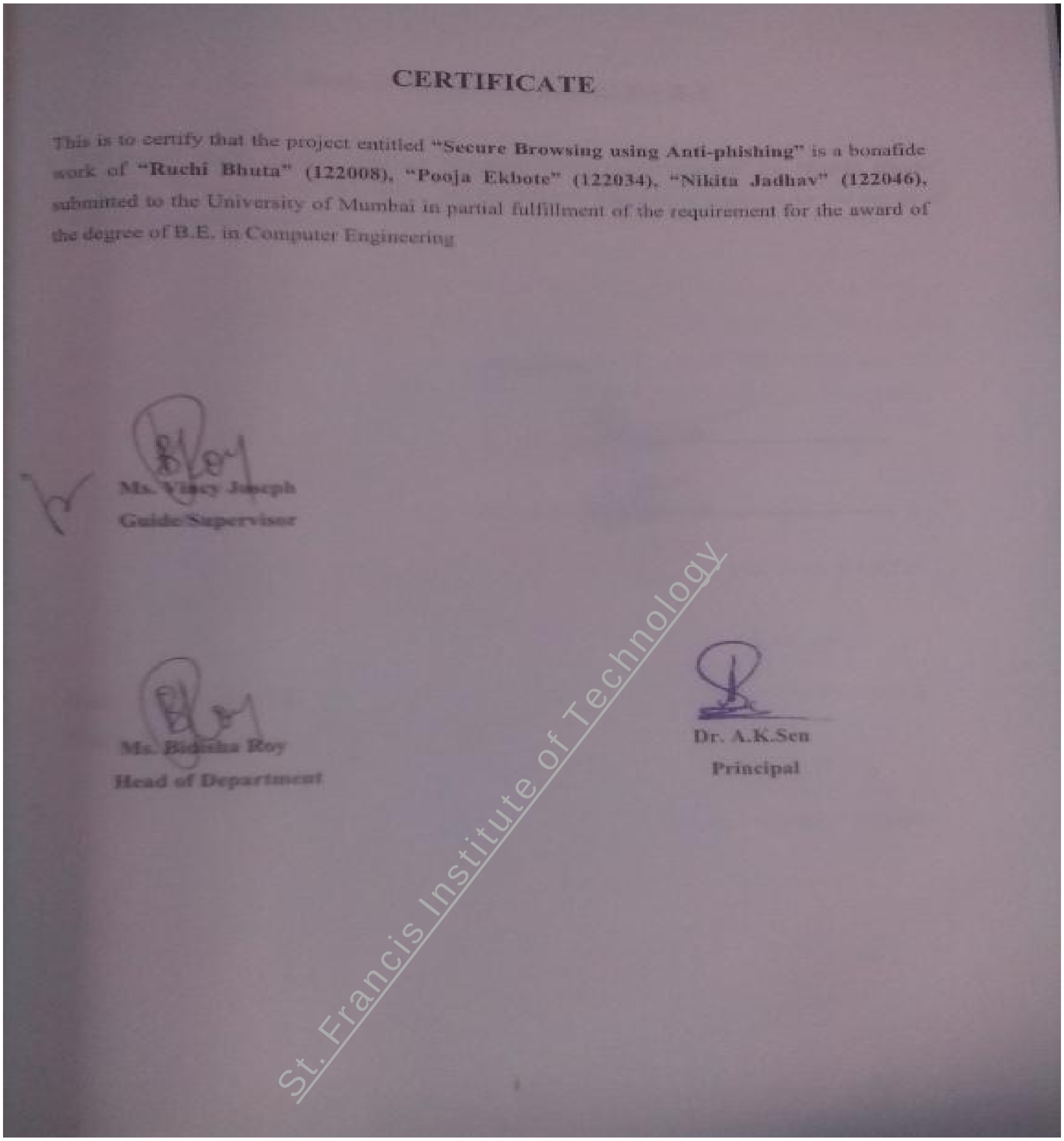
RUCHI BHUTA	**122008**
POOJA EKBOTE	**122034**
NIKITA JADHAV	**122046**

Supervisor:

Ms. Vincy Joseph

Assistant Professor

Department of Computer Engineering
St. Francis Institute of Technology
(Engineering College)
University of Mumbai
2015-2016



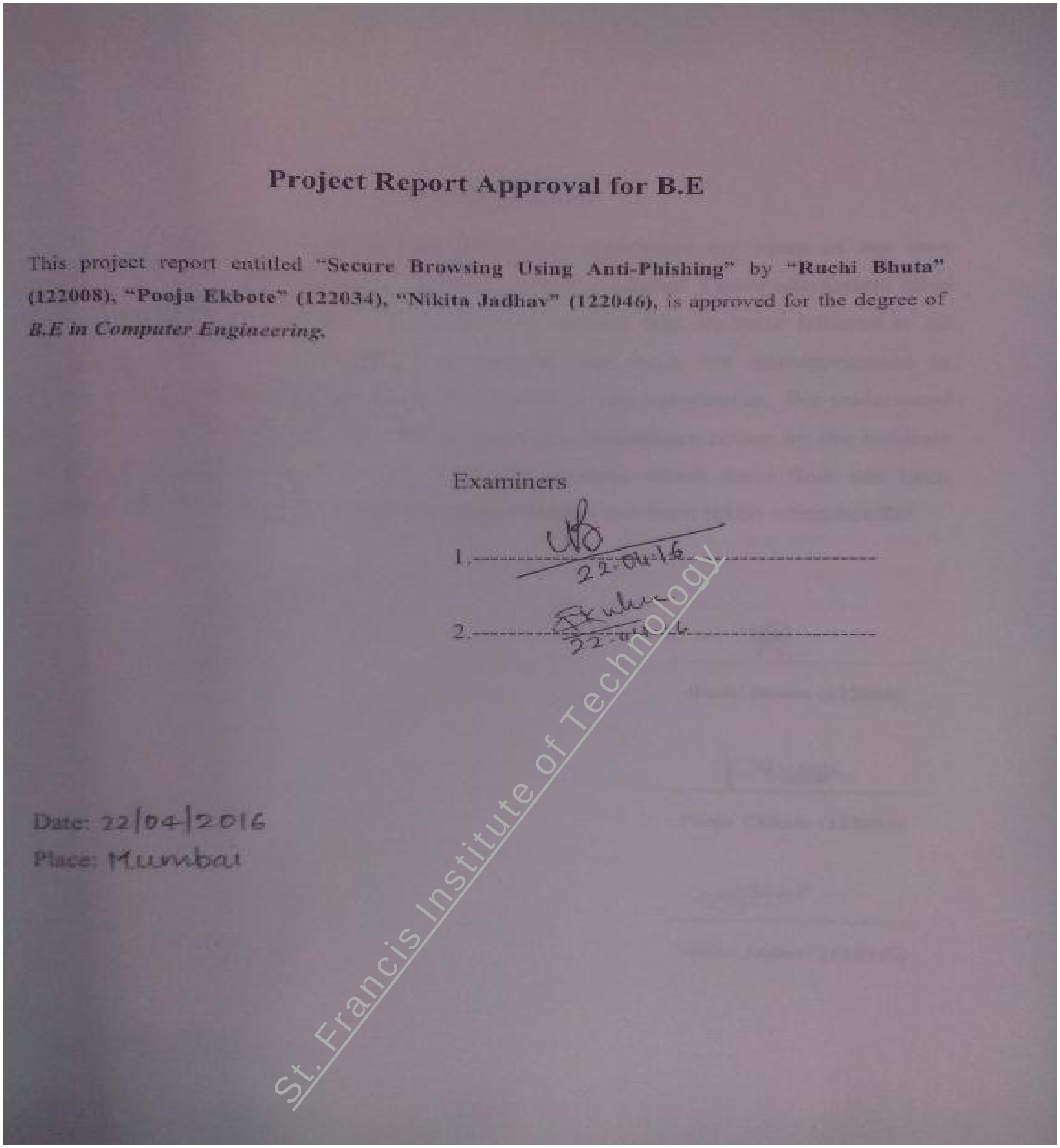
CERTIFICATE

This is to certify that the project entitled **"Secure Browsing using Anti-phishing"** is a bonafide work of **"Ruchi Bhuta" (122008), "Pooja Ekbote" (122034), "Nikita Jadhav" (122046),** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of B.E. in Computer Engineering.

Ms. Vincy Joseph
Guide/Supervisor

Ms. Bidisha Roy
Head of Department

Dr. A.K.Sen
Principal



Project Report Approval for B.E

This project report entitled "**Secure Browsing Using Anti-Phishing**" by "**Ruchi Bhuta**" **(122008)**, "**Pooja Ekbote**" **(122034)**, "**Nikita Jadhav**" **(122046)**, is approved for the degree of ***B.E in Computer Engineering.***

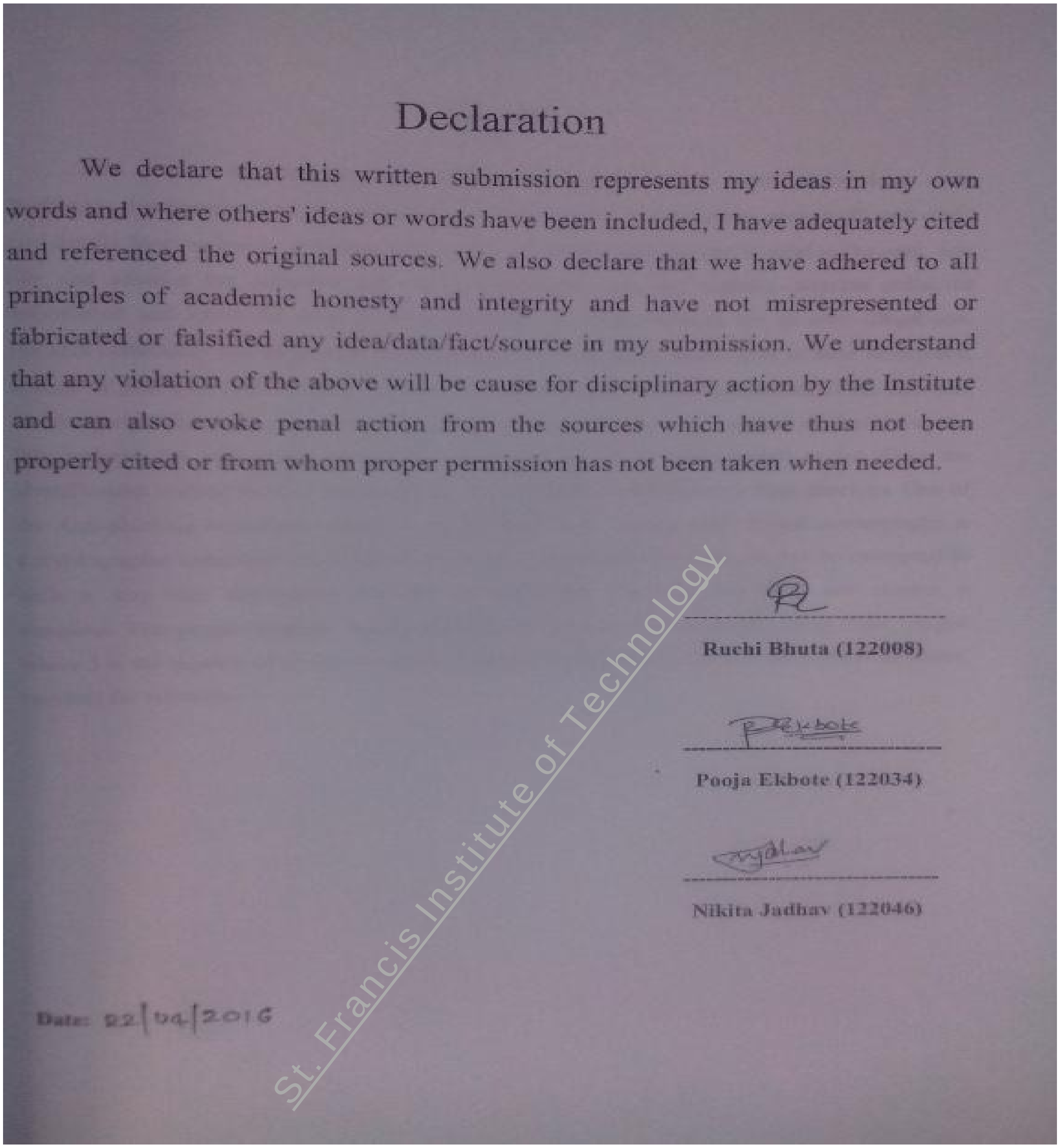
Examiners

1.-- 22-04-16

2.-- 22-04-16

Date: 22/04/2016

Place: Mumbai



Declaration

We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Ruchi Bhuta (122008)

Ekbote

Pooja Ekbote (122034)

Nikita Jadhav (122046)

Date: 22/04/2016

Abstract

Our project outlines the Anti-Phishing Technique on the social networking sites. Phishing is the attempt to acquire sensitive information such as usernames, passwords, and credit card details from websites. Attacker uses replica of original website that is send to the user, user fills and submits the sensitive and useful information into the website, attacker pulls the information and saves the data, credit card details etc from websites for its own illegal use. Hackers commonly use these sites to attack persons using these media sites in their workplace, homes, or public in order to take personal and security information that can affect the user and the company. Our main objective of this project is to make the user capable of identifying the fake website this will enable him to know whether the site is safe(original) or not. Once the identification is done the user can safely use the website with full security from attackers. One of the Anti-phishing techniques which we aim at is Visual Cryptography. Visual cryptography is a cryptographic technique which allows visual information (pictures, text, etc.) to be encrypted in such a way that decryption becomes a mechanical operation that does not require a computer. We are going to implement (2,3) visual cryptography using color images where 3 is the number of shares in which the image is divided and 2or 3 is the number of shares required for recovery.

Contents

List of Figures

List of Tables

List of Abbrevations

Sr. No	Abbreviations	Expanded form
1	VC	Visual Cryptography
2	ARGB	Alpha, Red, Green, Blue

Chapter 1
Introduction

Our project outlines the Anti-phishing Technique on the social networking sites. Phishing is the attempt to acquire sensitive information such as usernames, passwords. Phishing attacks mainly focus on websites, where attacker carry out fraudulent activities such as financial transactions on behalf of users by forging an email which contains an fake URL that redirects user to fake website masquerading as an online bank or a government entity.

Attacker uses replica of original website that is send to the user, user fills and submits the sensitive and useful information into the website. Hackers commonly use these sites to attack persons using these media sites in their workplace, homes, or public in order to take personal and security information that can affect the user and the company.

1.1 Description

Anti-phishing techniques are used to combat phishing. To make the user capable of identifying the fake website .Once the identification is done the user can safely use the website with full security from attackers.

One of the Anti-phishing techniques which we aim at is (2,3) colored Visual Cryptography. Visual cryptography is a cryptographic technique which allows encryption of visual information (pictures, text, etc.). Here 3 is the number of shares in which the image is divided and 2or 3 is the number of shares required for recovery of original image back. The individual shares revealed no information about the original image When 2 or 3 shares were overlaid, the original image would appear.

1.2 Problem Formulation

Phishing attacks mainly focus on websites, where attackers carry out fraudulent activities. Phished web pages have high visual similarities to the original web page. Attacker uses replica of original website that is send to the user, user fills and submits the sensitive information into the website. Victims of phishing web pages may expose their bank account, password, credit card number which can be misused by attacker.

Hence to provide high security against phishing attacks we are using (2, 3) colored Visual Cryptography as an anti-phishing technique.

1.3 Motivation

Visual Cryptography:-

Visual cryptography is a cryptographic technique which allows encryption of visual information (pictures, text, etc.). It is threshold scheme that takes a secret message and encrypts it in two different shares (2, 2) that reveal the secret image when they are overlaid. No additional information is required to create this kind of access structure.

Disadvantage:-

-This approach works only for black & white images.

-If any one of the shares is lost then a share generation algorithm is to be run to generate the share again which will take a lot of time.

-The performance measure is 72.55%.

1.4 Proposed Solution

Overcoming the drawbacks of Visual Cryptography (Basic) we are going to implement (2, 3) Visual Cryptography using color images.

Where 3 is the number of shares in which the image is divided and 2or 3 is the number of shares required for recovery. The individual shares revealed no information about the original image When 2 or 3 shares were overlaid, the original image would appear.

Using our proposed solution

- Faster to generate shares using randomized algorithm.
- Efficient reconstruction of image by working at pixel level.
- The problem of earlier algorithm of fixed size image is eliminated.
- Also saves time in case of a share lost as the generation of share is not required, reconstruction of image can be done using 2 shares.

1.5 Scope of the project

The scope of the system is in the security domain which is mainly related to online attacks. It provides a friendly environment to the user. General visual cryptography supports only black and white kind of images. Our system supports colored images.

Further our system helps the users in identifying the phished webpage on their own; this does not require any decryption technique. The reconstructed image is displayed on the user screen which helps him identify whether it is proper or not. This enables him to detect that the site is safe or not

Chapter 2
Review of Literature

Following are the reviews of the anti-phishing techniques taken into consideration:-

2.1 The Phishing Guide Understanding & Preventing Phishing Attacks:

The topic of phishing focusses on analyzing the tools and techniques which the professional criminals are being using currently. So analyzing the flaws in their techniques would aid in preventing many of the popular & successful phishing attacks. It is multi-tiered approach which comprises of client-side, server-side and enterprise.

Advantage:

Easy to implement.

Disadvantage:

It provides the limited scope to identify strategy.

The performance measure is 66.45%. [2]

2.2 CAPTCHA : Using Hard AI Problems For Security:

A captcha is a program that can generate and grade tests that most humans can pass, but current computer programs can't pass.

Advantage:

Introduces a new class of hard AI problems that can be exploited for security purposes.

Disadvantage:

Not all people have same intelligence to solve the problem, imply win-win situation: either the problems are solved or remain unsolved.

The performance measure is 85.30% [2]

2.3 A Text-Graphics Character captcha for Password Authentication:

The Text-Graphics Character (TGC) captcha is introduced for preventing dictionary attacks against password authenticated systems, in which attackers repeatedly attempt to gain access using the entries in a list of frequently used passwords.

Advantage:

The system improves the security of servers, allowing remote terminal access, also involves pattern recognition, computer graphics, and psychology.

Disadvantage:

But it is slow, large database are required and retrieving time is large.

The performance measure is 89.54% [2]

2.4 Hashed Based Visual Cryptography Scheme For Image Authentication

This method can be used to produce the smart cards to the users, which can be used to authenticate the users for providing the services. The authentication becomes very fast. It can be applied in biometric fingerprint scanning where the thumb image is captured and shares are generated by using the visual cryptography system.

Advantage:

This approach reduces the size of database & also the process of retrieving and comparing speeds up when compared to retrieving the images from database.

Disadvantage:

It needs separate hardware for authentication.

The performance measure is 90.15%. [2]

2.5 Visual Cryptography (Basic)

This is a simplest threshold scheme that takes a secret message and encrypts it in two different shares that reveal the secret image when they are overlaid. No additional information is required to create this kind of access structure. This approach works only for black & white images.

Advantage:

When both shares are present only that time the secrete image will revealed otherwise not.

Disadvantage:

The main flaw of this method is that if any one of the share is lost then a share generation algorithm is to be run to generate the share again which will take a lot of time. The size of image was restricted.

The performance measure is 72.55%. [2]

Chapter 3
System Analysis

3.1Functional Requirements

According to the functions of the system it is divided into three modules:-

Input Requirements:-

- User Details
- Image
- Shares

Processing Requirements:-

- It is core software which uses the image from the input module
- Generation of Shares
- Stacking of Shares

Output Requirements:-

- Stacked Image

3.2 Non-Functional Requirements

Performance Requirements

- Generation and stacking of shares will not take more time it should be done in seconds
- The system supports multiple user log in at the same time without affecting its consistency
- The consistency and security of database system must be maintained effectively without affecting its performance.

Safety and Security Requirements

- The main safety requirement is the security of the database servers.
- There should be a backup provided for the system for cases such as power failure, or some kind of hardware failure.

3.3 Specific Requirements

- **Hardware Requirements:-**
  - Windows OS
  - Pentium Processor
  - Minimum 512 MB RAM
  - 10 GB Hard Disk

- **Software Requirements:-**
  - Netbeans IDE 7.3.1 and above
  - Java Jdk version 1.7 and above
  - Xampp server
  - MySql database
  - JDBC Driver
  - Web Browser

3.4 Usecase Diagram for (2, 3) Visual Cryptography

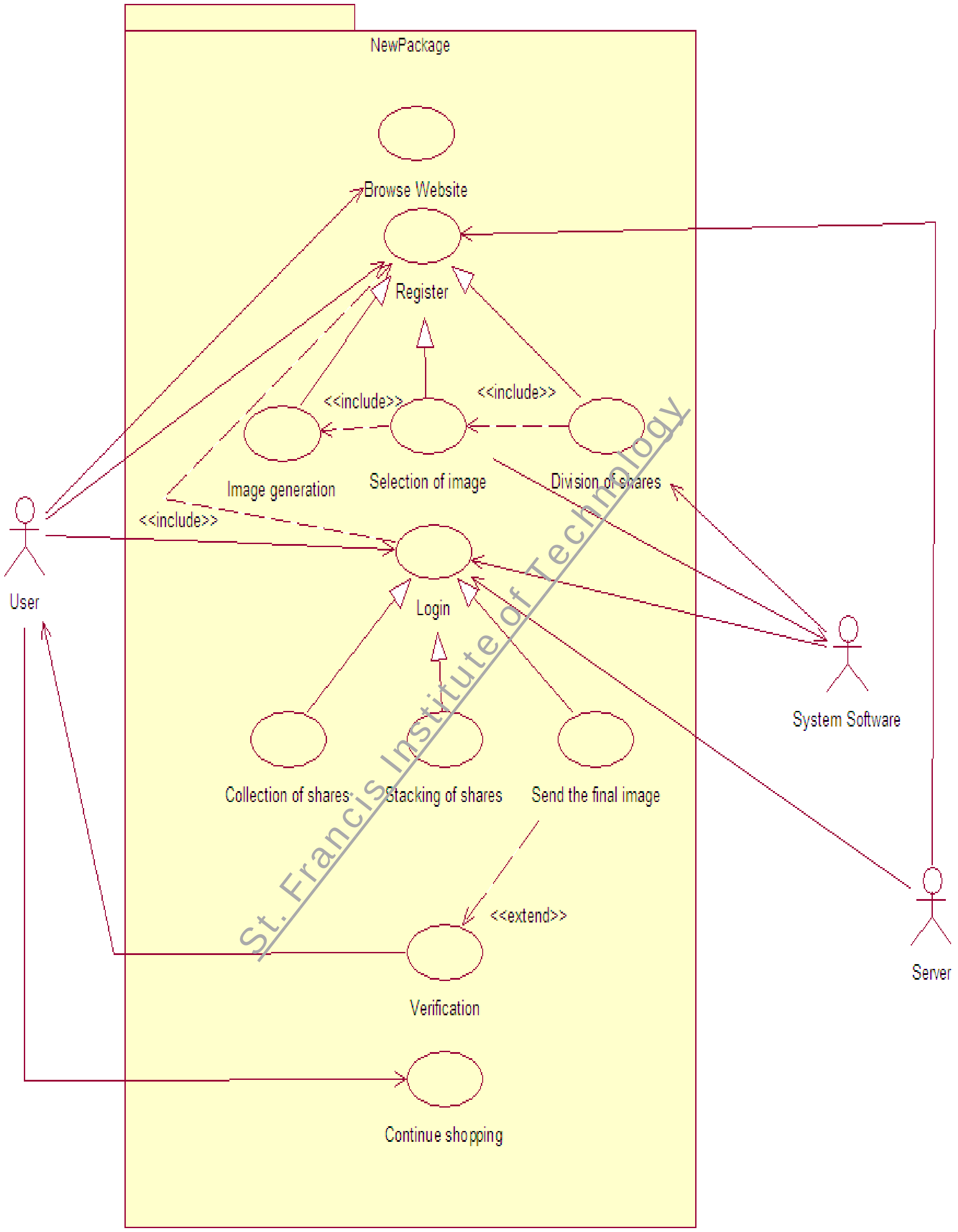


Figure 3.4.1. Usecase Diagram for (2, 3) Visual Cryptography

Table 3.4.1: Usecase Browse Website

Use Case ID	1
Use Case Name	Browse Website
Actors	User
Description	User browses the Shopping website and searches for the products he/she wishes to buy.
Pre-conditions	User should have an working internet connection.
Flow of Events	1. User searches for the shopping site 2. Browses for the different items to buy.
Post-conditions	-

Table 3.4.2: Usecase Register

Use Case ID	2
Use Case Name	Register
Actors	User
Description	User registers to the shopping website and enters his registration details.
Pre-conditions	User should enter all his details.
Flow of Events	3. User now registers onto the site. 4. Enters the details like username ,password, Date of birth and Gender etc.
Post-conditions	-

Table 3.4.3: Usecase Image Generation

Use Case ID	3
Use Case Name	Image Generation
Actors	System Software
Description	Randomly images are generated by the system software.
Pre-conditions	-
Flow of Events	5. Further in registration random images are displayed
Post-conditions	-

Table 3.4.4: Usecase Selection Of Image

Use Case ID	4
Use Case Name	Selection Of Image
Actors	User
Description	User selects one of the images generated from the system software.
Pre-conditions	Image should be generated.
Flow of Events	6. User selects one of them.
Post-conditions	-

Table 3.4.5: Usecase Division of Shares

Use Case ID	5
Use Case Name	Division Of Shares
Actors	System Software, User & Server
Description	Implement (2,3) VC algorithm and generate 3 shares which is then divided among the user and server.
Pre-conditions	One particular Image should be selected for its further division into shares.
Flow of Events	7. Image is divided into shares 8. These shares are given to the user and the server.
Post-conditions	-

Table3.4.6: Usecase Login

Use Case ID	6
Use Case Name	Login
Actors	User
Description	User logs in to the website and provides its share which was selected at the time of registration.
Pre-conditions	User should be registered.
Flow of Events	9. At the time of login user enters its username, password and its share.
Post-conditions	-

Table 3.4.7: Usecase Collection of Shares

Use Case ID	7
Use Case Name	Collection Of Shares
Actors	System Software
Description	Shares are collected at the time of login from the user as well as the server.
Pre-conditions	-
Flow of Events	10. Shares from the server and the user are collected.
Post-conditions	-

Table 3.4.8: Usecase Selection Of Image

Use Case ID	8
Use Case Name	Stacking Of Shares
Actors	Server
Description	Shares are stacked and stored onto the server.
Pre-conditions	-
Flow of Events	11. These shares are then stacked together at the server.
Post-conditions	-

Table 3.4.9: Usecase Send the Final Image

Use Case ID	9
Use Case Name	Send the Final Image
Actors	Server, System Software
Description	Final Image is sent to the user by the server.
Pre-conditions	-
Flow of Events	12. After stacking final image is sent to the user.
Post-conditions	Final Image verified with the original one.

Table 3.4.10: Usecase Verification

Use Case ID	10
Use Case Name	Verification
Actors	User
Description	Verification of the final image with the original image is done.
Pre-conditions	-
Flow of Events	13. Verification with the original one is performed.
Post-conditions	-

Table 3.4.11: Usecase Continue Shopping

Use Case ID	11
Use Case Name	Continue Shopping
Actors	User
Description	If the final image matches with the original one then user continues shopping.
Pre-conditions	-
Flow of Events	14. If the verification holds true then user continues shopping.
Post-conditions	-

Chapter 4

Analysis Modeling

4.1 DFD Level 0 for (2, 3) Visual Cryptography

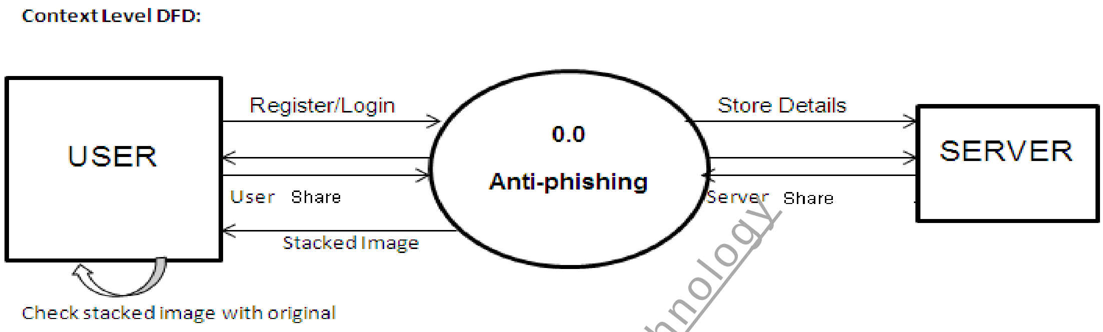


Figure 4.1.1 DFD level 0 for (2, 3) Visual Cryptography

DFD Level 1 for (2, 3) Visual Cryptography

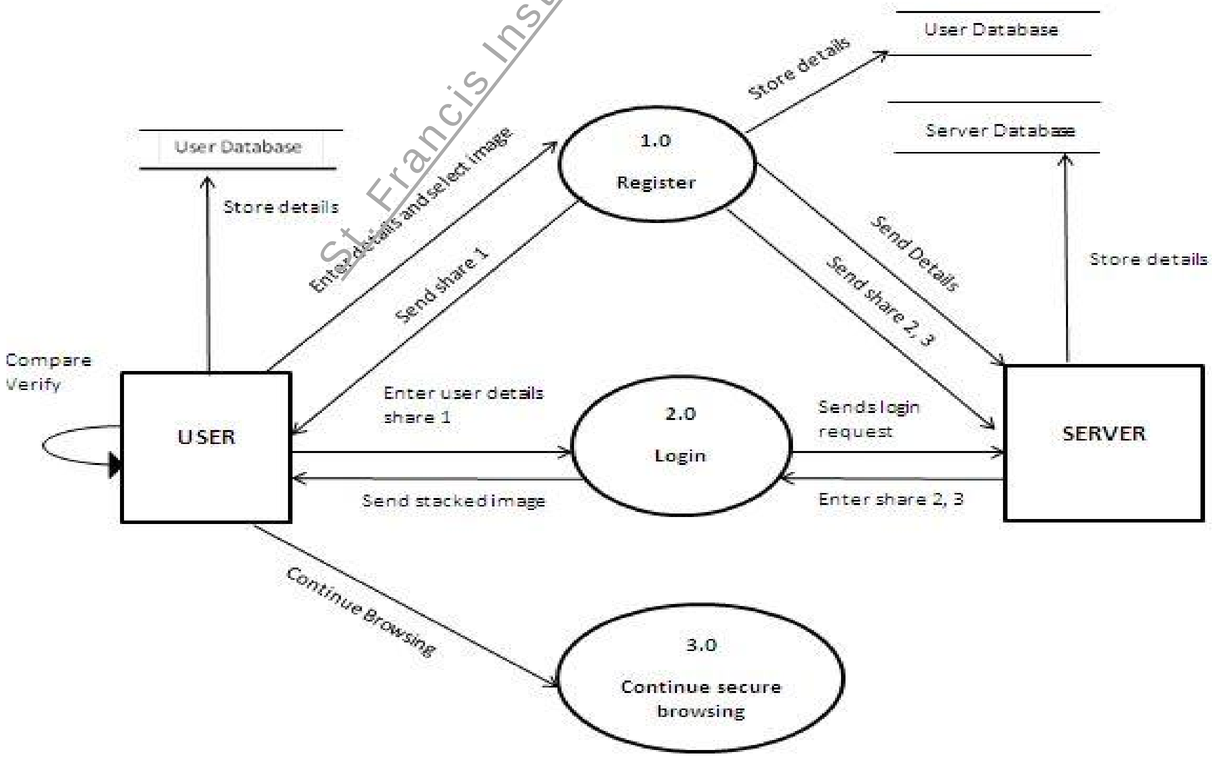


Figure 4.1.2 DFD Level 1for (2, 3) Visual Cryptography

DFD Level 2 for (2, 3) Visual Cryptography

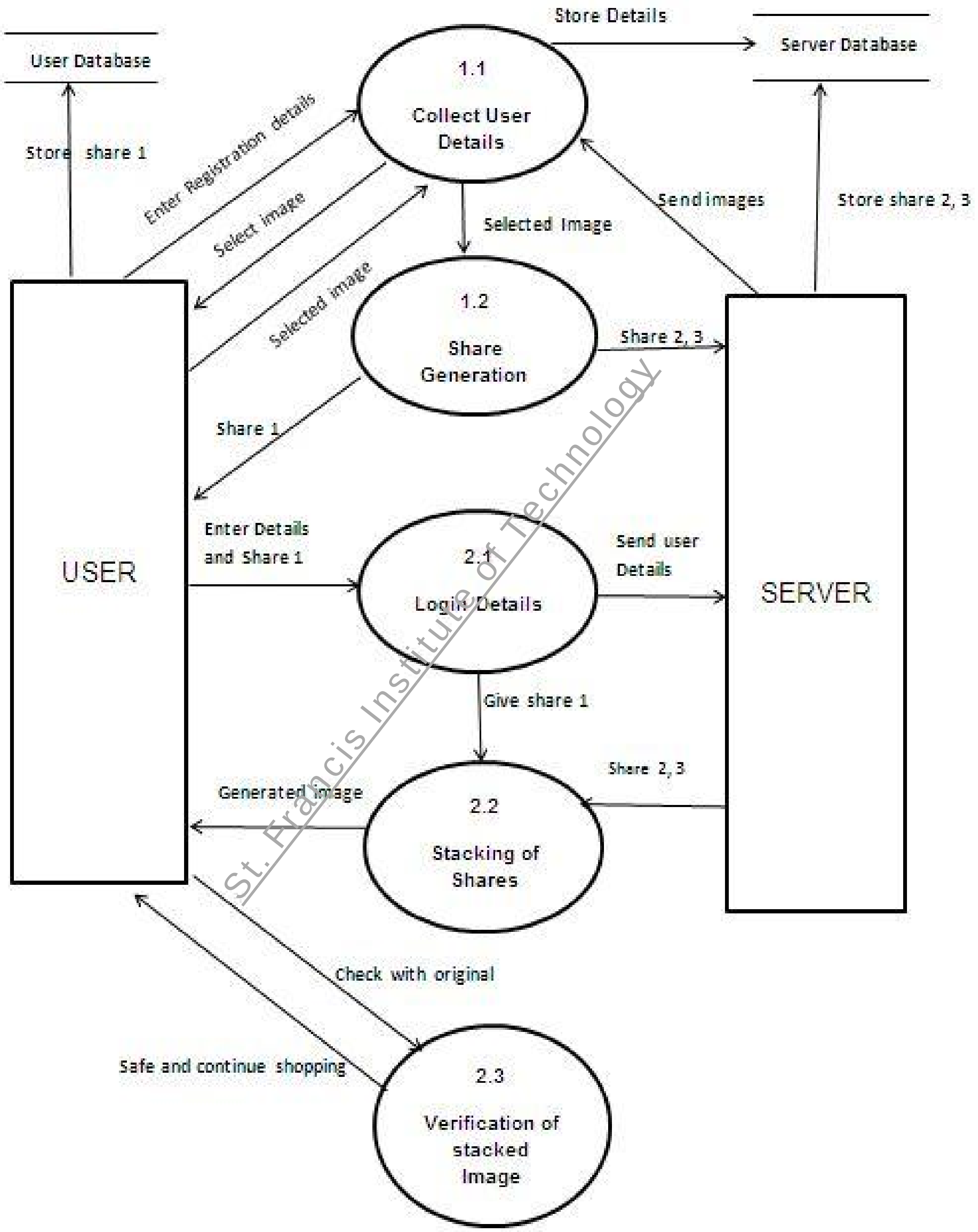


Figure 4.1.3 DFD Level 2 for (2, 3) Visual Cryptography

Data Flow Diagram (DFD) provides a visual representation of the flow of information (i.e. data) within a system. By drawing a Data Flow Diagram, you can tell the information provided by and delivered to someone who takes part in system processes, the information needed in order to complete the processes and the information needed to be stored and accessed.

DFD Level 0:-

DFD level 0 is the entrance of a data flow model. It contains one and only one process and does not show any data store. Here the two entities are user and the server. The one main process is anti-phishing process. In this, user registers himself, the share generation takes place using (2,3) colored VC algorithm and distributed to user (share 2) and server (share 1,3).Further user can login himself by entering his share. The shares are collected from user and server and stacked together using (2, 3) colored VC algorithm. Stacked image is them displayed on user screen. User verifies the image and decides to continue shopping or not. Refer figure [4.1.1]

DFD Level 1

At this level, the main process anti-Phishing is divided into 3 processes:-

1.0 Registration process,

2.0 login process

3.0 Continue secure shopping. Refer figure [4.1.2]

DFD Level 2

At this level the, the registration process further divided into following:-

1.1 **Collect user details**: Here the user details are collected and are stored into database. Further user selects image and that image is also stored.

1.2 **Share generation**: The shares are generated using (2, 3) color VC algorithm and they are distributed to user (share 2) and server(share1,3). These shares are stored at respective databases.

Further the login process divided as:-

2.1 **Login details**: Here details are collected from the user, and the user share is also collected.

2.2 **Stacking of shares:** The shares from user and server are stacked together to reconstruct the image. The generated Image is send to user. Refer figure [4.1.3]

Now, the last process where user verifies the image, if the reconstructed image matches with the original image then user identifies that the site is not phished and he continues browsing

4.2 Activity diagram for (2, 3) Visual Cryptography

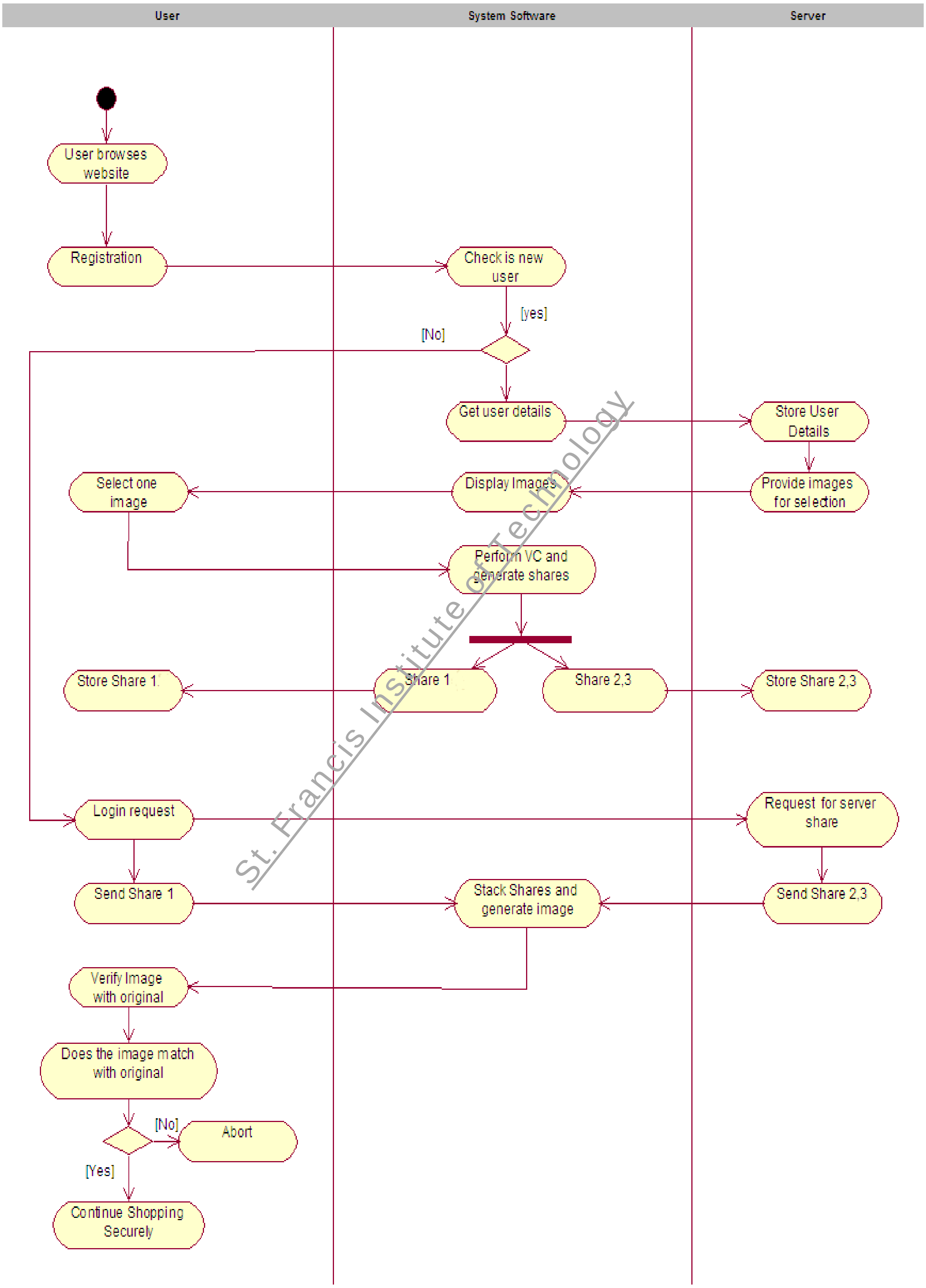


Figure 4.2.1 Activity Diagram for (2, 3) Visual Cryptography

Activity diagram is an UML diagram to describe dynamic aspects of the system. Activity diagram is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another which can be sequential, branched or concurrent.

In given Figure, the Activity starts from user actor. The first activity performed by user is browsing website. User goes for registration phase.

A diamond is the standard symbol for specifying condition.

A swim lane (or swim lane diagram) is a visual element used in process flow diagrams, or flowchart that visually distinguishes job sharing and responsibilities for sub-processes of a business process. Swim lanes may be arranged either horizontally or vertically.

Registration phase: If the user is new, then system software performs (2, 3) color VC algorithm to generate shares and these shares are distributed to the user (share 2) and server (share 1, 3).These shares are stored at respective databases.

Login phase: If user is old i.e. He is already registered user then he directly goes to login phase. System software collets shares from the user and server, stack them using (2, 3) color VC algorithm and reconstructed image displayed to user. User verifies the reconstructed image, and decides to continue shopping or not. Refer figure [4.2.1]

4.3 Sequence diagram for (2, 3) Visual Cryptography

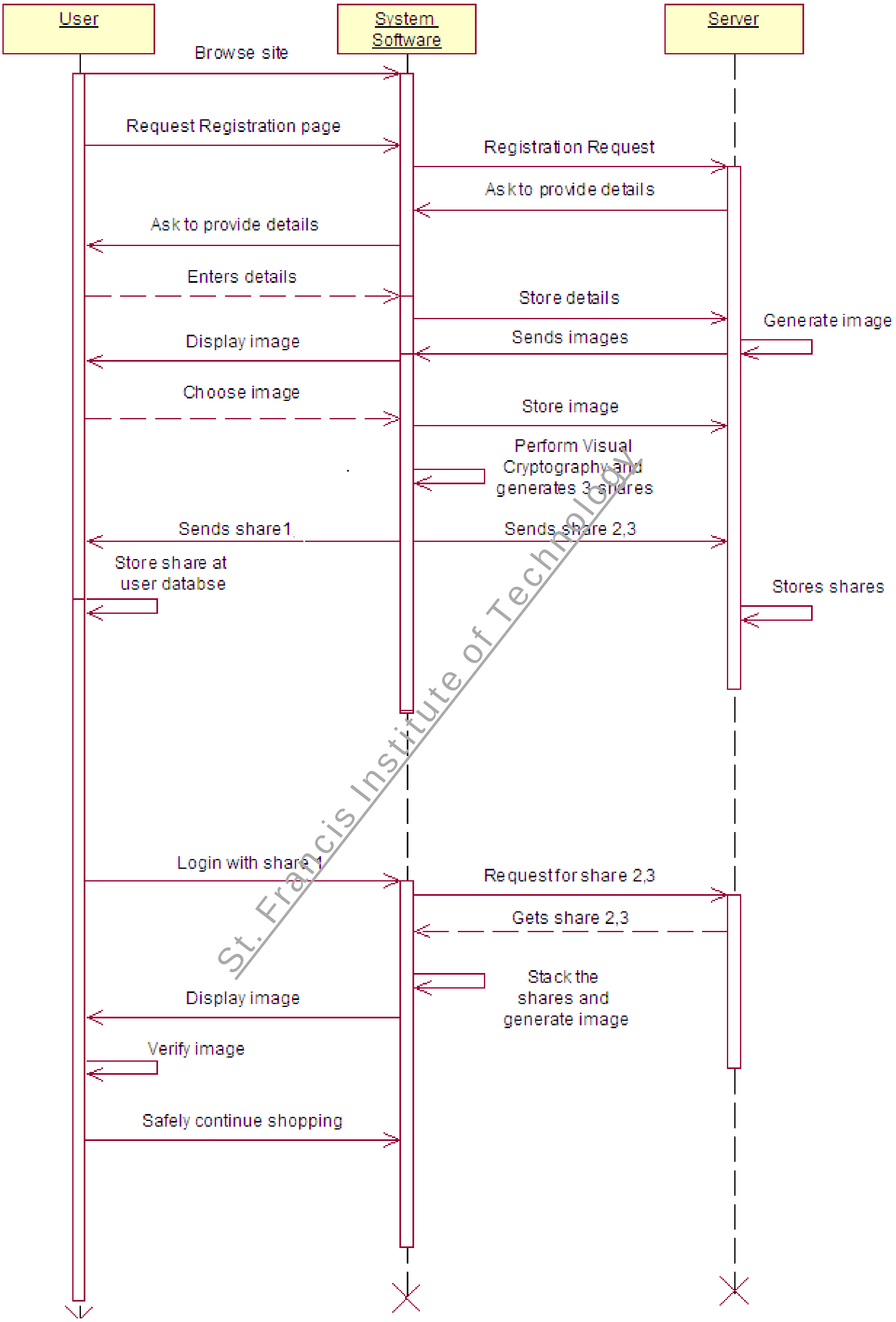


Figure 4.3.1 Sequence Diagram for (2, 3) Visual Cryptography

The Sequence Diagram models the collaboration of objects based on a time sequence. It shows how the objects interact with others in a particular scenario of a use case. With the advanced visual modeling capability, you can create complex sequence diagram in few clicks. Besides, Visual Paradigm can generate sequence diagram from the flow of events which you have defined in the use case description.

Actor: An Actor models a type of role played by an entity that interacts with the subject (e.g., by exchanging signals and data), but which is external to the subject (i.e., in the sense that an instance of an actor is not a part of the instance of its corresponding subject). Actors may represent roles played by human users, external hardware, or other subjects

Call Message: A message defines a particular communication between Lifelines of an Interaction. Call message is a kind of message that represents an invocation of operation of target lifeline.

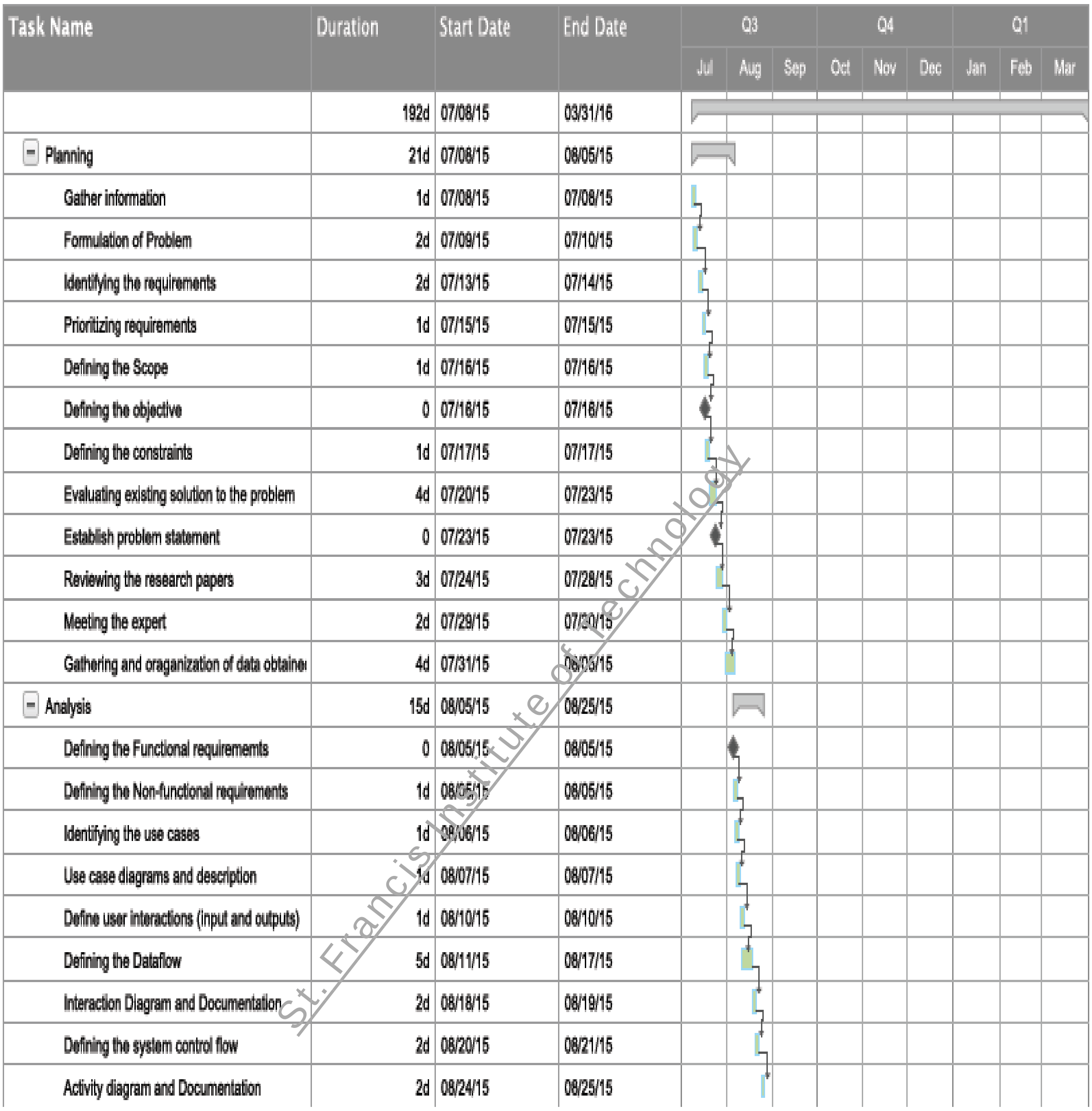
Lifeline: A lifeline represents an individual participant in the Interaction.

Recursive Message: Recursive message is a kind of message that represents the invocation of message of the same lifeline. It is target points to an activation on top of the activation where the message was invoked from.

Registration phase: New user requests for the registration page and the server asks the user to provide its details and then details are stored on the server and further images are generated which highlights the concept of recursive message , then system software performs (2,3) color VC algorithm to generate shares and these shares are distributed to the user and server. These shares can then be found at their respective databases.

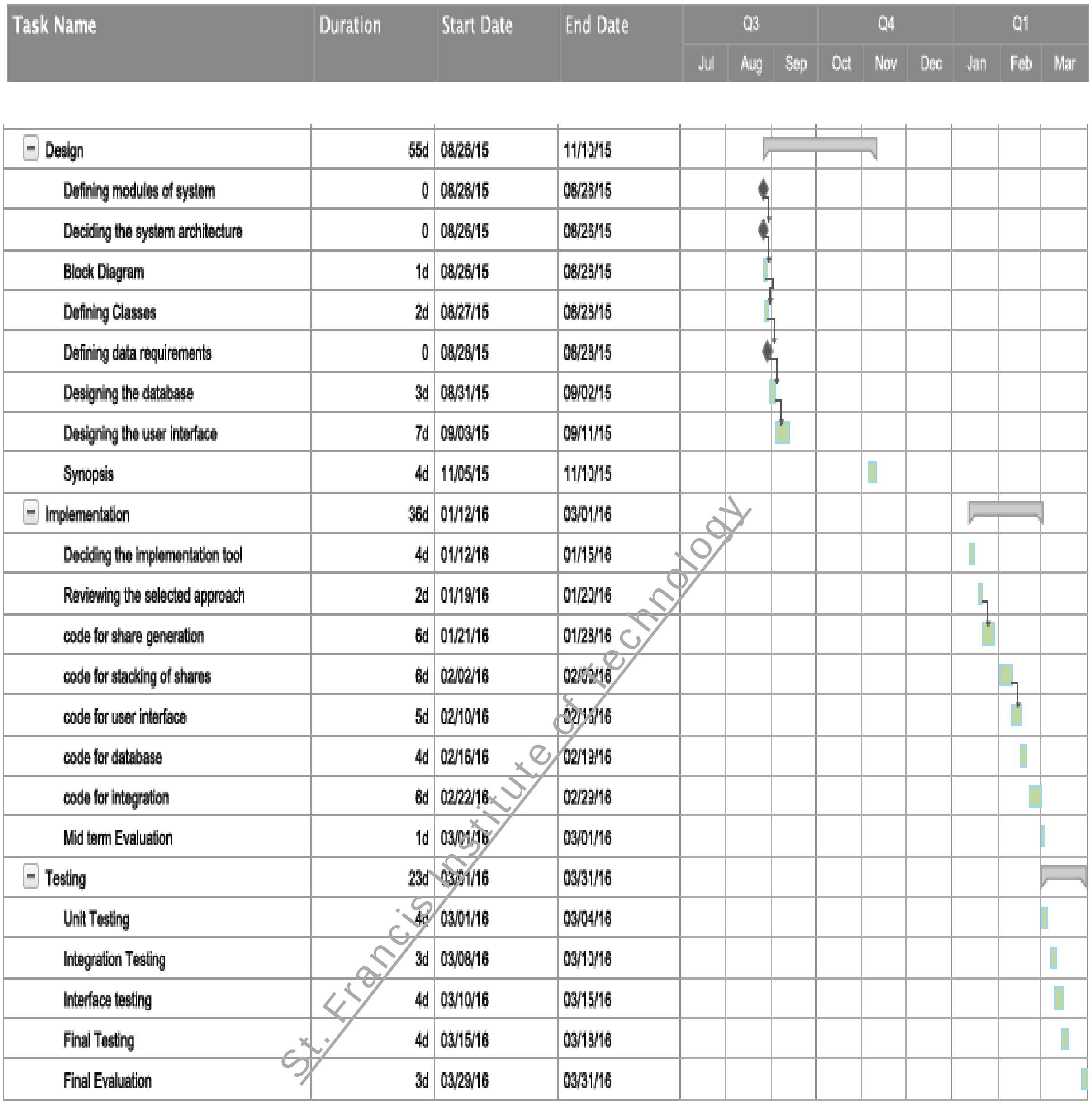
Login phase: If user is already registered then he logs in with his share. System software collets shares from the user and server, stacks them using (2, 3) color VC algorithm and reconstructed image displayed to user. User verifies the reconstructed image, and decides to continue shopping or not. Refer figure [4.3.1]

4.4 Timeline chart for (2, 3) Visual Cryptography



Task Name	Duration	Start Date	End Date
	192d	07/08/15	03/31/16
⊟ Planning	21d	07/08/15	08/05/15
Gather information	1d	07/08/15	07/08/15
Formulation of Problem	2d	07/09/15	07/10/15
Identifying the requirements	2d	07/13/15	07/14/15
Prioritizing requirements	1d	07/15/15	07/15/15
Defining the Scope	1d	07/16/15	07/16/15
Defining the objective	0	07/16/15	07/16/15
Defining the constraints	1d	07/17/15	07/17/15
Evaluating existing solution to the problem	4d	07/20/15	07/23/15
Establish problem statement	0	07/23/15	07/23/15
Reviewing the research papers	3d	07/24/15	07/28/15
Meeting the expert	2d	07/29/15	07/30/15
Gathering and oraganization of data obtaine	4d	07/31/15	08/05/15
⊟ Analysis	15d	08/05/15	08/25/15
Defining the Functional requirememts	0	08/05/15	08/05/15
Defining the Non-functional requirements	1d	08/05/15	08/05/15
Identifying the use cases	1d	08/06/15	08/06/15
Use case diagrams and description	1d	08/07/15	08/07/15
Define user interactions (input and outputs)	1d	08/10/15	08/10/15
Defining the Dataflow	5d	08/11/15	08/17/15
Interaction Diagram and Documentation	2d	08/18/15	08/19/15
Defining the system control flow	2d	08/20/15	08/21/15
Activity diagram and Documentation	2d	08/24/15	08/25/15

Figure 4.4.1Timeline Chart-I for (2, 3) Visual Cryptography



Task Name	Duration	Start Date	End Date
⊟ Design	55d	08/26/15	11/10/15
Defining modules of system	0	08/26/15	08/26/15
Deciding the system architecture	0	08/26/15	08/26/15
Block Diagram	1d	08/26/15	08/26/15
Defining Classes	2d	08/27/15	08/28/15
Defining data requirements	0	08/28/15	08/28/15
Designing the database	3d	08/31/15	09/02/15
Designing the user interface	7d	09/03/15	09/11/15
Synopsis	4d	11/05/15	11/10/15
⊟ Implementation	36d	01/12/16	03/01/16
Deciding the implementation tool	4d	01/12/16	01/15/16
Reviewing the selected approach	2d	01/19/16	01/20/16
code for share generation	6d	01/21/16	01/28/16
code for stacking of shares	6d	02/02/16	02/09/16
code for user interface	5d	02/10/16	02/16/16
code for database	4d	02/16/16	02/19/16
code for integration	6d	02/22/16	02/29/16
Mid term Evaluation	1d	03/01/16	03/01/16
⊟ Testing	23d	03/01/16	03/31/16
Unit Testing	4d	03/01/16	03/04/16
Integration Testing	3d	03/08/16	03/10/16
Interface testing	4d	03/10/16	03/15/16
Final Testing	4d	03/15/16	03/18/16
Final Evaluation	3d	03/29/16	03/31/16

Figure 4.4.2 Timeline Chart-II for (2, 3) Visual Cryptography

For our project we are following waterfall model as System development life cycle model. Systems Development Life Cycle (SDLC) is a process used by a systems analyst to develop an information system, including requirements, validation and user (stakeholder) ownership. Any SDLC should result in a high quality system that meets system objective customer, expectations, reaches completion within time and cost estimates, and works effectively and efficiently. The waterfall model consists of five phases:-

- Planning
- Analysis
- Design
- Implementation
- Testing and Deployment

Planning: This is the first phase where we gather information about what is phishing, what problems are arising because of that. What different techniques available to combat phishing. Further we formulate the problem. We defined the project scope and objectives of the project and met our project guide. Further we did research on previous papers, articles, tried to identified the advantage and disadvantages with the earlier approaches used for anti-phishing. We gathered the data, information and organized it and too review from the guide.

Analysis: In this phase, we identified what can be the functional and nonfunctional requirements of the project. We figured out the input and outputs for the system and sub process involved in it. Then we started working on defining the user interaction with the system in terms of inputs, outputs. Further we worked on usecase diagram, dataflow diagram, activity diagram, sequence diagram and also their documentation.

Design: Design phase involves defining of the major modules of the system like registration and login phases. Here we also designed the overall system architecture, the block diagrams of the two major modules of the system that is Registration phase and Login phase. Then we worked on the designing of the user interface which involves designing of online shopping website with general Home page, Registration page, Login page.

Implementation: In this phase we have performed the generation and stacking of shares.

Testing: In this phase we have tested the cases wherein the 3 shares are properly generated during the registration phase also the stacking of shares is performed using 2 and 3 share.

Refer figure [4.4.1] and figure [4.4.2]

Chapter 5
Design

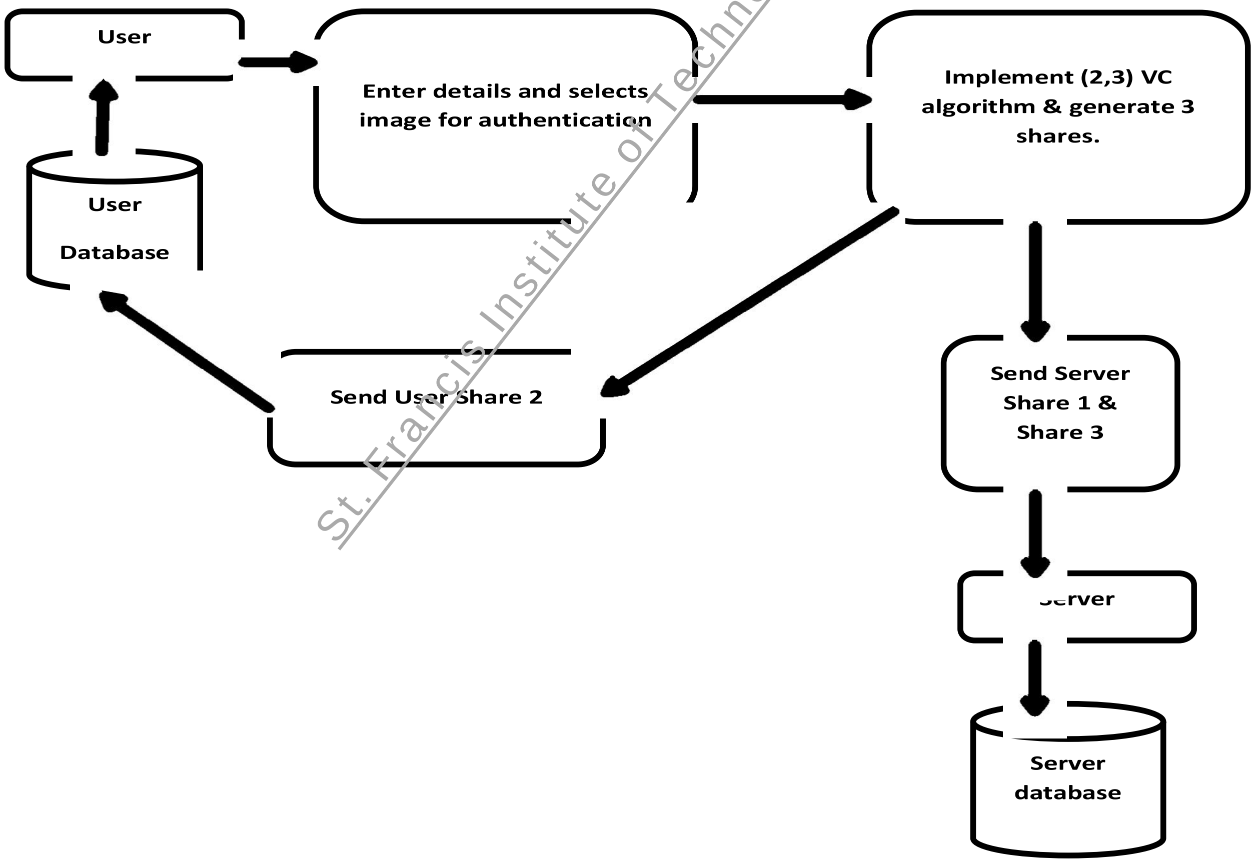
5.1Architectural Design

The proposed approach can be divided into two phases:

5.1.1 Registration phase

5.1.2 Login phase

Registration phase:



User
Enter details and selects image for authentication
Implement (2,3) VC algorithm & generate 3 shares.
User Database
Send User Share 2
Send Server Share 1 & Share 3
Server
Server database

Figure 5.1.1 Block Diagram-Registration Phase for (2, 3) Visual Cryptography

Registration phase:-

The user requests for the registration phase. The user enters the details and these details are stored at the server database. The images are then sent to user, and then one of the images is selected by user. The selected image is stored at database.

The image is divided into 3 numbers of shares. The share generation takes place with help of randomized algorithm. In this the bits in the image are marked as filled bit and empty bit. These bits are placed at alternate positions for generation shares.

The shares are then distributed to user (share2), and server (share 1, 3). The shares are stored at respective databases. Refer Figure [5.1.1]

Login phase:

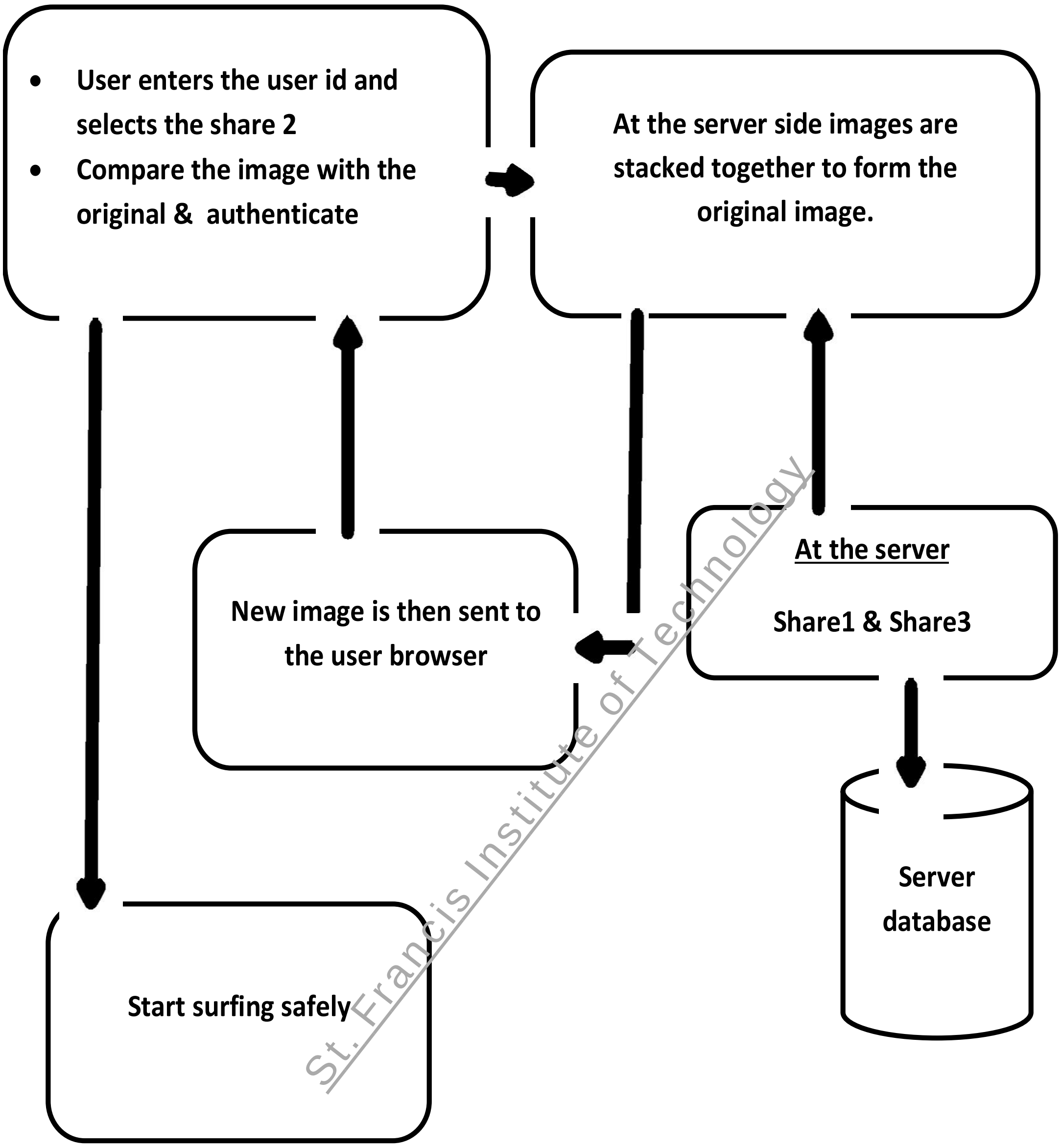

User enters the user id and selects the share 2
Compare the image with the original & authenticate
At the server side images are stacked together to form the original image.
New image is then sent to the user browser
At the server
Share1 & Share3
Server database
Start surfing safely

Figure 5.1.2 Block Diagram-Login Phase for (2, 3) Visual Cryptography

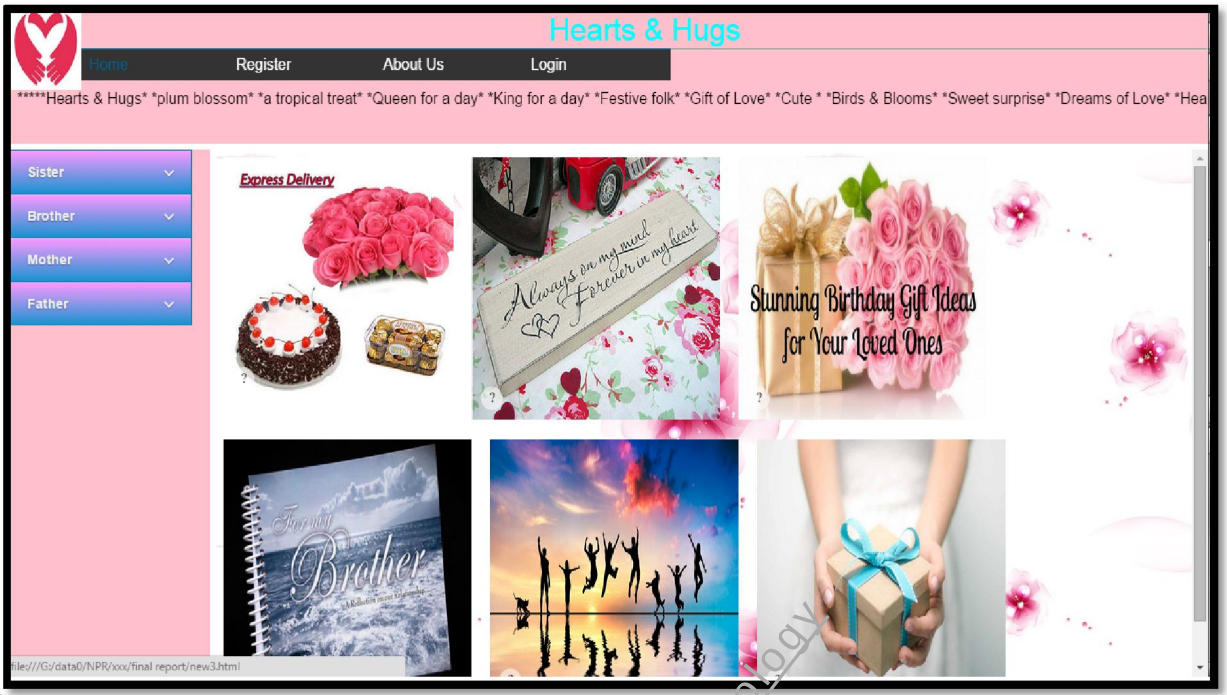
Login Phase:-

The login phase consists of the collection of user details. The user enters his/ her details in the system.

The shares are collected from the server (share 1, 3) and user (share 2).These shares are stacked together using (2, 3) Colored Visual Cryptography. And the reconstructed image is displayed on user screen.

User verifies the image, if the reconstructed image is similar to that of original image then user identifies the server is not faked one and he/she continues shopping safely. If the reconstructed image does not match with the original image then user identifies that the server is faked and thus detects the phishing successfully. Refer figure [5.1.2]

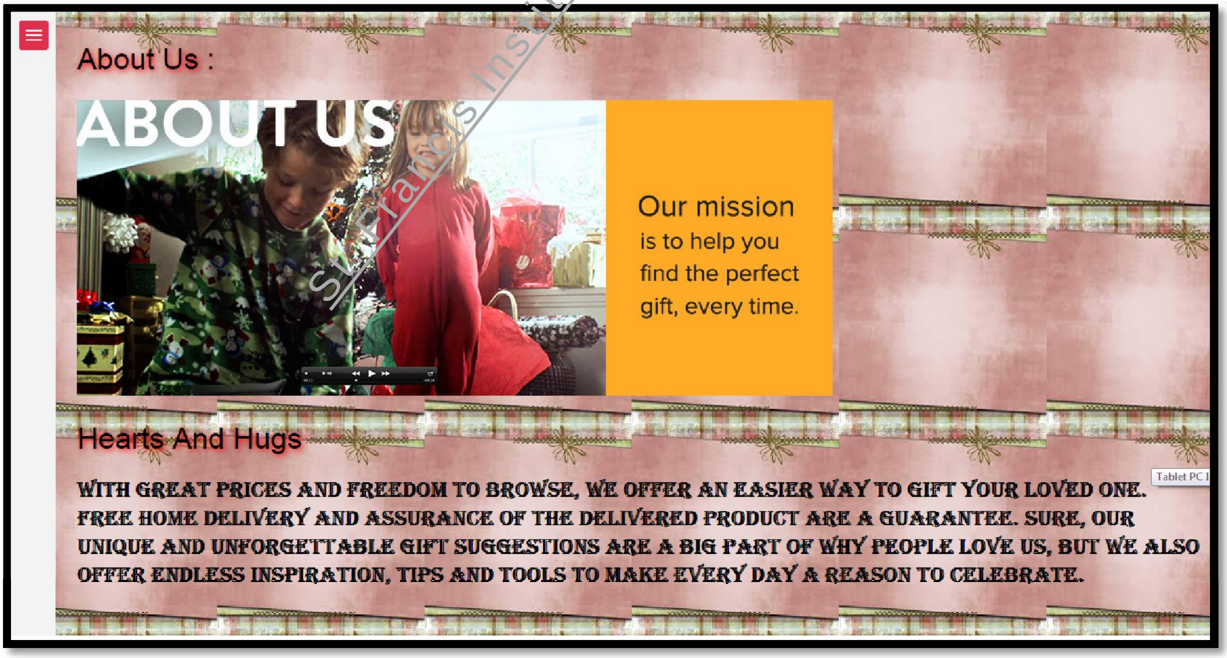
5.2 User Interface Design


Hearts & Hugs
Home
Register
About Us
Login
Sister
Brother
Mother
Father
Express Delivery
Always on my mind Forever in my heart
Stunning Birthday Gift Ideas for Your Loved Ones
Forever Brother
file:///G:/data0/NPR/xxx/final report/new3.html

*****Hearts & Hugs* *plum blossom* *a tropical treat* *Queen for a day* *King for a day* *Festive folk* *Gift of Love* *Cute * *Birds & Blooms* *Sweet surprise* *Dreams of Love* *Hea

Figure 5.2.1 UI Snapshot- Home Page

It is used for displaying the items available on the site for general view open to all. It displays different kinds of items availability for different user types



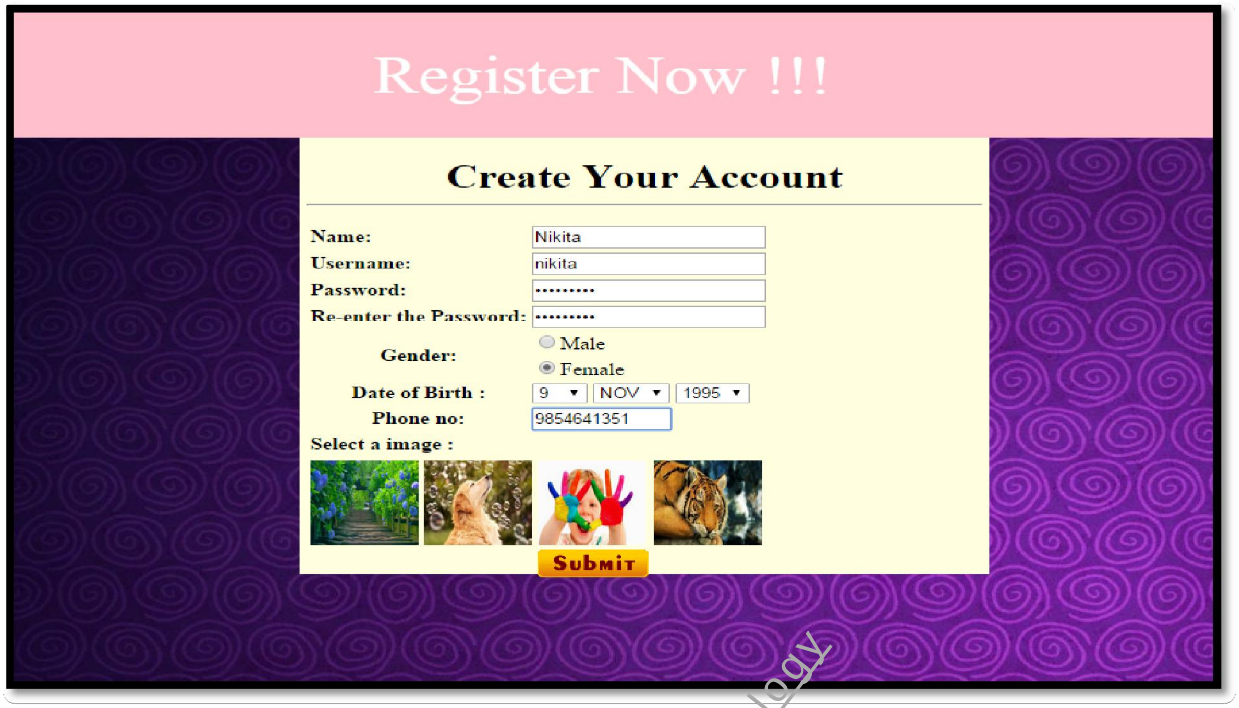
Our mission is to help you find the perfect gift, every time.

Hearts And Hugs

WITH GREAT PRICES AND FREEDOM TO BROWSE, WE OFFER AN EASIER WAY TO GIFT YOUR LOVED ONE. FREE HOME DELIVERY AND ASSURANCE OF THE DELIVERED PRODUCT ARE A GUARANTEE. SURE, OUR UNIQUE AND UNFORGETTABLE GIFT SUGGESTIONS ARE A BIG PART OF WHY PEOPLE LOVE US, BUT WE ALSO OFFER ENDLESS INSPIRATION, TIPS AND TOOLS TO MAKE EVERY DAY A REASON TO CELEBRATE.

Figure 5.2.2 UI Snapshot-About Us

This GUI is used to display the details about the website basically it tells about the mission of this website



Register Now !!!

Create Your Account

Name: Nikita

Username: nikita

Password: •••••••••

Re-enter the Password: •••••••••

Gender: ○ Male ● Female

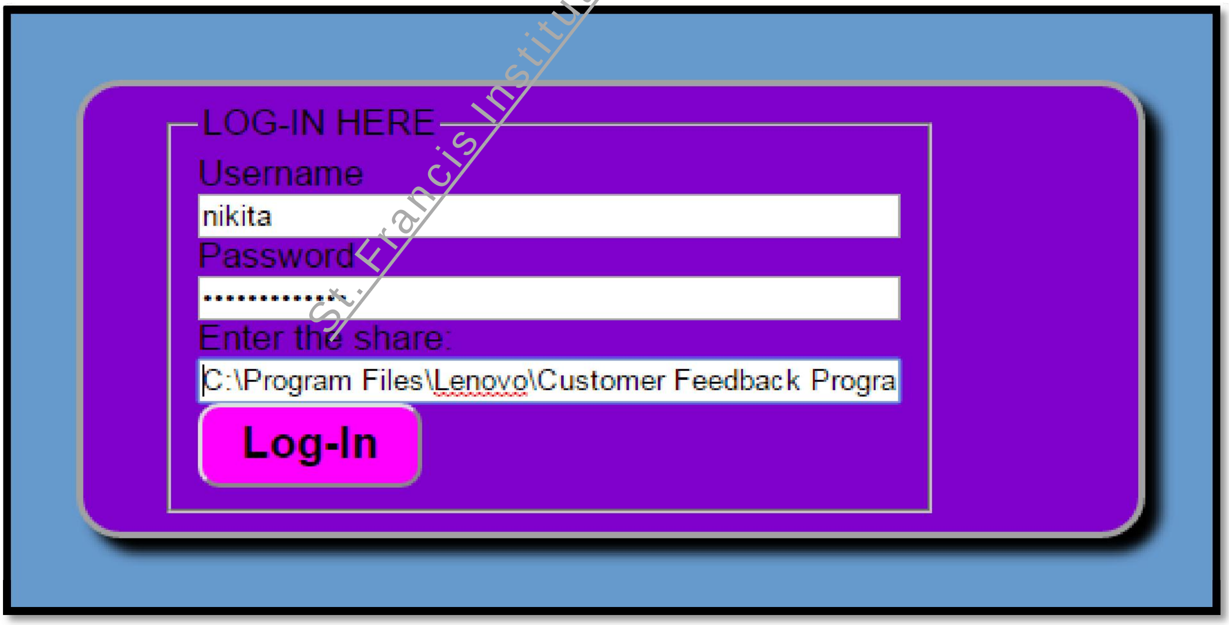
Date of Birth : 9 NOV 1995

Phone no: 9854641351

Select a image :

Figure 5.2.3 UI Snapshot- Registration Page

The user registers using this webpage where he enters the required details which when submitted is stored in the database and user gets its share



LOG-IN HERE

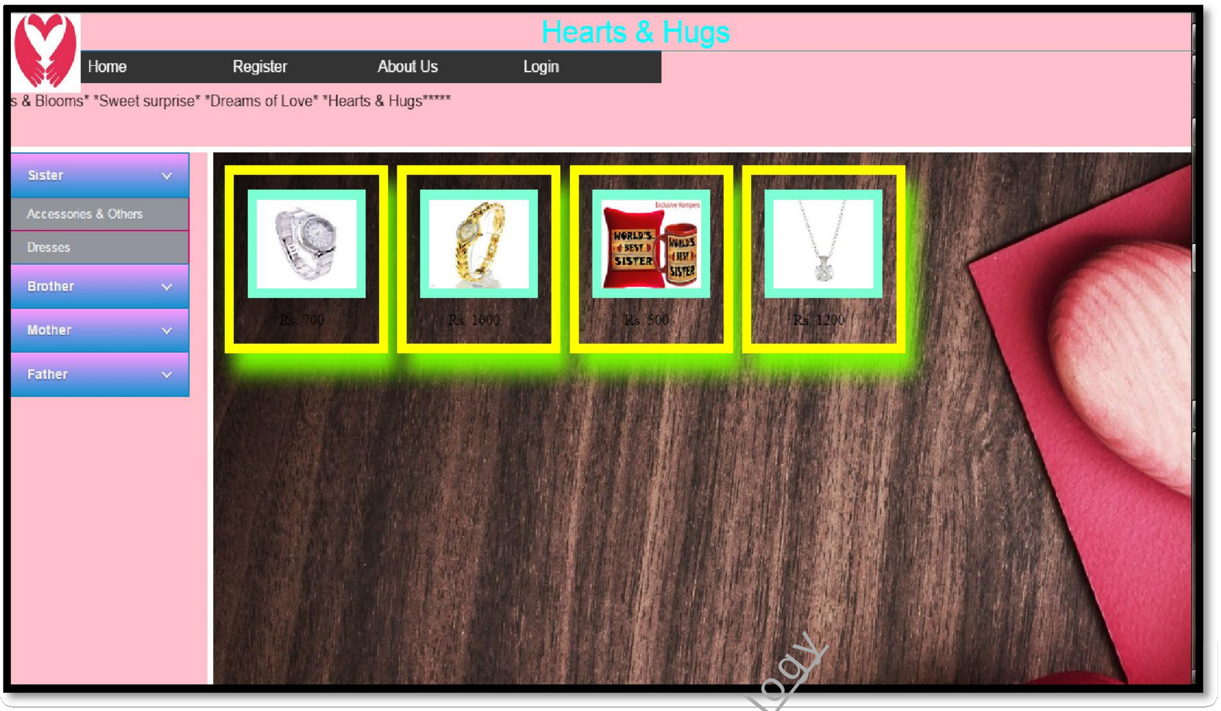
Username: nikita

Password: ••••••••••••••

Enter the share: C:\Program Files\Lenovo\Customer Feedback Progra

Figure 5.2.4 UI Snapshot- Login Page

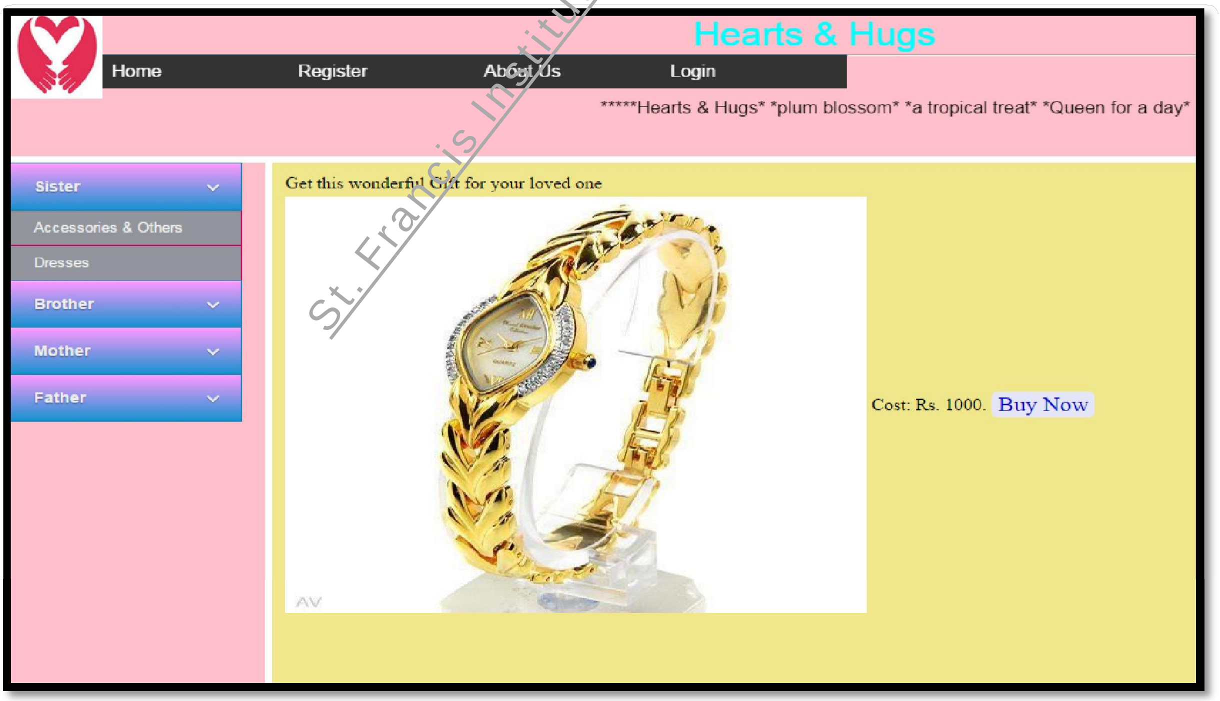
The user log in using this webpage where he enters the required details which when submitted is verified and on successful login the generated image is displayed on user screen.



s & Blooms* *Sweet surprise* *Dreams of Love* *Hearts & Hugs*****

Figure 5.2.5 UI Snapshot- Select Items

After successful login the user can now be able to select items of various categories and continue shopping



*****Hearts & Hugs* *plum blossom* *a tropical treat* *Queen for a day*

Get this wonderful Gift for your loved one

Cost: Rs. 1000.

Figure 5.2.6 UI Snapshot- Buy Items

After selecting items the user can now to proceed to the payment section.

Chapter 6
Implementation

6.1 Algorithms

6.1.1 Registration Phase: Share Generation

1. Take Input from user: User details ,an image and scan each pixel of image
2. Convert the image to ARGB
3. The number of shares the image would be divided is 3 and number of shares to reconstruct the image (2 or 3).

   Step: I: Take an image as input and calculate its width (w) and height (h).

   Step II: Scan each pixel of image and convert to 32 bit binary string let PIX

   Step III: Call Random Function for plotting the pixels for shares

   Step IV: Create 1D array to store constructed pixels of each share

   Step V: Construct three shares from the array
4. Shares are distributed to the user and the server, where Share 1 and common share is given to user and Share 2 and common is given to server.
5. End.

6.1.2 Login Phase: Stacking of Shares

1. Input from user it's details and share
2. Input Server share and common share
3. Scan pixels of each share and store it in an array
4. Combine the arrays to get a final array which contains a combination of pixels of each share using OR operation.
5. Generate an image of this array by plotting the pixels.
6. This generated image is then given to the user
7. User verifies the image with the original image
   - if (generated image==original)
     - ➢ Secure website continue shopping
   - Else
     - ➢ Not safe site abort.
8. End

Login Phase:

Reconstruction using 3 shares:-

Case 1:

- User Logs in using Share2(Mandatory)
- Server gives Share1 and Share3
- All the three shares are stacked together using (2,3) colored visual cryptography algorithm
- Stacked Image is displayed on user screen

Reconstruction using 2 shares:-

Case 2: (If share1 or share3 is lost)

- User Logs in using Share2(Mandatory)
- Server gives Share1 or Share3
- The two shares are stacked together using (2,3) colored visual cryptography algorithm
- Stacked Image is displayed on user screen

Case 3: (If user share is lost)

- User requests share
- Server authenticates user and sends share3
- The two shares are stacked together using (2,3) colored visual cryptography algorithm
- Stacked Image is displayed on user screen

Case 4: If site is phished

- User enters its share2
- Random share is given by fake site.
- Stacked image displayed on user screen.

6.2 Working of the Project

Share Generation

```
import com.mysql.jdbc.Driver;
import java.awt.image.BufferedImage;
import java.awt.image.WritableRaster;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.lang.Integer.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.*;
import java.util.Random;
import java.util.Scanner;
import javax.imageio.ImageIO;

public class Beproj{
 public static int[][][] img_share;        // to store the pixels of 3 number of shares
 public static int po=0,ki1=0,ki2=0,ki3=0;
 public static int ja=0;
 public static BufferedImage img1,img2,img3;
 public static int [] pixels1=new int[500000];
 public static int [] pixels2=new int[500000];
 public static int [] pixels3=new int[500000];
  public static void main(String[] args) throws Exception, IOException, SQLException  {
  String url = "jdbc:mysql://localhost:3306/sharegen";
        String user = "root";
        String password ="";
try {
File f,f1,f2,f3;
int n=3,k=2,rec=2,m,s=0,i,e,j,h,c,P;  //declare n=no of shares, r=for reconstruction
int k1,k2,k4 =0,k3, height,width;
String b="",val;
int[] rand=new int[3];
FileInputStream imguser=null,share2u=null,share1ser=null,share3ser=null;
BufferedImage image;
rec=2;
 f =new File("E:\\red.jpg");
imguser =new FileInputStream(f);
image = ImageIO.read(f);
height = image.getHeight();
width = image.getWidth();
img_share=new int[3][height*width][100];
```

```
img1 = new
BufferedImage(image.getWidth(),image.getHeight(),BufferedImage.TYPE_INT_ARGB);
img2 = new
BufferedImage(image.getWidth(),image.getHeight(),BufferedImage.TYPE_INT_ARGB);
img3 = new
BufferedImage(image.getWidth(),image.getHeight(),BufferedImage.TYPE_INT_ARGB);
for(i=0;i<width;i++)
 {   for(j=0;j<height;j++)
       {
         Integer a=new Integer(5000000);
         a=image.getRGB(i,j);                  //get each pixel value
         b =Integer.toBinaryString(a);         //convert to binary
          for(h=0;h<32;h++)
          {
         if(b.charAt(h)=='1')      //ith position of PIX contains ‘1’
          {
            rand= Random_Place(3,2);  //call rand fn
            imgshare(h,rand,i,j); }
     }
    }
  }
for(k1=0;k1<3;k1++)
{   val="";
  for(k2=0;k2<width;k2++)
  {
   for(k4=0;k4<height;k4++)
     {
      for(k3=0;k3<32;k3++)
     {
        val=val+img_share[k1][k2*k4][k3];
     }
         argb(val,k2,k4,k1);
   }
  }
  System.out.println("share"+(k1+1)+""+val);
  System.out.println(val.length());
    switch(k1)
     {
       case 0:
            createImage(pixels1,width,height,k1);
            File outputFile1 = new File("E://share111.jpg");
            ImageIO.write(img1, "jpg", outputFile1);
            share1ser= new FileInputStream(outputFile1);
            {  po=0;
               ja=0; }
            break;
       case 1:
            createImage(pixels2,width,height,k1);
            File outputFile2 = new File("E://share112.jpg");
            ImageIO.write(img2, "jpg", outputFile2);
            share2u= new FileInputStream(outputFile2);
            {  po=0;
```

```
             ja=0;  }
             break;
        case 2:
             createImage(pixels3,width,height,k1);
             File outputFile3 = new File("E://share113.jpg");
             ImageIO.write(img3, "jpg", outputFile3);
             share3ser= new FileInputStream(outputFile3);
             break;
        default: break;
      } //end switch write img
     } // end k1 for loop
   try {
       Connection conn = DriverManager.getConnection(url, user, password);
       String sql1 = "INSERT INTO user(uid,name,image,share2) values (?,?,?,?)";
       String sql2 = "INSERT INTO server(sid,uid,share1,share3) values (?,?,?,?)";
         PreparedStatement statement1 = conn.prepareStatement(sql1);
         statement1.setString(1,"112");
         statement1.setString(2,"Tomyy");
         statement1.setBlob(3,imguser);
         statement1.setBlob(4,share2u);
         int row1 = statement1.executeUpdate();
         PreparedStatement statement2 = conn.prepareStatement(sql2);
         statement2.setString(1,"2");
         statement2.setString(2,"112");
         statement2.setBlob(3,share1ser);
         statement2.setBlob(4,share3ser);
         int row2 = statement2.executeUpdate();
        conn.close();
       }
       catch (SQLException ex) {
          ex.printStackTrace();
       }
} //end of main try
catch (IOException io)
   {  }
} //end main
public static int[] Random_Place(int n, int rec)
{  int j,i;
   int[] rand=new int[3];
   for(i=0;i<=2;i++)
   {  int nik=rand_int(0,2);
      if(nik!=rand[2])
      {   rand[i]=nik;
      }
   }
   return rand;
}

public static int rand_int(int g, int n)
{
   g=0;
```

```
    Random rand1 =new Random();
    int rand_into=rand1.nextInt((2-g)+1)+g;
    return rand_into;
}

public static void imgshare(int h,int []rand,int width,int height)
{
     int c,e;
     e=width*height;
    for(c=0;c<3;c++)
            {
             img_share[rand[c]][e][h] =1;
            }
 }

public static void argb(String val,int width,int height,int no)
{       int i,P,shareno;
        shareno=no;
        String value=val;
        int x= width;
        int y= height;
        i=ja;
        int B= Integer.parseInt(value.substring(i, i+8), 2);
        int G= Integer.parseInt(value.substring(i+8, i+16), 2);
        int R= Integer.parseInt(value.substring(i+16, i+24), 2);
        int A= Integer.parseInt(value.substring(i+24, i+32), 2);
   switch(shareno)
      {
        case 0:
                  pixels1[ki1] =R;
                  pixels1[ki1+1]=G;
                  pixels1[ki1+2]=B;
                pixels1[ki1+3]=A;
                  ki1 = ki1+4;
                  break;
        case 1:
                  pixels2[ki2] =R;
                  pixels2[ki2+1] =G;
                  pixels2[ki2+2] =B;
                 pixels2[ki2+3]=A;
                  ki2 = ki2+4;
                  break;
        case 2:
                  pixels3[ki3] =R;
                  pixels3[ki3+1] =G;
                  pixels3[ki3+2] =B;
                 pixels3[ki3+3]=A;
                  ki3 = ki3+4;
                  break;
        default: break;
      }
           ja=ja+32;
```

```
            po=po+32;
  }

 public static void createImage(int[] pixels,int width,int height,int no) throws IOException
   {
    switch(no)
    {
      case 0:
            WritableRaster raster1 = img1.getRaster();
            raster1.setPixels(0, 0, width, height, pixels1);
            break;
      case 1:
            WritableRaster raster2 = img2.getRaster();
            raster2.setPixels(0, 0, width, height, pixels2);
            break;
      case 2:
            WritableRaster raster3 = img3.getRaster();
            raster3.setPixels(0, 0, width, height, pixels3);
            break;
      default:break;
    }
   }
}
```

Stacking of Shares

```
import java.awt.image.BufferedImage;
import java.awt.image.WritableRaster;
import javax.imageio.ImageIO;
import java.lang.Integer.*;
import java.util.Scanner;
import java.util.Random;
import java.text.*;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

public class dec {

public static int kii=0;
public static int [] pixels1=new int [9000000];
public static int [] pixels2=new int [9000000];
public static void main(String[] args) throws IOException
{               String driverName = "com.mysql.jdbc.Driver";
                String url = "jdbc:mysql://localhost:3306/sharegen";
```

```
        String userName = "root";
        String password = "";
        Connection con = null;
        try{
                Class.forName(driverName);
                con = DriverManager.getConnection(url,userName,password);
                Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery("select share2 from user where uid=12");
                while (rs.next()) {
        InputStream in = rs.getBinaryStream(1);
       OutputStream f = new FileOutputStream(new File("E://ru//share2.jpg"));
                int c = 0;
                while ((c = in.read()) > -1) {
                                f.write(c);
                        }
                f.close();
                in.close();
        }
        ResultSet rs1 = stmt.executeQuery("select share1 from server where uid=12");
                while (rs1.next()) {
        InputStream in1 = rs1.getBinaryStream(1);
        OutputStream f1 = new FileOutputStream(new File("E://ru//share1.jpg"));
                 int c = 0;
                        while ((c = in1.read()) > -1) {
                                f1.write(c);
                        }
                f1.close();
                in1.close();
        }
        ResultSet rs2 = stmt.executeQuery("select share3 from server where uid=12");
                while (rs2.next()) {
        InputStream in2 = rs2.getBinaryStream(1);
        OutputStream f2 = new FileOutputStream(new File("E://ru//share3.jpg"));
                int c = 0;
                        while ((c = in2.read()) > -1) {
                                f2.write(c);
                        }
                f2.close();
                in2.close();
        }
        }catch(Exception ex){
                System.out.println(ex.getMessage());
        }
System.out.println("Enter total number of shares for regeneration" );
Scanner in = new Scanner(System.in);
int no = in.nextInt();
switch(no)
{
  case 1:System.out.println("Invalid input");
      break;
  case 2:
    rec2();
```

```
          break;
       case 3 :
          rec3();
          break;
     }

}
 public static void createImage(int[] pixels,int width,int height) throws IOException
   {    BufferedImage img100 = new
BufferedImage(width,height,BufferedImage.TYPE_INT_RGB);
             WritableRaster raster1 = img100.getRaster();
             raster1.setPixels(0, 0, width, height, pixels1);
             File outputFile1 = new File("E://output.jpg");
             ImageIO.write(img100, "jpg", outputFile1);
   }

 public static void rec2()
    {

          try {
                 File f4,f5;
                  int n=3,k=2,a,b,l,m,i,j;
                  Integer c=new Integer(9000000);
                  String z="",z1="",z2="",z3="";
                 f4 =new File("E:\\share111.jpg");
                  BufferedImage image1;
                 image1 = ImageIO.read(f4);
                 int h1 = image1.getHeight();
                 int w1 = image1.getWidth();

                 f5 =new File("E:\\share112.jpg");
                 BufferedImage image2;
                 image2 = ImageIO.read(f5);
                 int h2 = image2.getHeight();
                 int w2 = image2.getWidth();
                  for(l=0; l< w1 ;l++)
                  {
                          for(m=0; m<h1 ;m++)
                           {
                           c=image1.getRGB(l,m);              //get each pixel value share1
                           z =Integer.toBinaryString(c);
                           z1=z1+z;
                          }
                   }
                 for(l=0; l< w2 ;l++)
                  {
                   for(m=0; m< h2 ;  m++)
                          {

                           c=image2.getRGB(l,m);              //get each pixel value share2
                            z =Integer.toBinaryString(c);
                            z2=z2+z;
```

```
              }
          }
    int x,y;
    if (z1.length()>z2.length())
    {
        y=z1.length()-z2.length();
        while(y!=0 && z1.length()!=z2.length())
        {
          z2=z2+"0";
          y--;
        }
        x=z1.length();
    }
    else
        {
        y=z2.length()-z1.length();
        while(y!=0 && z2.length()!=z1.length())
        {
          z1=z1+"0";
          y--;
        }
         x=z2.length();
    }

    z3="";
    l=0;
    while(l!=x)
     {
        {
         if(z1.charAt(l)=='1' && z2.charAt(l)=='1' ||z1.charAt(l)=='1' && z2.charAt(l)=='0'
||z1.charAt(l)=='0' && z2.charAt(l)=='1')
         z3=z3+"1";
         else
         z3=z3+"0";
        }

        l++;
     int len=0,q,ja=0,d;
      y=x/32;
     d=z3.length()-(y*32);
       while(d%32!=0)
        {
          z3=z3+"0";
          d++;
        }
     int r=y*32+d;
     while(len!=r)
     {
               q =ja;
               int B= Integer.parseInt(z3.substring(q, q+8), 2);
               int G= Integer.parseInt(z3.substring(q+8,q+16), 2);
               int R= Integer.parseInt(z3.substring(q+16,q+24), 2);
          int A= Integer.parseInt(z3.substring(q+24, q+32), 2);

             pixels1[kii] =R;
             pixels1[kii+1]=G;
             pixels1[kii+2]=B;
             pixels1[kii+3]=A;
             kii = kii+4;

             ja=ja+32;
             len=len+32;
   }

        createImage(pixels1,w1,h1);
     }

catch (IOException io)
  {
    } }

public static void rec3()

  {
    try{
   File f4,f5,f6;
   int n=3,k=2,a,b,l,m,i,j;
    Integer c=new Integer(9000000);
    Integer c1=new Integer(9000000);
    Integer c2=new Integer(9000000);
   String z="",z1="",z2="",z3="",z4="",z0="",z00="";
f4 =new File("E://share111.jpg");
BufferedImage image1;
image1 = ImageIO.read(f4);
int h1 = image1.getHeight();
int w1 = image1.getWidth();

f5 =new File("E://share112.jpg");
BufferedImage image2;
image2 = ImageIO.read(f5);
int h2 = image2.getHeight();
int w2 = image2.getWidth();
f6 =new File("E://share113.jpg");
BufferedImage image3;
image3 = ImageIO.read(f6);
int h3 = image3.getHeight();
int w3 = image3.getWidth();

BufferedImage img1 = new BufferedImage(image1.getWidth(),image1.getHeight(),
BufferedImage.TYPE_INT_ARGB);

     for(l=0; l<w1 ;l++)
       {
         for(m=0; m<h1 ;m++)
```

```
          {
              c=image1.getRGB(l,m);                //get each pixel value share1
              z =Integer.toBinaryString(c);
              z1=z1+z;
              c1=image2.getRGB(l,m);                //get each pixel value share2
              z0 =Integer.toBinaryString(c1);
              z2=z2+z0;
              c2=image3.getRGB(l,m);                //get each pixel value share2
              z00 =Integer.toBinaryString(c2);
              z3=z3+z00;
          }
         }
   int x,y;
  if (z1.length()>z2.length())
  {
     y=z1.length()-z2.length();
     while(y!=0 && z2.length()!=z1.length())
     {
      z2=z2+"0";
      y--;
     }
     x=z1.length();
  }
  else
     {
     y=z2.length()-z1.length();
     while(y!=0 && z1.length()!=z2.length())
     {
      z1=z1+"0";
      y--;
     }
      x=z2.length();
  }

 int x1,y1;
  if (x>z3.length())
    {
     y1=x-z3.length();
     while(y1!=0 && z3.length()!=x)
     {
      z3=z3+"0";
      y1--;
     }
     x1=x;
    }
  else
    {
     y1=z3.length()-x;
     while(y1!=0 && x!=z3.length())
     {
      z1=z1+"0";
      z2=z2+"0";
```

```
         y1--;
         x++;
        }
       x1=z3.length();
      }
   l=0;
   z4="";
   while(l!=x1)
   {
     {
       if(z1.charAt(l)=='1' && z2.charAt(l)=='1' && z3.charAt(l)=='1'||z1.charAt(l)=='1' &&
z2.charAt(l)=='0' && z3.charAt(l)=='1' ||z1.charAt(l)=='0' && z2.charAt(l)=='1' &&
z3.charAt(l)=='1' ||z1.charAt(l)=='1' && z2.charAt(l)=='1' &&
z3.charAt(l)=='0'||z1.charAt(l)=='1' && z2.charAt(l)=='0' &&
z3.charAt(l)=='0'||z1.charAt(l)=='0' && z2.charAt(l)=='0' &&
z3.charAt(l)=='1'||z1.charAt(l)=='0' && z2.charAt(l)=='1' && z3.charAt(l)=='0')
         z4=z4+1;
        else
         z4=z4+0;   }
       l++;
   }
   int y4=x1/32;
   int d=z4.length()-(y4*32);
         while(d%32!=0)
        {
         z4=z4+"0";
         d++;
        }
    int r=y4*32+d;
    int len=0,ja=0;
    kii=0;
    while(len!=r)
    {        int q =ja;
             int B= Integer.parseInt(z4.substring(q, q+8), 2);
             int G= Integer.parseInt(z4.substring(q+8, q+16), 2);
             int R= Integer.parseInt(z4.substring(q+16, q+24), 2);
             int A= Integer.parseInt(z4.substring(q+24, q+32), 2);
                 pixels1[kii] =R;
                 pixels1[kii+1] =G;
                  pixels1[kii+2] =B;
                  pixels1[kii+3] =A;
                 kii = kii+4;
                 ja=ja+32;
                 len=len+32;
    }
          createImage(pixels1,w1,h1);
}
catch (IOException io)
   {
   }
}
```

Chapter 7

Testing

7.1 Test cases

Testing is not isolated to only one phase of project but also should be exercised in all phases of project. After developing each unit of the software product, the developer goes to an extensive testing process of software. After the development of the software modules, developers perform a thorough unit testing of each software component and also perform integration testing of all combine modules

Table 7.1.1. Test case share generation

Test Name	Share Generation
Test Objective	Convert image into shares
Test configuration	User selects a image
Procedure	• Take image from user • Divide the image into 3 shares • Store in user and server database
Action	Enter user image
Expected Result	Shares generated
Actual Result	Shares generated
Pass/fail	Pass

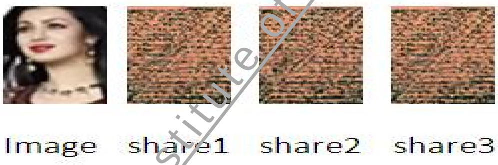


Figure 7.1.1 Share Generation

Table 7.1.2. Test case stacking of shares

Test Name	Stacking of Shares
Test Objective	Stack the shares and generate a image
Test configuration	Take shares from user and server
Procedure	Take user and server shares Stack the shares using or function Final single image is generated
Action	Take share from user and server database based on userid
Expected Result	Shares stacked and image generated
Actual Result	Image generated on stacking of shares
Pass/fail	pass

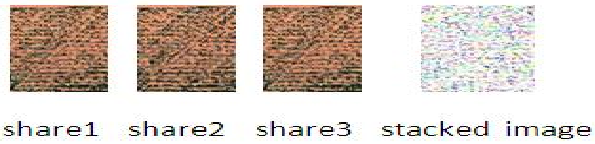


Figure 7.1.2 Stacking of shares

Chapter 8

Result and discussion

The previous anti phishing techniques did not help the user himself in detecting phishing. Further the basic version of visual cryptography worked only for black and white images.

Now in our system we have used the (2, 3) colored visual cryptography where we are helping the user himself in identifying the phished website. We tested our system when the user selects a image during registration the image is then divided into shares and the shares are stored in the user and server database. At the time of login, the shares from user and server were stacked together to generate the original image back. Depending on the image generated the user will be able to identify whether the site is phished or not.

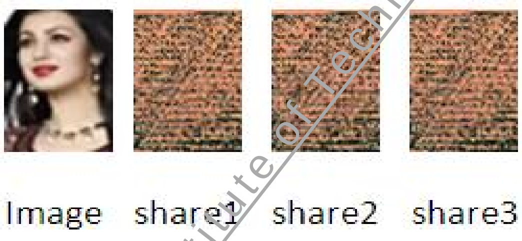


Figure 8.1 Output Share Generated

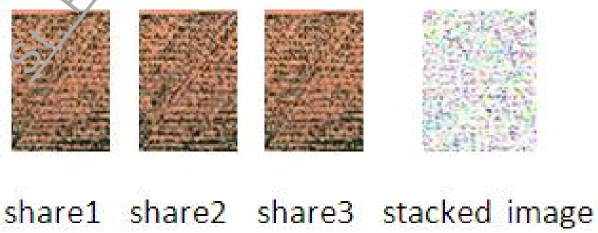


Figure 8.2 Output Stacked Image

Chapter 9
Conclusion and Future Scope

The system is mainly used in security domain which is related to online attacks. It provides a friendly environment to the user. General visual cryptography supports only black and white kind of images. Our system supports colored images.

In this project we have proposed a technique (2, 3) Visual Cryptography on color images. At the time of dividing an image into three numbers of shares we have used randomized algorithm. This technique needs very less mathematical calculation. Each share reflects very little or even no information regarding the original image to human eye. Further our system helps the users in identifying the phished webpage on their own this does not require any decryption technique. It is hence very useful as only encryption is required and decryption can be done by user.

In future to make the project more secure this implementation can be further extended as each time the user logs in and the stacked share shown to the user and he identifies the site is the original one he selects another image and the same process of share generation is repeated so each time user logs in with a new share this will help us prevent the attacker from gaining access to the share which is used each time we log in.

Appendix

A:

Anti-Phishing:

Anti-phishing services provide tools to help users recognize Web phishing.

P:

Phishing:

Phishing is the attempt to acquire sensitive information such as usernames, passwords, and credit card details (and sometimes, indirectly, money), often for malicious reasons, by masquerading as a trustworthy entity in an electronic communication.

S:

Shares:

Shares consist of Xeroxed transparencies which are stacked to recover the original image.

V:

Visual Cryptography:

Visual cryptography is a cryptographic technique which allows visual information (pictures, text, etc.) to be encrypted in such a way that decryption becomes a mechanical operation that does not require a computer.

Literature Cited

[1] Image Based Authentication Using Visual Cryptography and Encryption Algorithm Shreya Zarkar, Sayali vaidya, Arifa Tadvi, Tanashree Chavan, Prof. Achal Bharambe /(IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (2) , 2015, 1692-1696.

[2] K-n secret sharing visual cryptography scheme for color image using random number Shyamalendu Kandar et al. / International Journal of Engineering Science and Technology (IJEST)Haldia Institute of Technology, Haldia, India.

[3] M. Naor and A. Shamir, "Visual cryptography," Advances in Cryptology-Eurocrypt'94, pp. 1–12, 1995

[4] A Novel Anti Phishing framework based on Visual Cryptography. 978-1-4673-0449-8/12/$31.00 ©2012 IEEE.

[5] Design and Implementation of a (2, 2) and a (2, 3) Visual Cryptographic Scheme Special Issue of IJCCT Vol.1 Issue 2, 3, 4; 2010 for International Conference [ACCTA-2010], 3-5 August 2010

Acknowledgements

The work presented in this dissertation was completed as a part of B.E course of Computer Engineering at St. Francis Institute of Technology, Borivali (west).

Any project is not a product of one person but a team of people and our project is no such exception. Through this acknowledgement we express our gratitude to the people who have provided guidance and support which has helped us in our project.

With deep sense of gratitude we would like to acknowledge the inspiring guidance of our guide Ms. Vincy Joseph who has provided us with valuable inputs and constant support and motivation.

We humbly thank our Head of Department, Mrs. Bidisha Roy and other staff members for their help and providing facilities to complete this project successfully.

Ruchi Bhuta
Pooja Ekbote
Nikita Jadhav