SHIVAJI UNIVERSITY

A Novel on Antiphishing using Visual Cryptography

by

Author Name

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Department of Computer Science and Engineering
Sharad Institute of Technology College of Engineering,Yadrav-Ichalkaranji

April 2015

SHIVAJI UNIVERSITY

Abstract

Department of Computer Science and Engineering
Sharad Institute of Technology College of Engineering,Yadrav-Ichalkaranji

Doctor of Philosophy

zz by Author Name

Phishing is an attempt by an individual or a group to thieve personal confidential information such as passwords, credit card information etc.To do unsuspecting victims for identity theft, financial gain and other fraudulent activities. In this project we have proposed a new approach named as ”A Novel Antiphishing framework based on visual cryptography” to solve the problem of phishing. Here an image based authentication using Visual Cryptography (vc) is used. The use of visual cryptography is explored to preserve the privacy of image captcha by decomposing the original image captcha into two shares that are stored in separate database servers. The half share of client is in encrypted format to preserve the privacy.The original image captcha can be revealed only when both are simultaneously available the individual sheet images do not reveal the identity of the original image captcha. Once the original image captcha is revealed to the user it can be used as the password.

Acknowledgements

We take this opportunity to express our profound gratitude and deep regards to our guide Mr. S.Gurav for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him time to time shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to our Principal, Dr. S. A. Khot and our H.O.D., Prof. O.D.Joshi for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are obliged to all faculty and staff members of all departments of our Sharad Institute of Technology College of Engineering, Yadrav for the valuable information provided by them in their respective fields. We are grateful for their cooperation during the period of our assignment.

Lastly, we thank almighty, our parents, brothers, sisters, friends and our colleagues for their constant encouragement without which this assignment would not be possible.

Miss.Priyanka Lajpatrai Lodha

Miss. Bhavana Balgonda Patil

Miss. Sarita Gajanan Latthe

Miss. Komal Arun Khandekar

Miss. Shweta Gajanan Lambet

Contents

Abstract **i**

Acknowledgements **ii**

List of Figures **v**

1 INTRODUCTION **1**
1.1 Overview 1
1.2 Registration phase 2
1.3 Login phase 2
1.4 How it works ? 3
1.4.1 Registration phase 3
1.4.2 Login phase 3

2 LITERATURE REVIEW **5**
2.1 Captcha technique 5
2.2 Blacklist-based technique 5
2.3 Heuristic technique 5
2.4 Checking by IP address 6
2.5 Single sign on 6
2.6 Assessment based technique 6
2.7 Cryptography 6

3 OBJECTIVE AND SCOPE **7**
3.1 Objective 7
3.2 Scope 7
3.3 Out of scope 8

4 REQUIREMENT ANALYSIS **9**
4.1 Software requirements 9
4.1.1 ASP.NET 4.0 framework 9
4.1.2 C # (Visual studio 2010) 10
4.1.3 VB script 10
4.1.4 Internet information services 11
4.1.5 Oracle 10g 11
4.2 Hardware requirement 12

5 DESIGN **13**
5.1 System flow diagram 13
5.2 Use case diagram 15
5.3 Sequence diagram 18
5.4 Class diagram 20
5.5 Activity diagram 21
5.6 Deployment diagram 24

6 Coding **26**
6.1 C# 26
6.2 Features of C# 27
6.3 Coding 28
6.3.1 Code of generation of captcha 28
6.3.2 Code of spliting of captcha 29
6.3.3 Merging of captcha 32

7 TESTING **33**
7.1 What is software testing? 33
7.1.1 Goal 33
7.2 Test planning 34
7.2.1 Test cases 36

8 SNAPSHOTS **42**
8.1 Registration 42
8.2 Captcha generation 43
8.3 Login 44
8.4 Encrypted image captcha of client and server 45
8.5 User's account 45

9 CONCLUSION **46**
9.1 Conclusion 46

Bibliography **47**

List of Figures

1.1 Genaration of encrypted image captcha. 2

5.1 System flow diagram of registration phase 14
5.2 System flow diagram of login phase 15
5.3 Use case diagram 16
5.4 Sequence daigram for registration phase 18
5.5 Sequence daigram for Login phase 19
5.6 Class diagram 20
5.7 Activity diagram for registration phase 22
5.8 Activity diagram for login phase 23
5.9 Deployment diagram 25

6.1 Architecture of ASP.NET 26

8.1 Registration page 42
8.2 Captcha generation. 43
8.3 Login page 44
8.4 Encrpted image captcha of client 45
8.5 Encrpted image captcha of server 45
8.6 User's account window. 45

Chapter 1

INTRODUCTION

1.1 Overview

Recent Survey reveal that 90 percent of website have been hacked, among them the most popular attack is phishing. Phishing is an attempt by an individual or a group to get personal confidential information such as passwords, credit card information from unsuspecting victims for identity theft, financial gain and other fraudulent activities. Fake websites which appear very similar to the original ones are being hosted to achieve this. We have proposed a new approach named as "A Novel Anti-phishing framework based on visual cryptography "to solve the problem of phishing. Here an image based authentication using Visual Cryptography is implemented. The use of visual cryptography is explored to preserve the Privacy of an image captcha(Completely Automated Public Turing Test to tell Computers and Humans Apart) by decomposing the original image captcha into two shares (known as sheets) that are stored in separate database servers(one with user and one with server) such that the original image captcha can be revealed only when both are simultaneously available; the individual sheet images do not reveal the identity of the original image captcha. Once the original image captcha is revealed to the user it can be used as the password. Using this website cross verifies its identity and proves that. For phishing detection and prevention, we are proposing a new methodology to detect the phishing website. Our methodology is based on the Anti-Phishing Image Captcha validation scheme using visual cryptography. It prevents password and other confidential information from the phishing websites. The proposed approach can be divided into two phases:

- Registration Phase
- Login Phase

1.2 Registration phase

In the registration phase, User detailsare asked from the user at the time of registration for the secure website. The password can be a combination of alphabets and numbers to provide more secure environment. From password server genarate the Encrypted Image captcha .The image captcha is divided into two shares such that one of the share is kept with the user database and the other share is kept in the server database on server side. The image captcha is also stored in the actual database of any confidential website as confidential data Server ganerate random string (secret key) and send to user for login password in login phase.

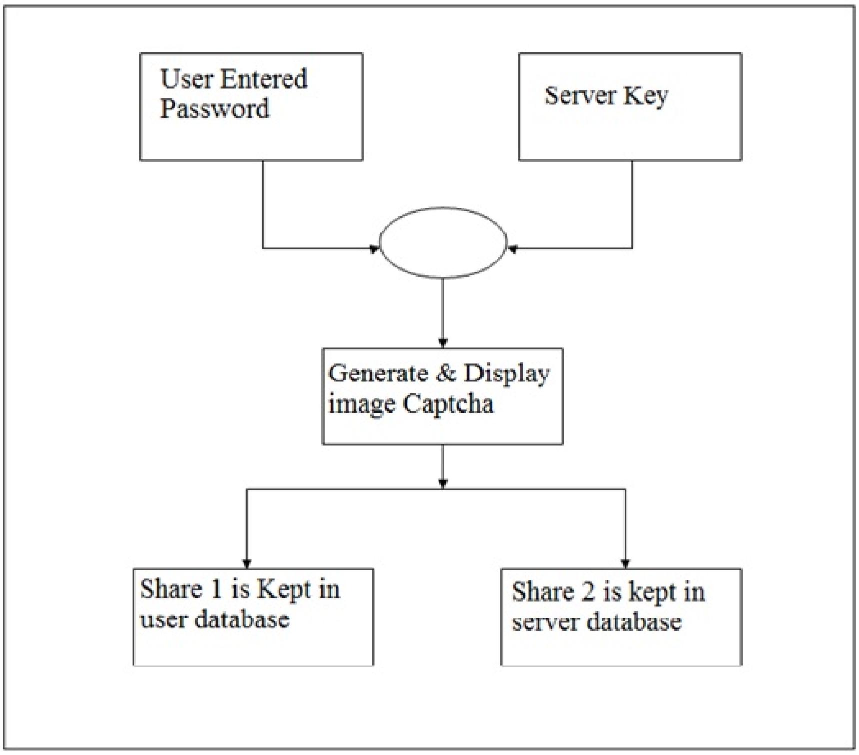

User Entered Password
Server Key
Generate & Display image Captcha
Share 1 is Kept in user database
Share 2 is kept in server database

Figure 1.1: Genaration of encrypted image captcha.

1.3 Login phase

When the user logs in by entering his confidential information for using his account, then first the user is asked to enter his username and password,with respective to them collects user share and server share from its databses are stacked together to produce the image captcha. The reconstructed image captcha is displayed to the user .Here the end user can check whether the displayed image captcha matches with the captcha created at the time of registration. The end user is required to enter the text displayed in the image captcha. After this enter the server string (Secret key) to login to website.Using the username, password and image captcha generated by stacking two shares one can verify whether the website is genuine/secure website or a phishing website and can also verify whether the user is a human user or not.Also by using secret key we can determine user is authonticated or not.

1.4 How it works ?

1.4.1 Registration phase

- Open required website.
- Display the Registration Form.
- A. Enter the user details like username, password and other details.
- B. It will generate the an Encrypted Image CAPTCHA by using combination of user Password and server key and Display the Encrypted CAPTCHA to the User.
- C. After displaying Encrypted CAPTCHA server will generate one string (Secret Key) which is give to user for later verification.
- D. When user complete registration process, in background Image CAPTCHA will be divided into the two shares (share 1 and share 2). We will create two database tables like client database which includes columns like User name, Password, Share 1.Also server database includes columns like User name, password , share 2 and Server string.

1.4.2 Login phase

Case I: (If website is genuine)

- **Enter the user name and password.**
- **With respect to user name and password share 1 and share 2 are taken from both database tables.**
- **Server will generate the image CAPTCHA by using user share and server share and display original reconstructed image CAPTCHA to the user.**

- **User will confirm the reconstructed image CAPTCHA is same as it was at the time of Registration. This step will confirm that the website is genuine or not.**
- **Enter the text from reconstructed image CAPTCHA.**
- **After this user will enter the server string (Secret Key).**
- **Login is successfully.**

Case II: (If website is phishing website)

- **If an attacker knows user name and password but still he doesnt have the user image shares of image CAPTCHA.**
- **If website is phishing website, it is not able to reconstruct the Original image CAPTCHA for that particular user.**

Chapter 2

LITERATURE REVIEW

2.1 Captcha technique

The use of visual cryptography is explored to preserve the Privacy of an image captcha by decomposing the original image captcha into two shares (known as sheets) that are stored in separate database servers(one with user and one with server) such that the original image captcha can be revealed only when both are simultaneously available.[1]

2.2 Blacklist-based technique

It is an Domain Name Server based approach various browser Such as Anti Phishing Work Group, Google and other organizations maintain Black List technique. The demerit of this technique is that Phishing websites we were found is in very small proportion, so the failed alarm probability is very high. The life cycle of a phishing website is only a few days. Before detecting the Phishing site website might be shut down.[2]

2.3 Heuristic technique

It estimate weather a page has some phishing heuristic characteristics it include checking URL, host name and previously seen images. Various Rules are applied for defining the heuristic characters. But this technique is not secure because Hackers can define rules and this rules can be break by anyone. The Accuracy is not enough.

2.4 Checking by IP address

In this Technique checking of IP address of particular site is done but if there is small change in URL or in IP Address then we cannot identify the phishing site. IP address of all the Website must be stored in blacklist but if IP address is not found in Blacklist then it does not provide the security.[3]

2.5 Single sign on

Its a Simple and Convient technique .It provides the Single Sign On(Sso) i.e User can Login Once and can Use Multiple site. In this Technique two passwords are used temporary and Fixed. Temporary password can be used from any PC for accessing the Website. Fixed Password must be used from particular PC which was used at the time of Registration. Using two Passwords Fixed and Temporary provides Conflict.[4]

2.6 Assessment based technique

It needs too long time to calculate a pair of pages, so using the method to detect phishing websites on the client terminal is not suitable. And there is low accuracy rate for this method depends on many factors, such as the text, images, and similarity measurement technique. However, this technique (in particular, image similarity identification technique) is not perfect enough yet.[5]

2.7 Cryptography

Information security is the process of protecting information. It protects its availability, privacy and integrity. Access to stored information on computer databases has increased greatly. More companies store business and individual information on computer than ever before. Much of the information stored is highly confidential and not for public viewing. This paper have developed a cryptography algorithm which is based on block cipher concept.

Chapter 3

OBJECTIVE AND SCOPE

3.1 Objective

The main objectives of our project are:

- **To Provide the Security.**
- **To Detect the Phishing website.**
- **To Avoid the Phishing Site to get the confidential information.**
- **To distinguish between human User and Machine User.**

The Security is provided by using Visual Cryptography. Visual Cryptography is a special encryption technique to hide information in images in such a way that it can be decrypted by the human vision if the correct key image is used. The Visual Cryptography uses two transparent images. One image will be at Client Side and the other image will be at Server side. It is impossible to retrieve the secret information from one of the images. Both transparent images or layers are required to reveal the information.

3.2 Scope

This Project is mainly used for Security. To achieve this we are using an Image Based Authentication

- **From user details server will generate captcha**

- **Captcha will get divided into two shares.**
- **Half share of the captcha will be with client database and another half share will be with server database.**
- **Both the Client and Server captcha is get compared.**
- **The designed system should tested with performance measuring parameters.**

3.3 Out of scope

- **It Provides Security to the Banking Website but this technique can also be used for Facebook, g-mails.**
- **Image Shares can be divided into more than two shares to provide more security.**

Chapter 4

REQUIREMENT ANALYSIS

4.1 Software requirements

- **Operating System: Windows 7 or above, Windows 2003**
- **Web Server: IIS (Internet Information Services)**
- **Front End: ASP.NET Framework 4.0**
- **Middleware: C # (Visual Studio 2010)**
- **Back End: Oracle 10g**
- **Other Technologies: Vb Script or Java Script, CSS, XML**
- **Browser: any browser**

4.1.1 ASP.NET 4.0 framework

Description
ASP.NET 4.0 application provides support to websites as well as web application projects. The default ASP.NET application provides a good starter project template with some common items which have to be built for almost all web based projects like a Master Page with a standard template, About us page, Login page, Register page, Change Password page, a default style sheet named Site.css which would already have been referred in the Master page, etc., This undoubtedly would reduce a lot of development effort. All you need to do is simply customize the predefined templates to your needs. If the developer is not interested in the default templates, ASP.NET 4.0

also provides web based projects called Empty Web Application and Empty Website, which will not have any default items in the project except the web.

4.1.2 C # (Visual studio 2010)

Description
It is an object-oriented language, C # supports the concepts of encapsulation, inheritance, and polymorphism. All variables and methods, including the Main method, the application's entry point, are encapsulated within class definitions. A class may inherit directly from one parent class, but it may implement any number of interfaces. Methods that override virtual methods in a parent class require the override keyword as a way to avoid accidental redefinition. In C#, a struct is like a lightweight class; it is a stack-allocated type that can implement interfaces but does not support inheritance. In addition to these basic object-oriented principles, C# makes it easy to develop software components through several innovative language constructs, including the following:

- Encapsulated method signatures called delegates, which enable type-safe event notifications.
- Properties, which serve as assessors for private member variables.
- Attributes, which provide declarative metadata about types at run time.
- Inline XML documentation comments.
- Language-Integrated Query (LINQ) which provides built-in query capabilities across a variety of data sources.

4.1.3 VB script

Description
VBScript stands for Visual Basic Script, a scripting language developed by Microsoft to be used with Microsoft products, mainly Internet Explorer. It has gone through many changes over the years and is now mainly used as the default scripting language of ASP.

- VBScript is a scripting language
- A scripting language is a lightweight programming language

- **VBScript is a light version of Microsoft's programming language Visual Basic**
- **VBScript is the default language in ASP (Active Server Pages)**

4.1.4 Internet information services

Description

Internet Information Services (IIS, formerly Internet Information Server) is an extensible web server created by Microsoft for use with Windows NT family. IIS supports HTTP, HTTPS, FTP, FTPS, SMTP and NNTP. It has been an integral part of the Windows NT family since Windows NT 4.0, though it may be absent from some editions (e.g. Windows XP Home edition). IIS is not turned on by default when Windows is installed. The IIS Manager is accessed through the Microsoft Management Console or Administrative Tools in the Control Panel.

4.1.5 Oracle 10g

Description

Database 10g provides a robust and complete grid computing solution that enables companies to easily align their resources as required. Information integration is a critical component of these solutions, as it enables companies to access information when and where its needed in a distributed environment. Oracle10g offers the most complete and the most comprehensive platform for information integration. As is demonstrated by its extensive history running critical business applications for the most demanding solutions. Oracle10g provides a robust set of features critical for integration, including high availability, security, scalability, and flexibility. It offers secure and standard communication mechanisms that enable communication between applications/users on the Oracle database using queues, data replication and distributed access in both homogeneous and heterogeneous environments.

4.2 Hardware requirement

- **Processor C2D 2.0 GHz or above**
- **Memory 4 GB or more**
- **N/W Card 10/100/1000 mbps**

Chapter 5

DESIGN

5.1 System flow diagram

Description

A system flow diagram can represent activities as simple as following a recipe or as complicated as international trade. A visual representation gives members of an organization a sense of their place in the entire system. Basic symbols used are flow charts usually include input, flow lines, process and output. The output from one process can begin another process as an input and multiple processes can be added to an entire system flow diagram. In the System flow diagram user will enter all the details about user. From the password of user server will generate captcha. Server will send key (half share of captcha) to user which will act as password in login phase. captcha is get divided into two shares and it is stored in client and server databases.

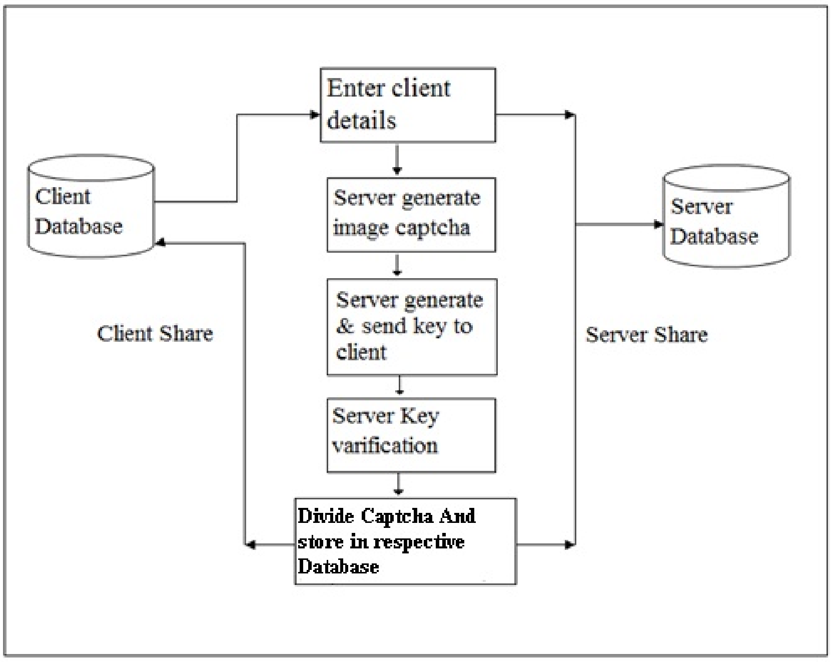

Enter client details
Server generate image captcha
Server generate & send key to client
Server Key varification
Divide Captcha And store in respective Database
Client Database
Client Share
Server Database
Server Share

Figure 5.1: System flow diagram of registration phase

User will enter its user name ,password and half captcha which was given at the time of registration.The half share which was kept with server will get combined with client share.If both shares matches then the website is geniune.

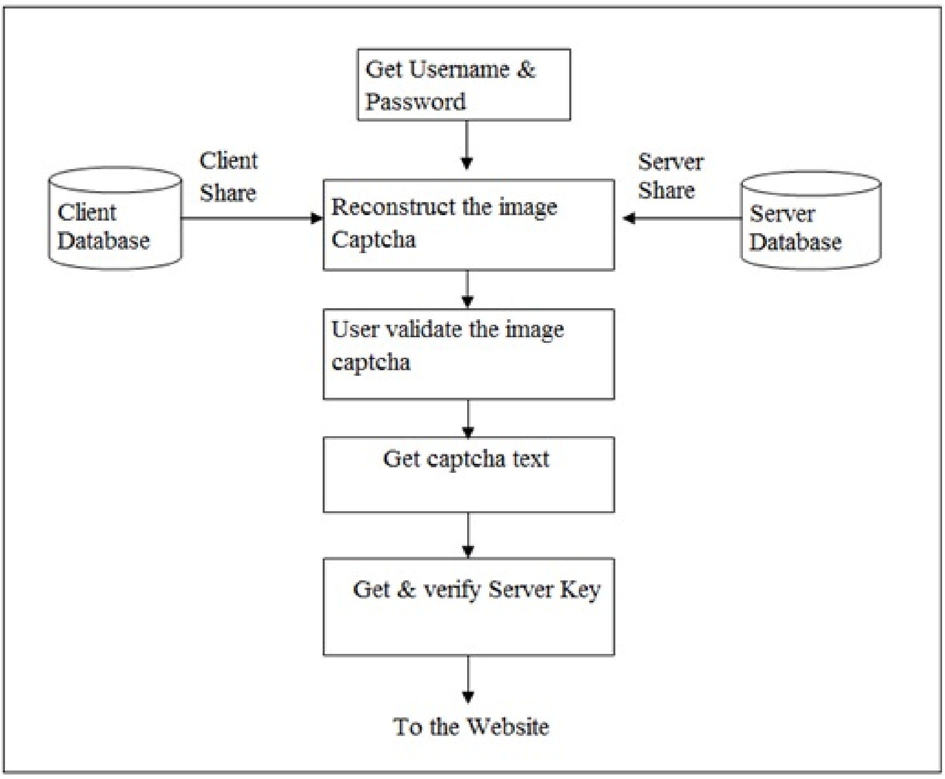


FIGURE 5.2: System flow diagram of login phase

5.2 Use case diagram

Description

A use case diagram is the simplest representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can define the different types of users of a system and the various ways that they interact with the system. They provide the simplified and graphical representation of what the system must actually do. The purpose of the use case diagrams is simply to provide the high level view of the system and convey the requirements. User is the main actor in the use case diagram. Actor will register its details and it will store in database. The same actor will login if he is registered and the login information is stored in database.

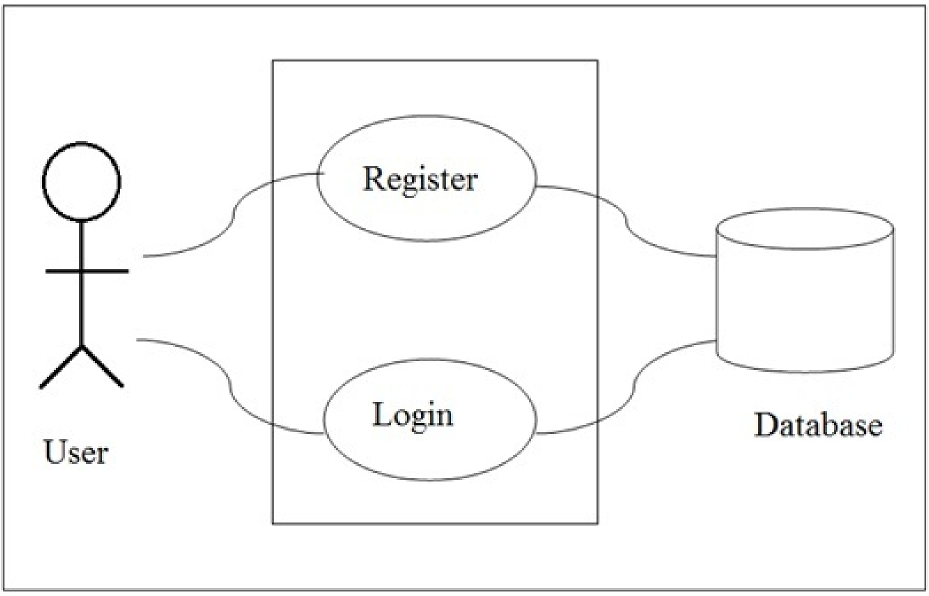


Figure 5.3: Use case diagram

Use case	Description
Register	 User is the main Actor in this phase. The main flow of events are as follow: 1. User will enter name. 2. User will enter address. 3. User will enter email id. 4. User will enter address. 5. User will enter contact no. 6. User will enter gender. 7. User will enter date of birth. 8. User will enter user name. 9. User will enter password. 10. User clicks on submit button. 11. Details are get successfully stored in server database. 12. The server will create captcha. Captcha will get divided into two parts. 13. One part will be stored in server database 14. Another half Share will be downloaded by user. 15. User will get successfully registered.
Login	User is the main actor in this phase. The main flow of events are as follow:: 1. User enter username. 2. User will enter password. 3. User will upload half share of captcha. 4. User client side captcha is combined with server side captcha. 5. If combined captcha is matched with original captcha then login is successful.

Table 5.1: Use case scenario

5.3 Sequence diagram

Description

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios. When user is entering all the details Registerform is active. When user clicks on submit button information is stored in database. While storing the information Register.cs is active.

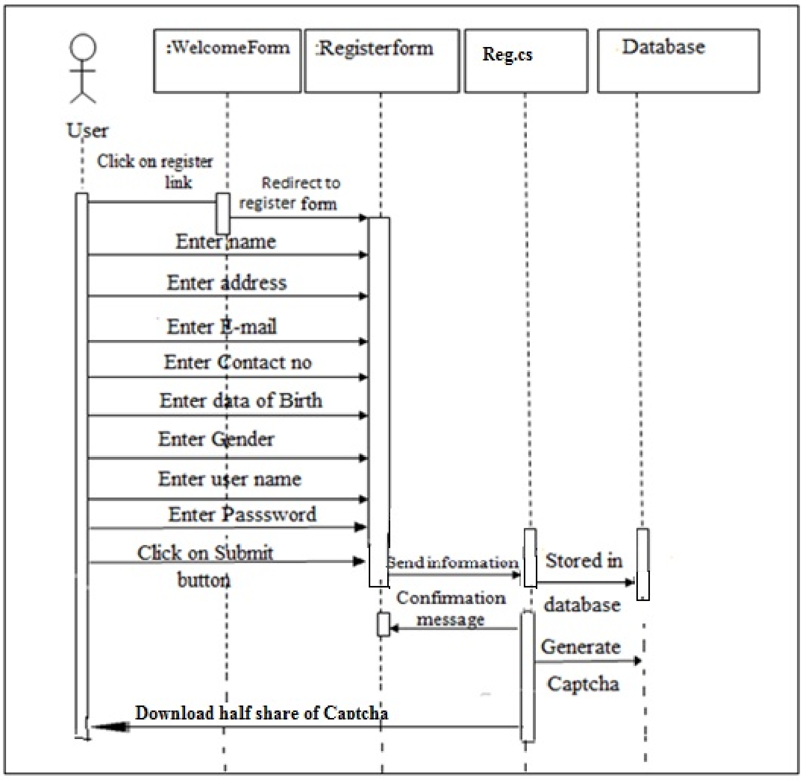


Figure 5.4: Sequence daigram for registration phase

There are three events in login phase. User will enter user name and password which was given at the time of registration. If the User name and password matches then user will upload his share at this time login event is active. While storing information database event is active.

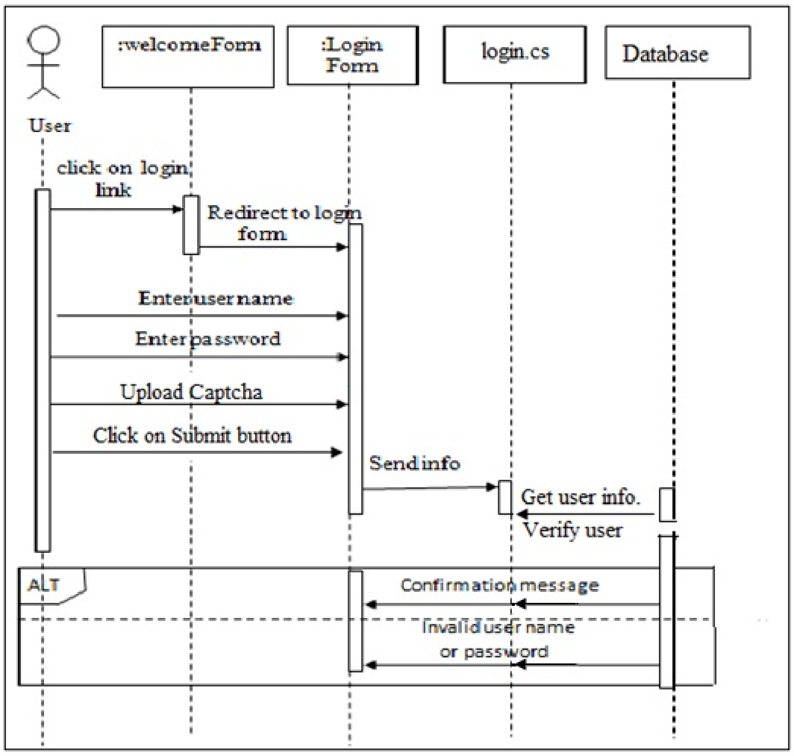


Figure 5.5: Sequence daigram for Login phase

5.4 Class diagram

Description

A UML class diagram describes the object and information structures used by your application, both internally and in communication with its users. It describes the information without reference to any particular implementation. Its classes and relationships can be implemented in many ways, such as database tables, XML nodes, or compositions of software objects.

- **Class: A definition of objects that share given structural or behavioral characteristics.**
- **Attribute: A typed value attached to each instance of a classifier.**
- **Operation: A method or function that can be performed by instances of a classifier.**

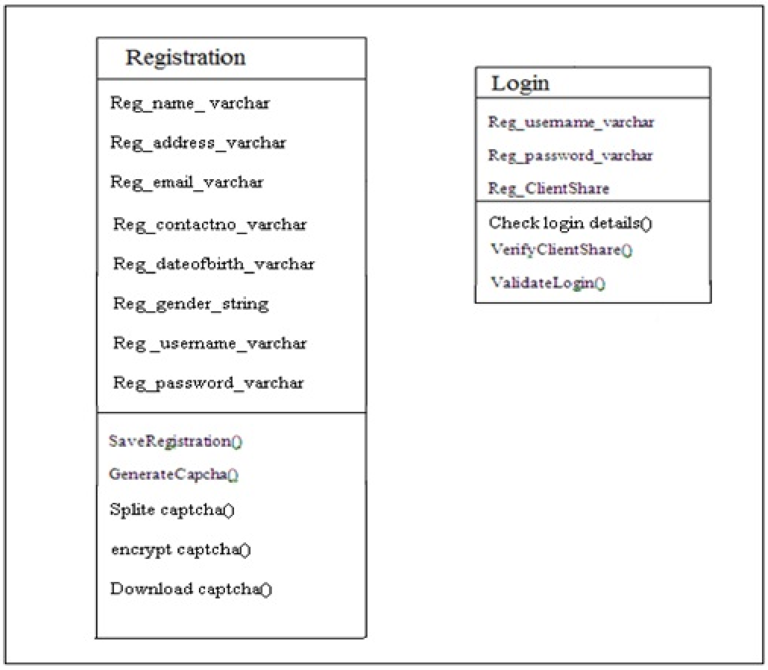


FIGURE 5.6: Class diagram

5.5 Activity diagram

Description

Activity diagrams are graphical representations of workflows of stepwise activities and action with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes. Activity diagrams show the overall flow of control. Activity diagrams are constructed from a limited number of shapes, connected with arrows. The most important shape types are:

- **Rounded rectangles represent actions,**
- **Diamonds represent decisions,**
- **Bars represent the start (split) or end (join) of concurrent activities,**
- **A black circle represents the start (initial state) of the workflow,**
- **An encircled black circle represents the end (final state).**

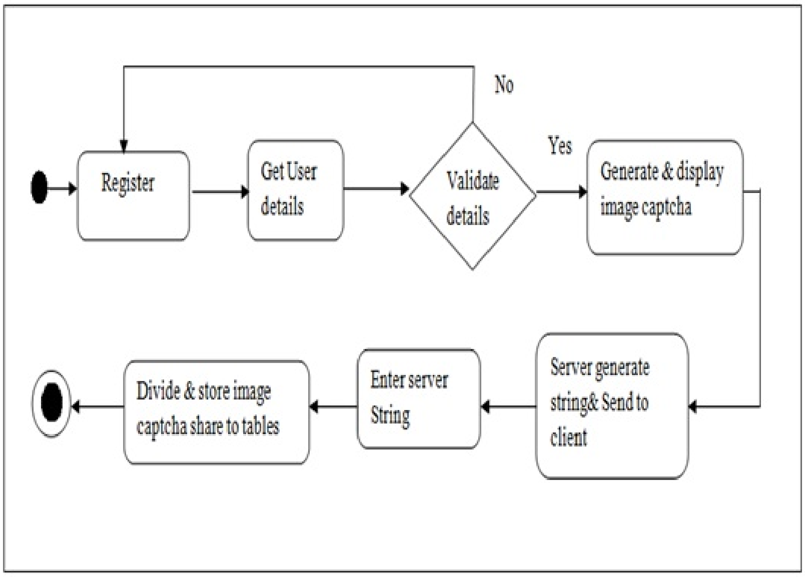

No
Yes
Register
Get User details
Validate details
Generate & display image captcha
Server generate string& Send to client
Enter server String
Divide & store image captcha share to tables

Figure 5.7: Activity diagram for registration phase

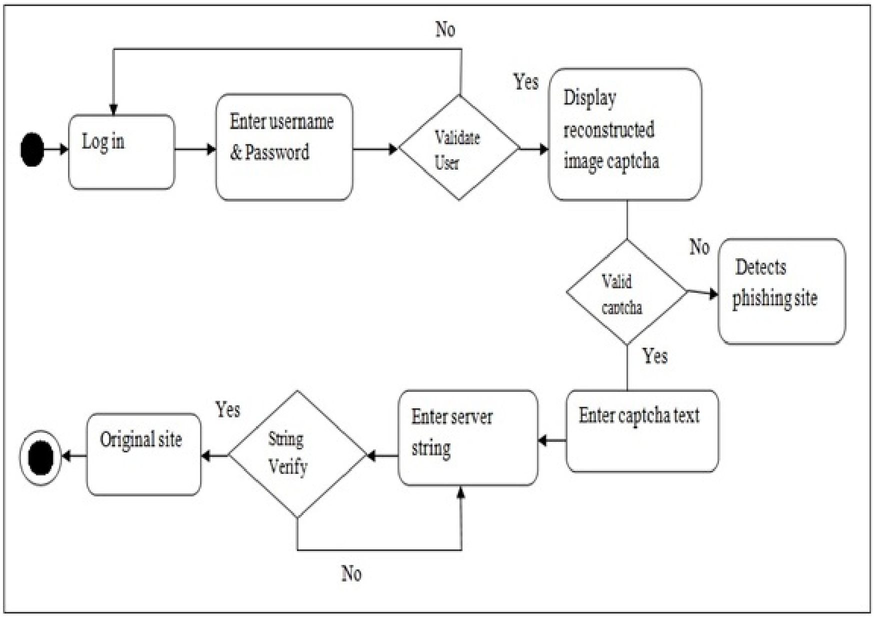


Figure 5.8: Activity diagram for login phase

In the Login Phase only the registered User can Login. The User will enter User name and Password. If provided details are Valid then user will upload his half Share, which was Provided at the time of Registration.

5.6 Deployment diagram

Description

A deployment diagram in the Unified Modeling Language models the physical deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components (”nodes”) exist (e.g., a web server, an application server, and a database server), what software components (”artifacts”) run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI). The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers. There are two types of Nodes:

- **Device node**
- **Execution environment node**

Device nodes are physical computing resources with processing memory and services to execute software, such as typical computers or mobile phones. An execution environment node (EEN) is a software computing resource that runs within an outer node and which itself provides a service to host and execute other executable software elements.

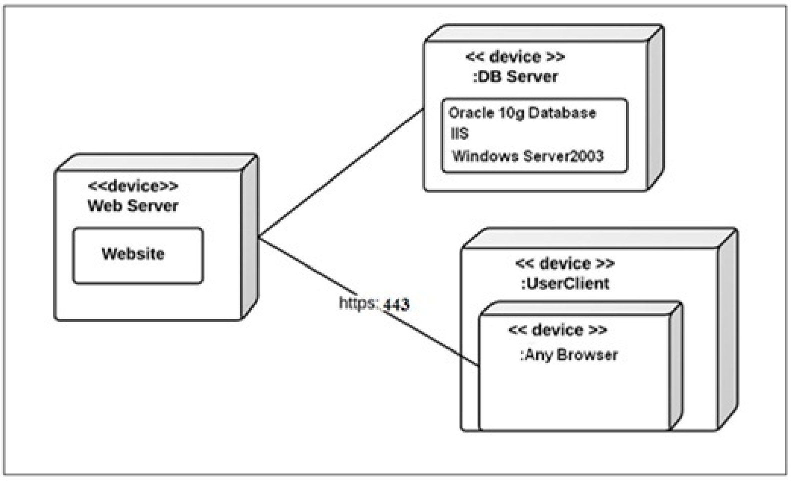


FIGURE 5.9: Deployment diagram

Chapter 6

Coding

6.1 C#

C# provides the easiest and most productive language tool for rapidly building windows and web based applications. C# comes with enhanced visual designers increased application performance and a powerful integrated development (IDE). It also supports creation of application for wireless, internet enabled hand-held devices.

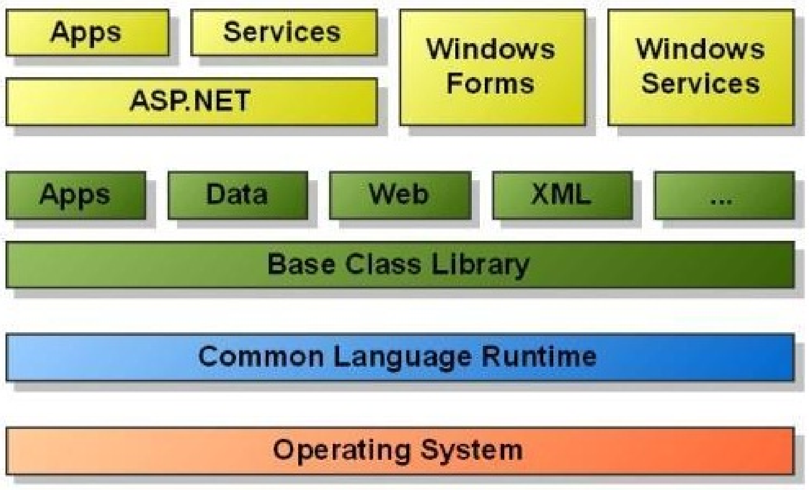


Figure 6.1: Architecture of ASP.NET

6.2 Features of C#

1. Powerful window based application:

C# comes with features such as a powerful new forms designer, an in-place menu editor and automatic control anchoring and docking. C# delivers new productivity features for building more robust applications easily and quickly. With an improved Integrated Development Environment (IDE) and significantly reduced start up time, C# offers fast, automatic formatting of code as you type, improved, an enhanced object browser and XML designer and much more.

2. Building Web based application:

With C# we can create web application using the shared Web Forms Designer and the familiar drag and drop feature. You can double click and write a code to respond to events. We can choose the editor for visual authoring of intelligence interactive web application.

3. Simplified deployment:

With C# we can build applications more rapidly and deploy and maintain them with efficiency. C# 2008 and .NET framework 3.0 makes DLL HELL a thing of the past. Side by-side versioning enables multiple versions of the same component. XCOPY deployment and web auto download of windows based application combine the simplicity of web page deployment and maintenance with the power of rich, responsive windows based application.

4. Powerful, simplified, flexible data access:

You can tackle any data access scenario easily with ADO.NET and ADO data access. The flexibility of data access enables data binding to any database, as well as classs collection and arrays and provides true XML representation of data. Seamless access to ADO enables simple data access for connected data binding scenarios. Using ADO.NET, C# can gain high speed access to MS-SQL server, oracle, DB2, Microsoft and more.

5. Improved coding:

You can code faster and more effectively. A multitude of enhancements to the code editor, including intelligence, smart listing of code for greater readability and a background compiler for real time notification of syntax errors transforms into a rapid application development (RAD) coding machine.

6. Direct access to the platform:

C# applications can have full access to the capabilities available in .NET framework 3.0 developers can easily program system services including the event log, performance, counters and file systems. The new window service project templates enable to build real Microsoft NT services. Programming against windows services and creating new window services is not available in C# standard, it requires Visual Basic 2008 professional or higher.

7. Full object oriented constructs:

You can create reusable, enterprise- class, code using full object oriented constructs. Language features include full implementation inheritance, encapsulation and polymorphism. Structured exception handling provides a global error handler.

6.3 Coding

6.3.1 Code of generation of captcha

```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
using System.Drawing;
using System.Drawing.Imaging;
using System.Text;
using System.Net;
using System.IO;

public partial class _Default : System.Web.UI.Page
{
    string code = string.Empty;
    private Random rand = new Random();
    string c;
    private Size IMAGE_SIZE = new Size(260, 60);
    private const int GENERATE_IMAGE_COUNT = 3;
    // string code1 = string.Empty;
    private Bitmap[] m_EncryptedImages;
    protected void Page_Load(object sender, EventArgs e)
    {
        //            btn_refrsh_Click(sender, e);
        if (!IsPostBack)
        {
            StringBuilder randomText = new StringBuilder();
            string alphabets = "012345679ACEFGHKLMNPRSWXZabcdefghijkhlmnopqrstuvwxyz";
            Random r = new Random();

            for (int j = 0; j <= 4; j++)
            {
                randomText.Append(alphabets[r.Next(alphabets.Length)]);
            }
            c = randomText.ToString();
            Session["CaptchaCode"] = c;


            lblcaptcha.Text = c;

            // CreateCaptchaImage();
            reg_captcha.ImageUrl = "~/captchgenerate.aspx?New=";
```

6.3.2 Code of spliting of captcha

```
string captcha = Session["CaptchaCode"].ToString();
            m_EncryptedImages = SplitImage(captcha);
            m_EncryptedImages[0].Save(Server.MapPath("~\\split\\" + regno + "_S.jpg"));
            m_EncryptedImages[1].Save(Server.MapPath("~\\split\\" + regno + "_C.jpg"));
            m_EncryptedImages[2].Save(Server.MapPath("~\\split\\" + regno + "_RC.jpg"));

        }
        //Image1.ImageUrl = Server.MapPath("~\\split\\1_S.jpg");

        //Image2.ImageUrl = Server.MapPath("~\\split\\1_C.jpg");



    }

    private Bitmap[] SplitImage(string inputText)
    {
        Bitmap finalImage = new Bitmap(IMAGE_SIZE.Width, IMAGE_SIZE.Height);
        Bitmap tempImage = new Bitmap(IMAGE_SIZE.Width / 2, IMAGE_SIZE.Height);
        Bitmap[] image = new Bitmap[GENERATE_IMAGE_COUNT];

        Random rand = new Random();
        SolidBrush brush = new SolidBrush(Color.Red);
        // Point mid = new Point(IMAGE_SIZE.Width / 2, IMAGE_SIZE.Height / 2);

        Graphics g = Graphics.FromImage(finalImage);
        Graphics gtemp = Graphics.FromImage(tempImage);

        StringFormat sf = new StringFormat();
```

```
sf.Alignment = StringAlignment.Center;
sf.LineAlignment = StringAlignment.Center;
// Font font = new Font("Times New Roman", 60);
Color fontColor;

//g.DrawString(inputText, font, brush, mid, sf);
SolidBrush black = new SolidBrush(Color.Black);
string code = string.Empty;
code = inputText;
int counter = 0;
for (int i = 0; i < code.Length; i++)
{
    g.DrawString(code[i].ToString(), new Font("Times New Roman", 10 + rand.Next(15, 20),
    counter += 28;
}

gtemp.DrawImage(finalImage, 0, 0, tempImage.Width, tempImage.Height);




for (int i = 0; i < image.Length; i++)
{
    image[i] = new Bitmap(IMAGE_SIZE.Width, IMAGE_SIZE.Height);
}


int index = -1;
int width = tempImage.Width;
int height = tempImage.Height;

for (int x = 0; x < width; x += 1)
{
    for (int y = 0; y < height; y += 1)
    {
        fontColor = tempImage.GetPixel(x, y);
        index = rand.Next(image.Length);
        if (fontColor.Name == Color.Empty.Name)
        {
            for (int i = 0; i < image.Length - 1; i++)
            {

                if (index == 0)
                {
                    image[i].SetPixel(x * 2, y, Color.Black);
                    image[i].SetPixel(x * 2 + 1, y, Color.Empty);
                }
                else
                {
                    image[i].SetPixel(x * 2, y, Color.Empty);
                    image[i].SetPixel(x * 2 + 1, y, Color.Black);
                }
                //////////////
```

```
                    }
                }
                else
                {
                    for (int i = 0; i < image.Length - 1; i++)
                    {

                        if ((index + i) % image.Length == 0)
                        {
                            image[i].SetPixel(x * 2, y, Color.DarkSlateGray);
                            image[i].SetPixel(x * 2 + 1, y, Color.Empty);
                        }
                        else
                        {
                            image[i].SetPixel(x * 2, y, Color.Empty);
                            image[i].SetPixel(x * 2 + 1, y, Color.DarkViolet);
                        }
                        ////////////


                    }
                }
            }
        }

        for (int x = 0; x < width; x += 1)
        {
            for (int y = 0; y < height; y += 1)
            {
                fontColor = tempImage.GetPixel(x, y);
                index = rand.Next(image.Length);
                if (fontColor.Name == Color.Empty.Name)
                {

                    if (index == 0)
                    {
                        image[2].SetPixel(x * 2, y, Color.Black);
                        image[2].SetPixel(x * 2 + 1, y, Color.Empty);
                    }
                    else
                    {
                        image[2].SetPixel(x * 2, y, Color.Empty);
                        image[2].SetPixel(x * 2 + 1, y, Color.Black);
                    }

                }

                else
                {
                    image[2].SetPixel(x * 2, y, Color.Red);

                }
            }
        }
        brush.Dispose();
        tempImage.Dispose();
        finalImage.Dispose();

        return image;
    }
```

6.3.3 Merging of captcha

```
string str;
    DataTable dt;
    int RowCount;

    byte[] arr1;
    byte[] arr2;

    string UserName, Password;
    public enum CompareResult
    {
        ciCompareOk,
        ciPixelMismatch,
        ciSizeMismatch
    };
ImageConverter ic = new ImageConverter();
            System.Drawing.Image img = (System.Drawing.Image)ic.ConvertFrom(arr1);
            Bitmap bmp1 = new Bitmap(img);
            System.Drawing.Image img1 = (System.Drawing.Image)ic.ConvertFrom(arr2);
            Bitmap bmp2 = new Bitmap(img1);
            if (Compare(bmp1, bmp2) == CompareResult.ciCompareOk)
            {
ImgOrgCShare.ImageUrl = "~/Split/" + id + "_C.jpg";
                lblresult.Text = "Success!!!";
}
```

Chapter 7

TESTING

7.1 What is software testing?

Software testing is an investigation conducted to provides stakeholder with information about the major quality of the product or services under test. Correctness testing and reliability testing are two major areas of testing. Software testing is trade-off between budget, time and quality.

7.1.1 Goal

The primary goal of testing is to uncover requirement, design or coding errors in the programs. Testing is process of executing a program with the intent of finding an error. A good test case is one that has high probability of finding an as-yet-undiscovered errors. A successful test is one that uncovers an as-yet-undiscovered errors. A component in sense refers to an integrated aggregate of more than one unit in a realistic scenario , many unit are combine into components which are in turns aggregated into even larger part of program. After successful testing of all the basic modules, initially registration modules user search and new mail were combined and tested to check whether they execute properly on the same machine. Finally all the modules were combined into single application and finally integrated modules were tested.

7.2 Test planning

Test planning covers four key activities:

- **The creation of a test strategy that will guide all testing activities.**
- **The creation of test plans for the different stages of testing.**
- **The setting up of the test environment so that the test plan can be carried out.**
- **The creation of test scripts for automated testing.**

All 4 categories use the same source material for designing the test cases, which consists of the following:

- **Business model- This is probably the most important source, as it describes the business activities that the application supports.**
- **Application model- This is also important in that it provides a picture of what the application is supposed to do, and can compensate for gaps in the user documentation.**
- **System model- This is important as a formal and detailed technical source, and is especially useful in relation to non-functional requirements.**
- **Non-functional requirements- This covers the important constraints that the application should satisfy (e.g., performance requirements).**
- **User documentation concept- It is unlikely that by the time system testing commences, the user documentation would be ready. However, it is reasonable to expect that by then concept documents be at least produced, providing a terse version of the intended documentation.**

Table 7.1: Test planning

ID	Purpose	Testing Type	Stage	In Project	
1	To detect any variances between the unit behaviour and its specification	Unit Testing	After a unit has been coded.	After registration phase all the fields entered bythe user are tested. After login phase user name and password are tested.	
2	To detect any discrepancies in the interfaces between the units.	Integration Testing	When a number of units are combined to create an executable module	Registration form and CAPTCHA is created.Spliting of the Captcha is tested.Merging of the CAPTCHA is tested.Storing of CAPTCHA at Client and Serverside is tested.	
3	To detect any variances between the way the application behaves and its official requirements model	System Testing	When an integrated Application is robust enough to undergo extensive testing	Time of submission of registration form is tested. Time of CAPTCHA generation Is tested. Browsing of CAPTCHA time is tested.	
4	Process of giving input to system and checking output of system	Black Box Testing	After completion of entire Project	At time of login half share of CAPTCHA is tested.	
5	It takes into account how system flow and internal Structure of system	White Box Testing	After completion of entire Project	CAPTCHA generation ,splitting and merging are tested.	
6	When a solution is delivered to its intended customer.	Acceptance Testing	When a solution is delivered to its intended customer	Authorized login is tested.	

7.2.1 Test cases

TC ID	OBJECTIVE	PREREQUISITES	STEPS TO BE FOLLOWED	EXPECTED RESULT	ACTUAL RESULT	REMARK
1	Register into website	User must be register	1.User must enter correct details 2.User shold be fill the password in the password field. 3.Password and Confirm password must be same. 4.must click on Submit button.	Granted permission to register successfully.	Granted permission to register successfully.	Pass
2	Register into website	User must be register	1.User Should be enter the incorrect user name. 2.User shold be fill the password in the password field. 3.Click on Submit button.	The Website will give an unauthorized access	Check User name and password to register successful	pass

Continued on next page

Table 7.2 – continued from previous page

TC ID	OBJECTIVE	PREREQUISITES	STEPS TO BE FOLLOWED	EXPECTED RESULT	ACTUAL RESULT	REMARK
3	Register into website	User must register	1.User Should be enter the correct user name. 2.User shold be fill the wrong password in the password field. 3.Click on Submit button.	The website will give an unauthorized access.	Check User name and password to register successful	pass
4	Register into website	User must register	1.User Should be enter the correct user name. 2.User shold be fill the password in the password field. 3.User should fill the incorrect confirm password field. 4.Click on Submit button.	The website will give an unauthorized access.	The website will give an unauthorized access.	pass

Continued on next page

Table 7.2 – continued from previous page

TC ID	OBJECTIVE	PREREQUISITES	STEPS TO BE FOLLOWED	EXPECTED RESULT	ACTUAL RESULT	REMARK
5	Register into website	User must register	1.Captcha is generated 2.Enter the Correct text of captcha 3.Click on register button.	The Website will give an authorized access	The Website will give an authorized access	pass
6	Register into website	User must register	1.Captcha is generated 2.Enter the incorrect text of Captcha 3.Click on register button.	The website will give an unauthorized access.	The website will give an unauthorized access.	pass

Continued on next page

Table 7.2 – continued from previous page

TC ID	OBJECTIVE	PREREQUISITES	STEPS TO BE FOLLOWED	EXPECTED RESULT	ACTUAL RESULT	REMARK
7	**Login to website**	**User must login**	**1.User should enter correct user name.** **2.User should fill the correct password** **3.Browse his own share** **4.Click on Login button.**	**The Website will give an authorized access**	**The Website will give an authorized access**	**pass**
8	**Login to website**	**User must login**	**1.User should enter correct user name.** **2.User should fill the correct password** **3.Browse incorrect share.** **4.Click on Login button.**	**The website will give an unauthorized access.**	**The website will give an unauthorized access.**	**pass**

Continued on next page

Table 7.2 – continued from previous page

TC ID	OBJECTIVE	PREREQUISITES	STEPS TO BE FOLLOWED	EXPECTED RESULT	ACTUAL RESULT	REMARK
9	Login to website	User must login	1.User should enter incorrect user name. 2.User should enter incorrect password. 3.Browse incorrect share. 4.Click on Login button.	The website will give an unauthorized access.	The website will give an unauthorized access.	pass
10	Login to website	User must login	1.User should enter correct user name. 2.User should fill the correct password 3.No Captcha 4.Click on Login button.	The website will give an unauthorized access.	The website will give an unauthorized access.	pass

Continued on next page

Table 7.2 – continued from previous page

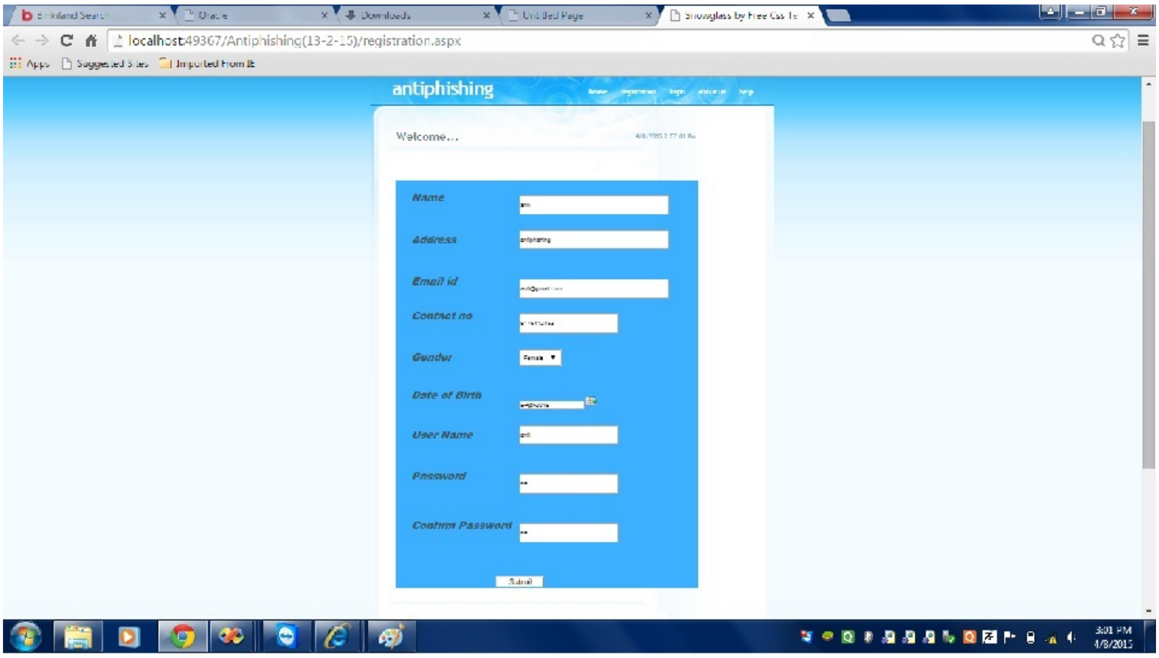
TC ID	OBJECTIVE	PREREQUISITES	STEPS TO BE FOLLOWED	EXPECTED RESULT	ACTUAL RESULT	REMARK
11	Login to website	User must login	1.No username 2.No password 3.No Captcha	The website will give an unauthorized access.	The website will give an unauthorized access.	pass

Table 7.2: Test cases

Chapter 8

SNAPSHOTS

8.1 Registration



localhost:49367/Antiphishing(13-2-15)/registration.aspx

antiphishing

Welcome...

Name	anti
Address	antiphishing
Email id	[illegible]
Contact no	[illegible]
Gender	Female
Date of Birth	[illegible]
User Name	anti
Password	••
Confirm Password	••

Submit

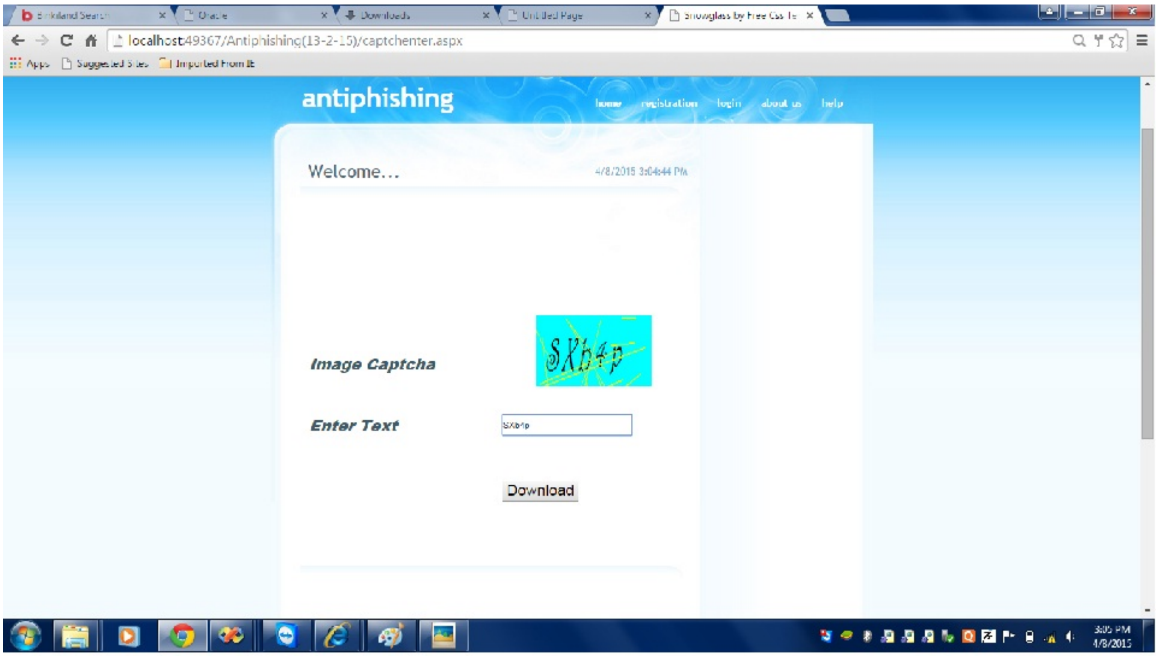
FIGURE 8.1: Registration page

When new user is going to register, he has to fill the following information-

- **Enter name.**
- **Enter address.**
- **Enter email id.**
- **Enter contact no.**

- **Enter gender.**
- **Enter date of birth.**
- **Enter user name.**
- **Enter password.**
- **Re-enter password to confirm password.**
- **Click on submit button.**

8.2 Captcha generation



Welcome...

4/8/2015 3:04:44 PM

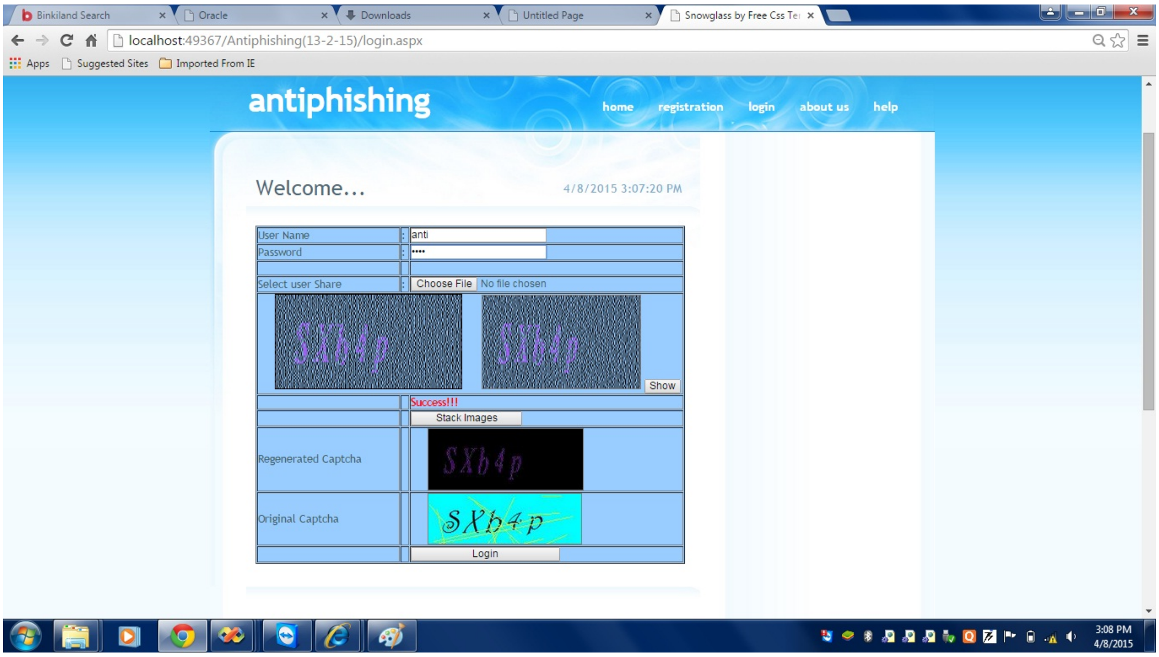
Image Captcha

Enter Text SXb4p

FIGURE 8.2: Captcha generation.

- **Enter the text of captcha.**
- **Click on download button.**

8.3 Login



localhost:49367/Antiphishing(13-2-15)/login.aspx

Welcome...

4/8/2015 3:07:20 PM

User Name : anti

Password : ••••

Select user Share : Choose File No file chosen

Success!!!

Regenerated Captcha

Original Captcha

Figure 8.3: Login page

When user is going to login, he has to fill the following information-

- **Enter user name.**
- **Enter password.**
- **Select user share.**
- **Click on show button.**
- **Click on stack images button.**
- **Click on login button**

8.4 Encrypted image captcha of client and server

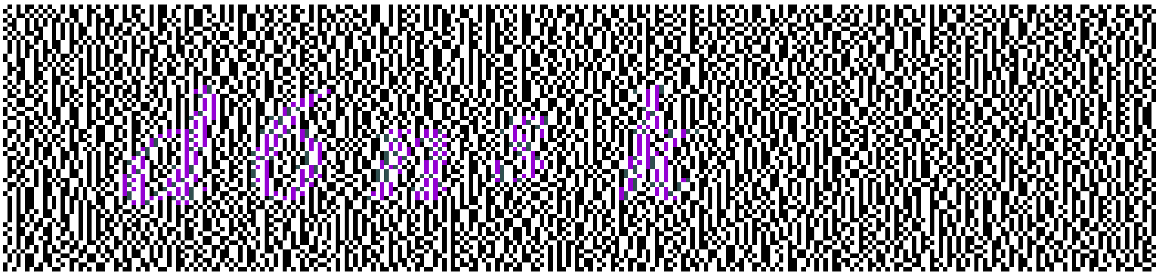


FIGURE 8.4: Encrpted image captcha of client

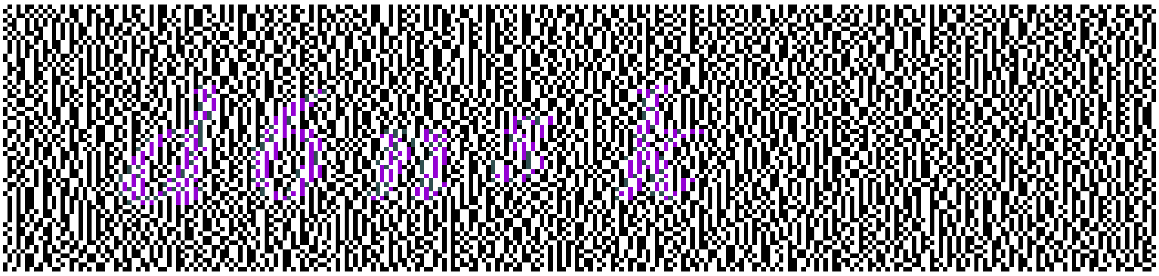
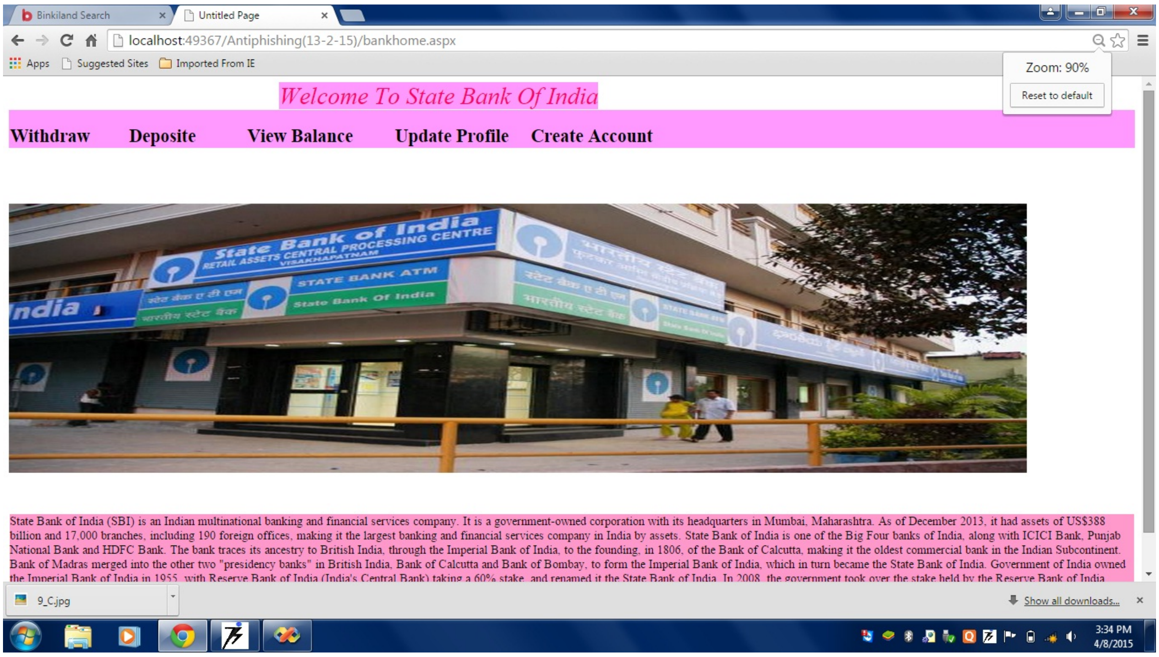


FIGURE 8.5: Encrpted image captcha of server

8.5 User's account



State Bank of India (SBI) is an Indian multinational banking and financial services company. It is a government-owned corporation with its headquarters in Mumbai, Maharashtra. As of December 2013, it had assets of US$388 billion and 17,000 branches, including 190 foreign offices, making it the largest banking and financial services company in India by assets. State Bank of India is one of the Big Four banks of India, along with ICICI Bank, Punjab National Bank and HDFC Bank. The bank traces its ancestry to British India, through the Imperial Bank of India, to the founding, in 1806, of the Bank of Calcutta, making it the oldest commercial bank in the Indian Subcontinent. Bank of Madras merged into the other two "presidency banks" in British India, Bank of Calcutta and Bank of Bombay, to form the Imperial Bank of India, which in turn became the State Bank of India. Government of India owned the Imperial Bank of India in 1955, with Reserve Bank of India (India's Central Bank) taking a 60% stake, and renamed it the State Bank of India. In 2008, the government took over the stake held by the Reserve Bank of India

FIGURE 8.6: User's account window.

t

Chapter 9

CONCLUSION

9.1 Conclusion

Phishing is comman attacks because it can attack globally and store the users confidential information. This information is used by the attackers which are indirectly involved in the phishing process. Phishing websites as well as human users can be easily identified using our project "A Novel on Anti-phishing using Visual Cryptography". This project preserves confidential information of users using 3 layers of security. 1st layer verifies whether the website is a genuine website or a phishing website. If the website is a phishing website then in that situation the phishing website can not display the image captcha for that specific user due to the fact that the image captcha is generated by the stacking of two shares, one with the user and the other with the actual database of the website. Second layer cross validates image Captcha corresponding to the user. The image Captcha is readable by human users alone and not by machine users. Only human users accessing the website can read the image Captcha and ensure that the site as well as the user is permitted one or not. So, using image Captcha technique, no machine based user can crack the password or other confidential information of the users. And as a third layer of security it prevents intruders attacks on the users account. This method provides additional security in terms of not letting the intruder log in into the account even when the user knows the username of a particular user. This project is also useful to prevent the attacks of phishing websites on financial web portal, banking portal, online shopping market.

Bibliography

[1] Divya James and Mintu Philip. A novel anti phishing framework based on visual cryptography. In *Power, Signals, Controls and Computation (EPSCICON), 2012 International Conference on*, pages 1–5. IEEE, 2012.

[2] Qingxiang Feng, Kuo-Kun Tseng, Jeng-Shyang Pan, Peng Cheng, and Charles Chen. New anti-phishing method with two types of passwords in openid system. In *Genetic and Evolutionary Computing (ICGEC), 2011 Fifth International Conference on*, pages 69–72. IEEE, 2011.

[3] Arash Nourian, Sameer Ishtiaq, and Muthucumaru Maheswaran. Castle: A social framework for collaborative anti-phishing databases. In *Collaborative Computing: Networking, Applications and Worksharing, 2009. CollaborateCom 2009. 5th International Conference on*, pages 1–10. IEEE, 2009.

[4] Dorothy Elizabeth Robling Denning. *Information warfare and security*, volume 4. Addison-Wesley Reading, MA, 1999.

[5] Wenyin Liu, Xiaotie Deng, Guanglin Huang, and Anthony Y Fu. An antiphishing strategy based on visual similarity assessment. *Internet Computing, IEEE*, 10(2):58–65, 2006.