Color Scheme Authentication

OPERATIONAL FEASIBILITY

The proposed system was found to feasible in all these different classifications of the study and this led to the design of the system.

The important points to be studied before the design of the system are the software engineering principles such as:

- Rigor and formality
- Separation of concerns
- Modularity
- Abstraction
- Anticipation of change
- Generality
- Incrementality

The above principles were carefully studied and implemented in various phases of the project, which will be discussed in the following sections.

- Rigor and formality: Software Engineering is a creative design activity, but it must be practiced systematically. Rigor is a necessary complement to creativity that increases our confidence in our developments. Formality is rigor at the highest degree. Rigorous documentation of development steps helped project management and assessment of timeliness.
- Separation of concerns: To dominate complexity, separate the issue to concentrate on one at a time. “Divide & conquer” supports parallelization of efforts and separation of responsibilities. Go through phases one after the other(as in water fall model). We did separation of concerns by separating activities with respect to parts. This resulted in two parts of the project which made the task quite simple.
- Modularity: A complex system may be divided into simpler pieces called modules. A system that is composed of modules is called modular. The concepts of cohesion and coupling are to be applied. When dealing with a

module we can ignore details of other modules. There are two modules in this project which handles server and client.

- Abstraction: Identify the important aspects of a phenomenon and ignore its details. It is a special case of separation of concerns. The type of abstraction to apply depends on its purpose. For example: The complexity of authentication process is hidden from the user.
- Anticipation of change: Ability to support software evolution requires anticipating potential future changes. It is the basis for software evolvability. For example: Further versions of encryption algorithm can easily be implemented in the proposed system.
- Generality: The principle of generality is closely related to the principle of anticipation of change. It is important in designing software that is free from unnatural restrictions and limitations. While solving a problem, try to discover if it is an instance of a more general problem whose solution can be reused in other cases. Carefully, balance generality against performance and cost, sometimes a general problem is easier to solve than a special case. Our system is more generalised to all classes of biometric data sets which includes face recognition, iris, hand geometry etc.
- Incrementality: An incremental software development process simplifies verification. If you develop software by adding small increments of functionality then, for verification, you only need to deal with the added portion. If there are any errors detected then they are already partly isolated so they are much easier to correct. A carefully planned incremental development process can also ease the handling of changes in requirements. To do this, the planning must identify use cases that are most likely to be changed and put them towards the end of the development.process.

SYSTEM DESIGN AND ANALYSIS

INPUT DESIGN

The input design is very important for any application. The input design describes how the software communicates within itself, to system that interested with it and with human who use it. The input design is the process of converting the user-oriented inputs into the computer-based format. The data is fed into the system using simple interactive forms. The forms have been supplied with message so that user can enter the data without facing any difficulty. The data is validated wherever it requires in the project.

OUTPUT DESIGN

Outputs are the most important and direct source in information to the consumer and administrator. Intelligent output design will improve the system's relationship with user and help in decision making. It has a conversation panel to display the connection information, remote messages and information for user.

Efficient, intelligent output design should improve the system's relationship with the user and help in decisions making. Since the reports are directing reffered by the management for taking decision and to draw conclusion they must be designed with almost care and the details in the reports must be simple, descriptive and clear to the user. So while designing output the following things are to be considered.

EXISTING SYSTEM

Traditonal method used for authentication is textual password. The volunerabilities of this method like eves dropping,dictionary attack, social enginnering and shoulder suffering are well known. Random and lengthy passwords can make the system secure. But the main problem is the difficulty of remembering those passwords. Studies have shown that users tend to pick short passwords or passwords that are easy to remember. Unfortunately, these passwords can be easily guessed or cracked.

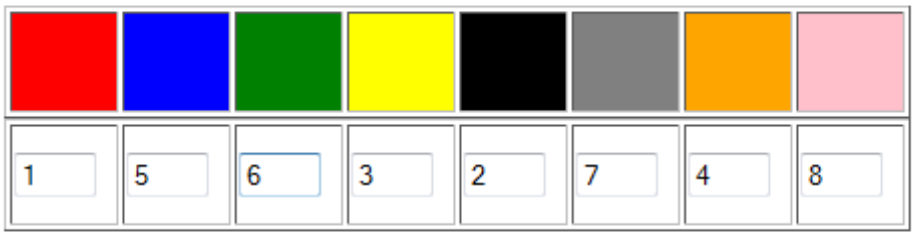
PROPOSED SYSTEM

We propose a new authentication scheme "Color Scheme Authentication ". Instead of just words we propose a system in which authentication is done using colors and numbers. Users can give values from 1 to 8 for the given 8 colors. Usres can even

give same value for two different colors.this makes the authentication method risk free of shoulder attack, dictionary attack,eves dropping etc. That is evolved if the logic every user uses for giving values for colors is simple.

During registration, user should rate colors. The user should rate colors from 1 to 8and he can remember it as “RLYOBGIP”. Same rating can given to different colors. During the login phase, when the user enters his user name an interface is displayed based on the colors selected by the user. The login interphase consists of grid of size 8x8. This contains digit 1-8 placed randomly in grid cells. The interface also contains strips of colors. The color grid consists of 4 pairs of colors. Each pair of color represents the row and the colomn of the grid.

MINI PROJECT 2014



	1	2	3	4	5	6	7	8
1	5	6	3	4	8	1	2	7
2	2	8	6	5	1	3	7	4
3	4	3	7	8	2	5	1	6
4	7	4	1	2	5	6	3	8
5	8	2	4	6	3	7	5	1
6	1	5	8	7	4	2	6	3
7	3	7	5	1	6	4	8	2
8	6	1	2	3	7	8	4	5

The login interface having the color grid and number grid of 8x8 having numbers 1 to 8 randomly placed in the grid. Depending on the ratings given to colors, we get the session passwords. As discussed above, the first color of every pair in color grid represents row and second represents colomn of the number grid. The number in the intersection of the row and colomn of the grid is part of the session password. The first pair has red and yellow colors. The red color rating is 1 and yellow color rating is 3. So the first letter of session password is 1st row and 3rd colomn intersecting element i.e, 3. The same method is followed for other pairs of colors. So the password is “3573”. For every login,both the number grid and the color grid get randomizes so the session password changes for every session.

DATABASE DESIGN

A database design gives us a description about how data is stored and retrieved from storage area. i.e, the database. In our project , database consist of one table. It is used for storing the user details.

Colomn name	Data types	Allow nulls
Uid	Vchar(MAX)	unchecked
Uname	Vchar(MAX)	Unchecked
Secq	Vchar(MAX)	Unchecked
Ans	Vchar(MAX)	Unchecked
Eid	Vchar(MAX)	Unchecked
Phno	Vchar(MAX)	Unchecked
Lack	Vchar(MAX)	Unchecked
Blue	Vchar(MAX)	Unchecked
Red	Vchar(MAX)	Unchecked
Orange	Vchar(MAX)	Unchecked
Yellow	Vchar(MAX)	Unchecked
Green	Vchar(MAX)	Unchecked
Brown	Vchar(MAX)	Unchecked
Violet	Vchar(MAX)	Unchecked

SYSTEM REQUIREMENTS

Hardware Requirements

- Processor : Intel or AMD processor computer
- RAM : 256 MB or more
- Hard Disk space : 8 GB or more

Software Requirements

- Operating System : Windows XP SP3 or above
- Enviornment : Visual Studio.Net 2010
- Language : C#.Net
- Backend : Microsoft SQL Server 2008

SYSTEM SPECIFICATION

FRONT END

C#.Net

C# is an elegant and type safe object oriented language that enables developers to build a variety of secure and robust applications that run on the .Net frame work.You can use c# to create traditional windows client applications,xml webservices,distributed components,clientserver applications,database applications,and much,much more.Visual c# 2010 provides an advanced code editor,convenient user

interface designers,integrated debugger,and many other tools to make it easier to develop applications based on version 4.0 of the c# language and version 4.0 of the .Net framework.

ASP.NET frame work

The .Net languages: these include visual basic, c#, jscript . Net , j# and c++.

The CommonLanguage Runtime (CLR): this is the engine that executes all .Net programs and provides automatic services for these applications,such as security checking,memory management, and optimization.

The .Net frame work class library : the class library collects thousands of pieces of prebuilt functionality that you can “snap in” to your applications.These feature sare sometimes organised into technology sets , such as ADO.NET(the technology for creating database applications)and windows forms(the technology for creating desktop user interfaces).

ASP.NET:this is the engine that hosts the web applications you create with .Net and supports almost any feature from the .Net class library.ASP.NET also includes a set ofwebspecific services,like secure authentication and data storage.

Visual studio :this optional development tool consists of a rich set of productivity and system development life cycle (SDLC)models have been created: waterfall,fountain,spiral,build and fix,rapid prototyping,incremental and synchronized and stabilise.

BACK END

SQL Server 2008

Microsoft SQLserver is a relational database server , developed by Microsoft : it is a software product whose primary function is to store and retrieve data as requested by other software applications ,be it those on the same computer or those running onanother computer across anetwork.There are atleast a dozen different editions of Microsoft SQLserver aimed at different audiences and for different workloads (ranging from small applications that store and retrieve data on the same computer,to millions of users and computers that access hugeamounts of data from the internet at the same time).

SQLserver 2008 was released on august 6,2008 and aims to make data management self turing ,self organizing ,and self maintaining with the development of SQL server alwaya on technologies,to provide near zero down time.SQLsever 2008 also includes support for structured and semi structured data. Microsoft,SQLserver 2008 can be a data storage back end for different varieties of data : xml,email,time/calendar,file,document,spatial etc as well as perform search ,query ,analysis,sharing and synchronization across all data types

IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned into a working system. This is the final and important phase in the life cycle. It is actually a process of converting a new system into an operational one.

Task of implementation

It is a process of bringing a developed system into components which are to be tested ina structured and planned manner. The software should be delivered to the users and they should have confidence that the system work efficiently and effectively,

The more complex the system being implemented the more involved will be the system analysis and design efforts required for implementation.

The system carrying three modules has been implemented with confirmed effectiveness,detection and correction of errors and making necessary all decisions on their true or false side changes so as to satisfy the requirements.

The system is using very few software packages like .Net and it use the back end as SQL.This will very useful to store the information of the user and also easily can transmit the data to the other user securely. This is the main advantage of the system. With the system is implementing the resources can be get in very low cost.

Front End Implementation

We have designed the front end using .Net framework version 4.0. It is user friendly and easy to use. the necessary forms requested for the project could easily be built with this. We have created three forms in the user side. We have drag and drop options for placing necessary buttons on the form. There are many inbuilt packages

visual studio to make the design phase attaractive; we just have to import them to the program code. since for performing all these activities we use visual studio 2010..Net is the suited language for implementing our project.

Module vice Implementation

Mainly we have 2 main modules in our project

- Admin Module
- User Module

Admin Modules

- Encryption
- Authentication
- Randomization

User Modules

- Sign up
- Log in
- Recovery

Admin module implementation

It consists of Encryption, Authentication and Randomization. These are done on the coding part.

Function for encryption is given

```
Public string Encrypt (string plaintext, string key)
    {
        byte [] clearBytes =
System.Text.Encoding.Unicode.GetBytes(plainText);
```

```
        PasswordDeriveBytes pdb = new PasswordDeriveBytes(key, new
byte[] { 0x49, 0x76, 0x61, 0x6e, 0x20, 0x4d, 0x65, 0x64, 0x76, 0x65,
0x64, 0x65, 0x76 });
        byte[] encryptedData = Encrypt(clearBytes, pdb.GetBytes(32),
pdb.GetBytes(16));
        return Convert.ToBase64String(encryptedData);
    }
    Public static byte[] Encrypt(byte[] clearData, byte[] key, byte[]
IV)
    {
        MemoryStream ms = new MemoryStream();
        Rijndael alg = Rijndael.Create();
        alg.Key = key;
        alg.IV = IV;
        CryptoStream cs = new CryptoStream(ms, alg.CreateEncryptor(),
CryptoStreamMode.Write);
        cs.Write(clearData, 0, clearData.Length);
        cs.Close();
        byte[] encryptedData = ms.ToArray();
        return encryptedData;
    }
```

Encryption algorithm

Rijndael's Algorithm

Rijndael (pronounced rain-dahl) is the algorithm that has been selected by the U.S. National Institute of Standards and Technology (NIST) as the candidate for the Advanced Encryption Standard (AES). It was selected from a list of five finalists, that were themselves selected from an original list of more than 15 submissions. Rijndael will begin to supplant the Data Encryption Standard (DES) - and later Triple DES - over the next few years in many cryptography applications. The algorithm was designed by two Belgian cryptologists, Vincent Rijmen and Joan Daemen, whose surnames are reflected in the cipher's name. Rijndael has its origins in Square, an earlier collaboration between the two cryptologists.

The Rijndael algorithm is a new generation symmetric block cipher that supports key sizes of 128, 192 and 256 bits, with data handled in 128-bit blocks - however, in excess of AES design criteria, the block sizes can mirror those of the keys. Rijndael uses a variable number of rounds, depending on key/block sizes, as follows:

9 rounds if the key/block size is 128 bits

11 rounds if the key/block size is 192 bits

13 rounds if the key/block size is 256 bits

Rijndael is a substitution linear transformation cipher, not requiring a Feistel network. It use triple discreet invertible uniform transformations (layers). Specifically, these are: Linear Mix Transform; Non-linear Transform and Key Addition Transform. Even before the first round, a simple key addition layer is performed, which adds to security. Thereafter, there are Nr-1 rounds and then the final round. The transformations form a State when started but before completion of the entire process.

The State can be thought of as an array, structured with 4 rows and the column number being the block length divided by bit length (for example, divided by 32). The cipher key similarly is an array with 4 rows, but the key length divided by 32 to

give the number of columns. The blocks can be interpreted as unidimensional arrays of 4-byte vectors.

The exact transformations occur as follows: the byte subtransformation is nonlinear and operates on each of the State bytes independently - the invertible S-box (substitution table) is made up of 2 transformations. The shiftrow transformation sees the State shifted over variable offsets. The shift offset values are dependent on the block length of the State. The mixcolumn transformation sees the State columns take on polynomial characteristics over a Galois Field values (28), multiplied x4 + 1 (modulo) with a fixed polynomial. Finally, the roundkey transform is XORed to the State. The key schedule helps the cipher key determine the round keys through key expansion and round selection.

Overall, the structure of Rijndael displays a high degree of modular design, which should make modification to counter any attack developed in the future much simpler than with past algorithm designs.

High-level description of the algorithm

1.KeyExpansion—round keys are derived from the cipher key using Rijndael's key schedule

2.Initial Round

1)AddRoundKey—each byte of the state is combined with the round key using bitwise xor

3.Rounds

1)SubBytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.

2) ShiftRows—a transposition step where each row of the state is shifted cyclically a certain number of steps.

3)MixColumns—a mixing operation which operates on the columns of the state, combining the four bytes in each column.

4)AddRoundKey

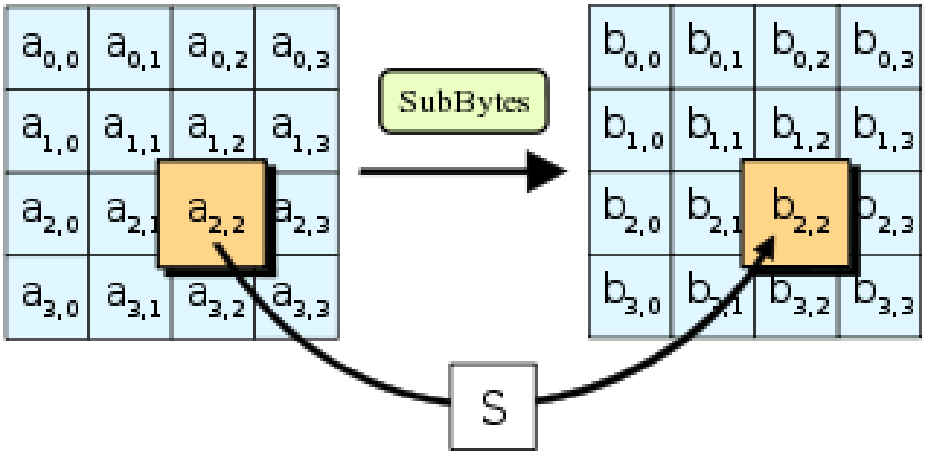
4.Final Round (no MixColumns)

1) SubBytes

2)ShiftRows

3)AddRoundKey

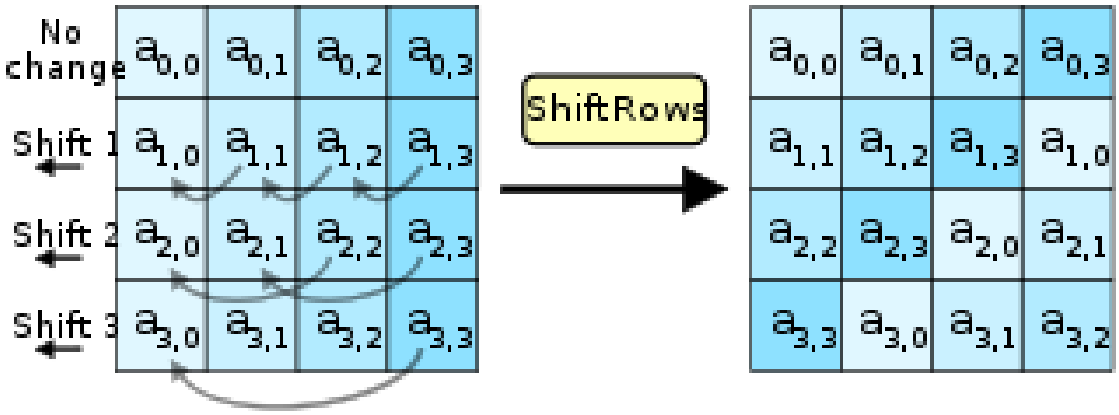
The SubBytes step



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

In the SubBytes step, each byte in the matrix is updated using an 8-bit substitution box, the Rijndael S-box. This operation provides the non-linearity in the cipher. The S-box used is derived from the multiplicative inverse over **GF**(2^8), known to have good non-linearity properties. To avoid attacks based on simple algebraic properties, the S-box is constructed by combining the inverse function with an invertible affine transformation. The S-box is also chosen to avoid any fixed points (and so is a derangement), and also any opposite fixed points.

The Shift Rows step

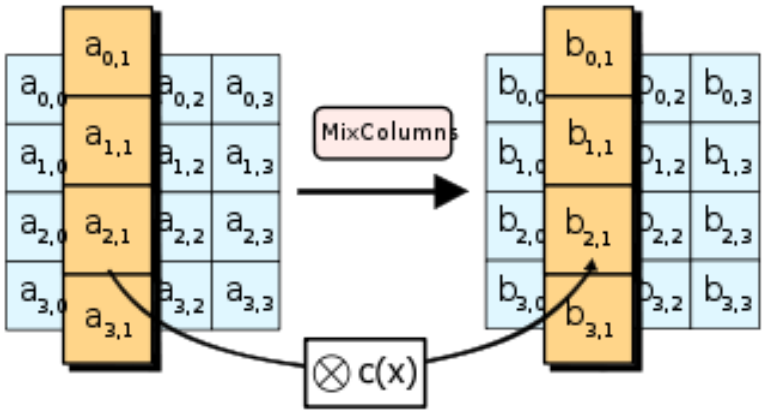


In the ShiftRows step, bytes in each row of the state are shifted cyclically to the left. The number of places each byte is shifted differs for each row.

The ShiftRows step operates on the rows of the state; it cyclically shifts the bytes in each row by a certain offset. For AES, the first row is left unchanged. Each byte of the second row is shifted one to the left. Similarly, the third and fourth rows are shifted by offsets of two and three respectively. For the block of size 128 bits and 192 bits the shifting pattern is the same. In this way, each column of the output state of the ShiftRows step is composed of bytes from each column of the input state. (Rijndael variants with a larger

block size have slightly different offsets). In the case of the 256-bit block, the first row is unchanged and the shifting for second, third and fourth row is 1 byte, 3 bytes and 4 bytes respectively—this change only applies for the Rijndael cipher when used with a 256-bit block, as AES does not use 256-bit blocks.

The MixColumns step



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

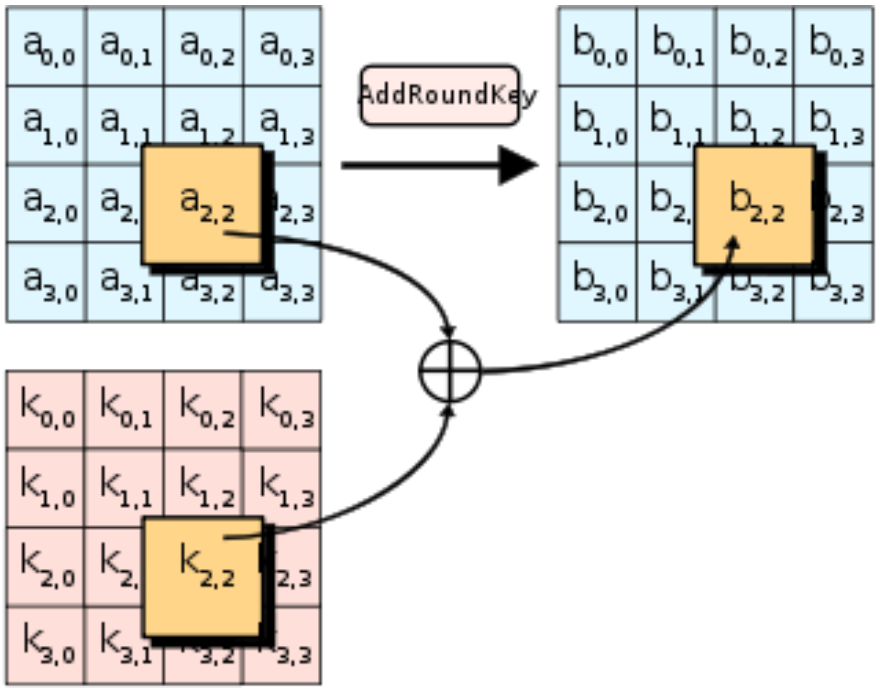
In the MixColumns step, the four bytes of each column of the state are combined using an invertible linear transformation. The MixColumns function takes four bytes as input and outputs four bytes, where each input byte affects all four output bytes. Together with ShiftRows, MixColumns provides diffusion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

During this operation, each column is multiplied by the known matrix that for the 128 bit key.

The multiplication operation is defined as: multiplication by 1 means leaving unchanged, multiplication by 2 means shifting byte to the left and multiplication by 3 means shifting to the left and then performing xor with the initial unshifted value. After shifting, a conditional xor with 0x1B should be performed if the shifted value is larger than 0xFF.

In more general sense, each column is treated as a polynomial over GF(28) and is then multiplied modulo x4+1 with a fixed polynomial c(x) = 0x03 • x3 + x2 + x + 0x02. The coefficients are displayed in their hexadecimal equivalent of the binary representation of bit polynomials from GF(2)[x]. The MixColumns step can also be viewed as a multiplication by a particular MDS matrix in a finite field. This process is described further in the article Rijndael mix columns.

The AddRoundKey step



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

User module implementation

User module consists of 3 modules

- Sign up
- Log in
- Recovery

Sign up

Sign up module is the main part of the project. This module is used to sign up color scheme authentication. The signup includes username, email id, security question and answer along with password. Unless user don't sign up user cannot login.

Login

The login module is nothing but the processes of logging of a signed up user using color scheme authentication. The user has to login using color scheme authentication to get to his account.

Recovery

The Recovery module does the recovery of the user account in case user forgets the password. This is done by using a custom security question and answer. Once the user passes this process he will be redirect to page from which the user can get back access to his account.

TESTING

Testing is the vital to the success of the system.System itesting makes a logical assumption that if all the part of the system are correct,the goal will be successfully

achieved.System testing is the stage of implementation that we aimed at assuring that the system works accurately and efficiently before live operation commences.

Software testing is a critical element of software quqlity assurance represents the ultimate review of specification,design and coding.The user tests the developed system and changes are made according to there needs.The testing phase involves the testing of developed system using various kinds of data.

Testing Objectives

- Testing is the process of executing the program with the intention of finding an error
- A good test is one that has high probability of finding an as yet undiscovered
- A successful test is that which uncovers as-yet-undiscovered error

White box testing

White box testing foccuses on the program control structure.In our project when user enters his details,thesystem will check the database and if it is valid then the whole lines are from.And also when the message to the user the whole lines are reading from the text area and it will display in the same format in the receiver side.These are all indicates that the control structures work efficiently. If there is any error it will display the null pointer excepion in the console.

Black box testing

Black box testing is a method of software testing that tests the functionality of application as opposed to its internal structures of workings(see white box testing).Specific knowledge of application’s code/internal structure and programming knowledge in general is not required.The tester is only aware of what the software is supposed to do,but not how i.e.When he enters a certain input,he gets a crtain output; without being aware of how the output was produced in the first place.Test cases are

built around specifications and requirements,i.e.what the application is supposed to do.It uses external descriptions of the software,including specifications,requirements,and design to derive test cases.These tests can be functional or non-functional,though usually functional.The designers selects valid and invalid inputs and determines the correct output.There is no knowledge of the test object's internal structure.

Unit testing

Unit testing is carried out to screen wise,each screen being identified as an object.Attention is diverted to individual modules, independently to one another to locate errors in coding and logic

In unit testing

- Module is tested to ensure that information properly flows into and out of the program uder test.
- Local data structures are examined to ensure that data stored temporarily maintains its integrity during all steps in algorithm execution
- Boundary condition is tested to ensure that module operates properly at boundaries established to limit or restrict processing.
- All independent paths through the control structures are executed to ensure that all atatements in the module have been executed at least once.
- Error handling paths are also tested.this tests focuses verification effort on the smallest unit of software design modules.

Here the module interfacs boundary conditions and all indipendent paths were verified by inputting false data.Each single operation is tested individually for its correct functionality.

Integration testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors asssociated with interfacing.

Unit tested module were taken and single program structure was built that has been dictated by the design.Incremental integration has adpoted here.The entire software was developed and tested in small segments; where errors were easily to locate and rectify.Each database or table manipulation operation were separately tested and combined to form a single program.After integation,the single program was tested again with numerous test data to check for its functionality.

The integration can be performed in two ways Top Down integration and Bottom Up integration

Alpha testing

A series of acceptance tests were conducted to enable the employees of the firm to validate requirements.The end user conducted it.The suggestions,along with the additional requirements of the end user were included in the project.

Beta testing

It is to be conducted by the end-user without the presence of the developer.It can be conducted over a period of weeks or month.Since it is a long time consuming activity,it result is out of scope of this project report.But its result will help to enhance the product at later time.

Final testing

When a system is developed, it is hoped that it performs properly.In practice, however , some errors always occur.The main purpose of testing an information system is to find the errors and correct them.A successful test is one, which finds an error.

APPENDIX A

USER MANUAL

1. Install .net and SQL Server.
2. Create database and tables on SQL Server.
3. Run the program code on the visual studio.
4. Create a new account as shown in figure.
5. A registered user can directly login by typing his/her username and password as shown in figure.
6. If verification is success then, he/she can enter.
7. If the user lost his password they can retrieve it from the password recovery by using security question.

APPENDIX B

SCREEN SHOTS

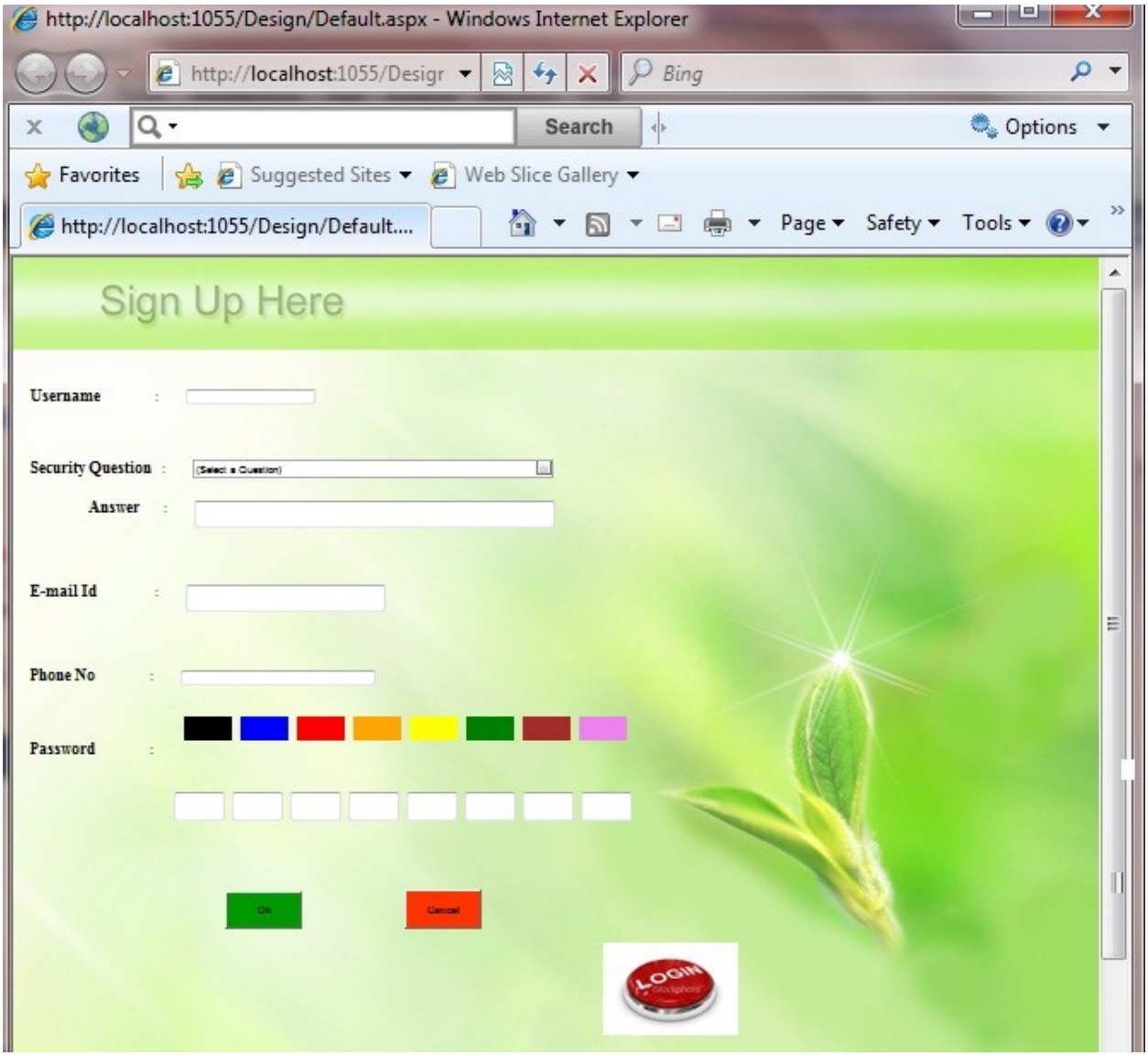

http://localhost:1055/Design/Default.aspx - Windows Internet Explorer
http://localhost:1055/Desigr
Bing
Search
Options
Favorites
Suggested Sites
Web Slice Gallery
http://localhost:1055/Design/Default....
Page
Safety
Tools
Sign Up Here
Username :
Security Question :
(Select a Question)
Answer :
E-mail Id :
Phone No :
Password :
Ok
Cancel
LOGIN

Fig 1 Sign up

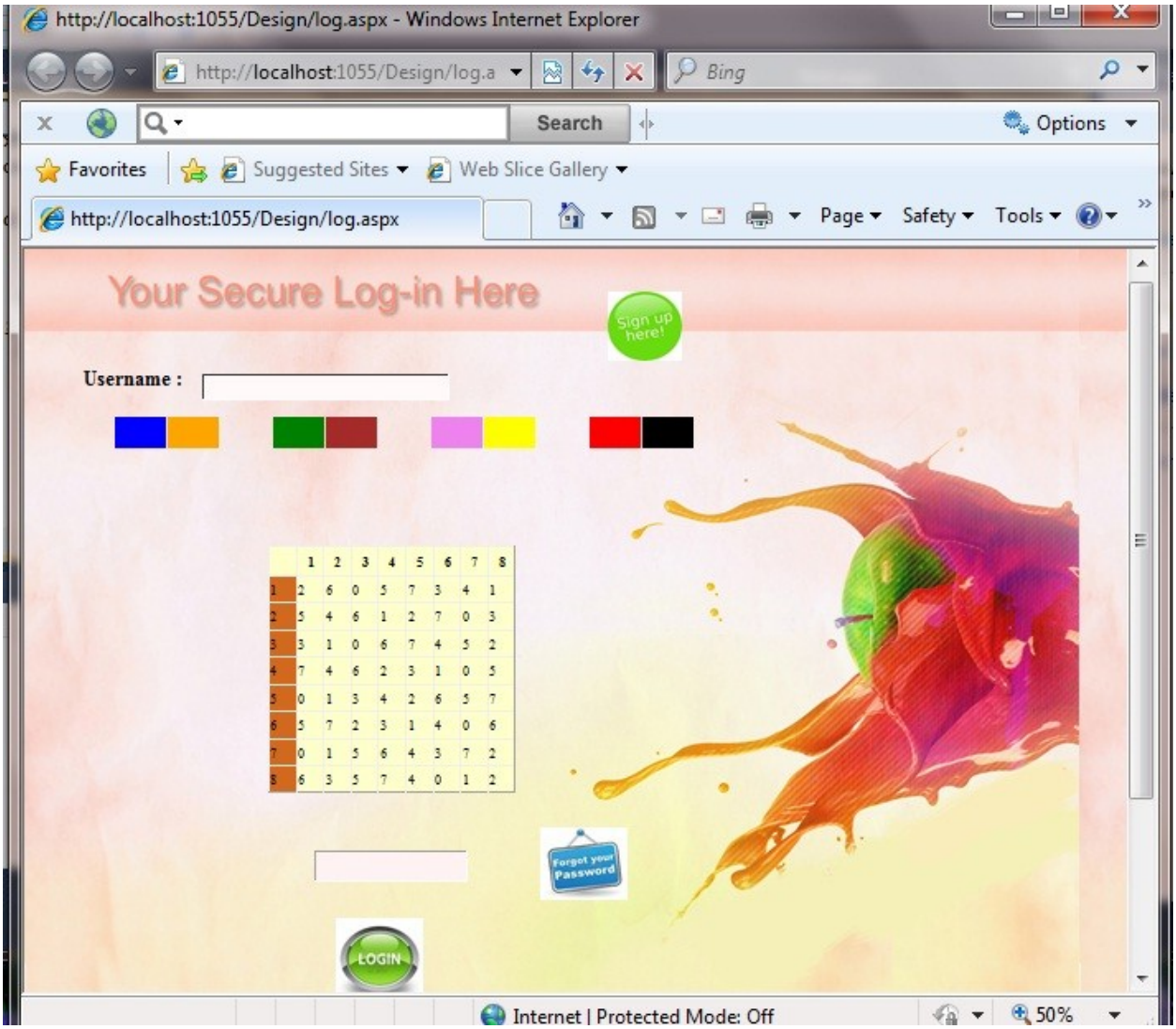

http://localhost:1055/Design/log.aspx - Windows Internet Explorer
http://localhost:1055/Design/log.a
Bing
Search
Options
Favorites
Suggested Sites
Web Slice Gallery
http://localhost:1055/Design/log.aspx
Page
Safety
Tools
Your Secure Log-in Here
Sign up here!
Username :
Forget your Password
LOGIN
Internet | Protected Mode: Off
50%

Fig 2 Login

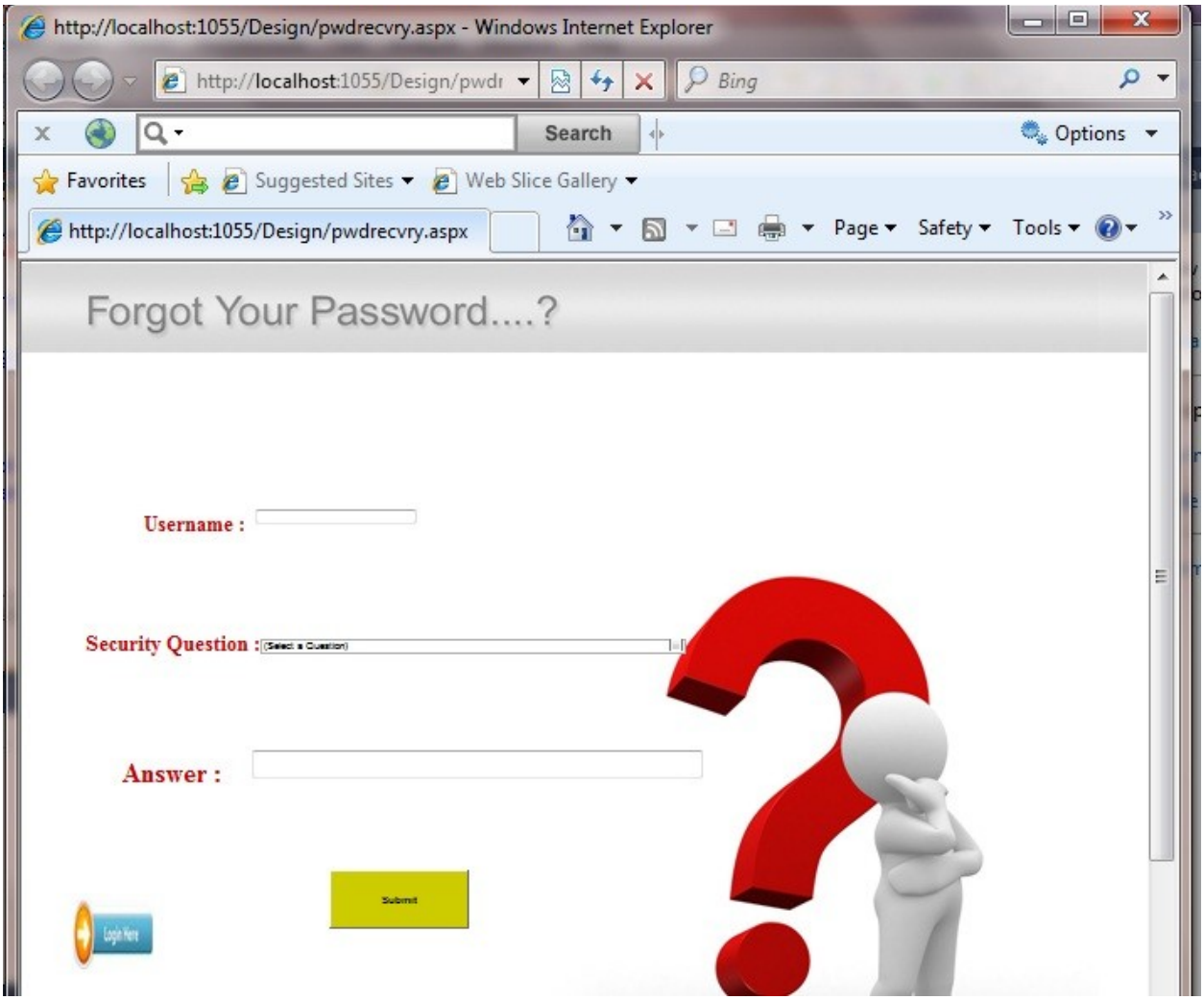


Fig 3 Password Recovery

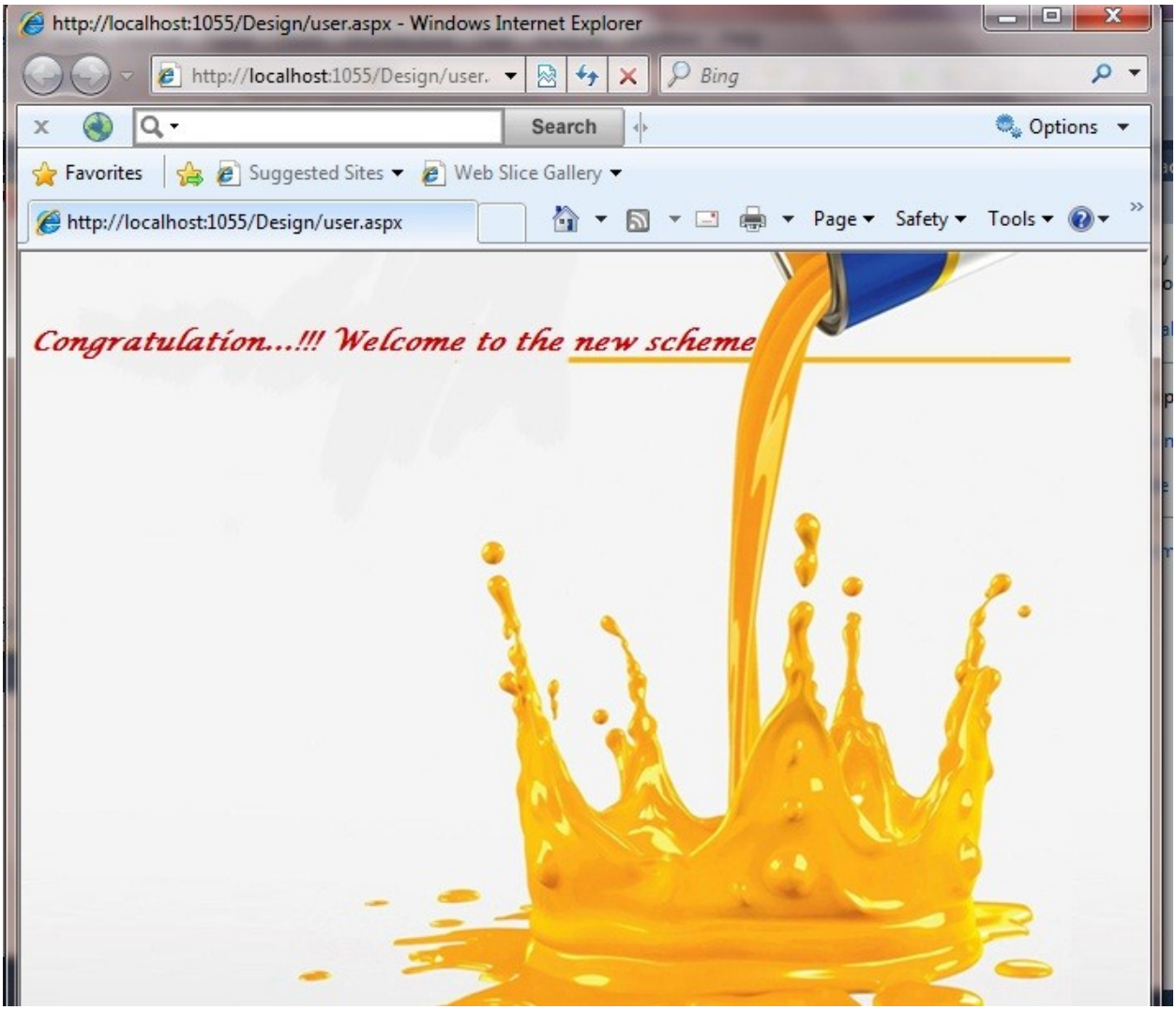


Fig 4 Default