1. **INTRODUCTION**

There has been a great deal of hype for graphical passwords since two decade due to the fact that Primitive’s methods suffered from an innumerable number of attacks which could be imposed easily. Here we will progress down the taxonomy of authentication methods.

To start with we focus on the most common computer authentication method that makes use of text passwords. Despite the vulnerabilities, it’s the user natural tendency of the users that they will always prefer to go for short passwords for ease of remembrance and also lack of awareness about how attackers tend to attacks.

Unfortunately, these passwords are broken mercilessly by intruders by several simple means such as masquerading, Eaves dropping and other rude means say dictionary attacks, shoulder surfing attacks, social engineering attacks. To mitigate the problems with traditional methods, advanced methods have been proposed using graphical as passwords.

The idea of graphical passwords was first described by Greg Blonder (1996). For Blonder, graphical passwords have a predetermined image that the sequence and the tap regions selected are interpreted as the graphical password. Since then, many other graphical password schemes have been proposed.

The desirable quality associated with graphical passwords is that psychologically humans can remember graphical far better than text and hence is the best alternative being proposed. There is a rapid and growing interest in graphical passwords for they are more or infinite in numbers thus providing more resistance.

The major goal of this work is to reduce the guessing attacks as well as encouraging users to select more random, and difficult passwords to guess.

1.1 EXISTING SYSTEM:

Existing approaches to Users often create memorable passwords that are easy for attackers to guess, but strong system-assigned passwords are difficult for users to remember.

Despite the vulnerabilities, it’s the user natural tendency of the users that they will always prefer to go for short passwords for ease of remembrance and also lack of awareness about how attackers tend to attacks.

Unfortunately, these passwords are broken mercilessly by intruders by several simple means such as masquerading, Eaves dropping and other rude means say dictionary attacks, shoulder surfing attacks, social engineering attacks.

Disadvantages:

1. The strong system-assigned passwords are difficult for users to remember.
2. Easily remembered passwords are easy to guess for the intruders or hackers.

1.2 PROPOSED SYSTEM:

We propose is to reduce the guessing attacks as well as encouraging users to select more random, and difficult passwords to guess. The proposed system work merges persuasive cued click points and password guessing resistant protocol.

Advantages:

1. Human brain is good in remembering picture than textual character.
2. Stronger security is possible with this system.
3. It is of great use in systems where security is a major concern and not time

**1.2.1 SYSTEM SPECIFICATION**

***Hardware Requirements:***

* System : Pentium IV 2.4 GHz.
* Hard Disk : 40 GB.
* Floppy Drive : 1.44 Mb.
* Monitor : 14’ Colour Monitor.
* Mouse : Optical Mouse.
* Ram : 512 Mb.
* Keyboard : 101 Keyboard.

**1.2.1.2 *Software Requirements:***

* Operating system : Windows XP.
* Coding Language : ASP.Net with C#
* Data Base : SQL Server 2005.

1.3 MODULES:

1. Pass Points Module.
2. Cued Click Points Module.
3. Persuasive Cued Click- Points Module.

*1.Pass Points Module:*

Based on Blonder’s original idea, Pass Points (PP) is a click-based graphical password system where a password consists of an ordered sequence of five click-points on a pixel-based image. To log in, a user must click within some system-defined tolerance region for each click-point. The image acts as a cue to help users remember their password click-points.

*2.Cued Click Points Module:*

Cued Click Points (CCP) was developed as an alternative click based graphical password scheme where users select one point per image for five images. The interface displays only one image at a time; the image is replaced by the next image as soon as a user selects a click point. The system determines the next image to display based on the user’s click-point on the current image. The next image displayed to users is based on a deterministic function of the point which is currently selected. It now presents a one to-one cued recall scenario where each image triggers the user’s memory of the one click-point on that image. Secondly, if a user enters an incorrect click-point during login, the next image displayed will also be incorrect. Legitimate users who see an unrecognized image know that they made an error with their previous click-point. Conversely, this implicit feedback is not helpful to an attacker who does not know the expected sequence of images.

*3.Persuasive Cued Click- Points Module:*

To address the issue of hotspots, Persuasive Cued Click Points (PCCP) was proposed. As with CCP, a password consists of five click points, one on each of five images. During password creation, most of the image is dimmed except for a small view port area that is randomly positioned on the image. Users must select a click-point within the view port. If they are unable or unwilling to select a point in the current view port, they may press the Shuffle button to randomly reposition the view port. The view port guides users to select more random passwords that are less likely to include hotspots. A user who is determined to reach a certain click-point may still shuffle until the view port moves to the specific location, but this is a time consuming and more tedious process.

1.4 Scope

* Allows user to select random and difficult passwords.
* Graphical passwords are generated based on the clicks given by user.
* Different techniques like pass points, cued click points, persuasive cued click points are used.
* Allowing users to upload their own image.
* Allowing users to select number of images in cued click points technique.
* Different images are processed and passwords are generated.
* Well known security threats like brute force attacks and dictionary attacks can be successfully abolished using this method.
* Sometimes it becomes difficult for the user to remember the randomly generated password.
* Allowing users to recover their password.
* Sending login details to the email id of the respective user.

2 .TECHNOLOGIES USED

2.1 Features of. Net

Microsoft .NET is a set of Microsoft software technologies for rapidly building and integrating XML Web services, Microsoft Windows-based applications, and Web solutions. The .NET Framework is a language-neutral platform for writing programs that can easily and securely interoperate. There’s no language barrier with .NET: there are numerous languages available to the developer including Managed C++, C#, Visual Basic and Java Script. The .NET framework provides the foundation for components to interact seamlessly, whether locally or remotely on different platforms. It standardizes common data types and communications protocols so that components created in different languages can easily interoperate.

“.NET” is also the collective name given to various software components built upon the .NET platform. These will be both products (Visual Studio.NET and Windows.NET Server, for instance) and services (like Passport, .NET My Services, and so on).

2.2 .NET FRAMEWORK

The .NET Framework has two main parts:

1. The Common Language Runtime (CLR).

2. A hierarchical set of class libraries.

The CLR is described as the “execution engine” of .NET. It provides the environment within which programs run. The most important features are

* Conversion from a low-level assembler-style language, called Intermediate Language (IL), into code native to the platform being executed on.
* Memory management, notably including garbage collection.
* Checking and enforcing security restrictions on the running code.
* Loading and executing programs, with version control and other such features. The following features of the .NET framework are also worth description:

*Managed Code*

The code that targets .NET, and which contains certain extra information - “metadata” - to describe itself. While both managed and unmanaged code can run in the runtime, only managed code contains the information that allows the CLR to guarantee, for instance, safe execution and interoperability.

*Managed Data*

With Managed Code comes Managed Data. CLR provides memory allocation and Deal location facilities, and garbage collection. Some .NET languages use Managed Data by default, such as C#, Visual Basic.NET and JScript.NET, whereas others, namely C++, do not. Targeting CLR can, depending on the language you’re using, impose certain constraints on the features available. As with managed and unmanaged code, one can have both managed and unmanaged data in .NET applications - data that doesn’t get garbage collected but instead is looked after by unmanaged code.

*Common Type System*

The CLR uses something called the Common Type System (CTS) to strictly enforce type-safety. This ensures that all classes are compatible with each other, by describing types in a common way. CTS define how types work within the runtime, which enables types in one language to interoperate with types in another language, including cross-language exception handling. As well as ensuring that types are only used in appropriate ways, the runtime also ensures that code doesn’t attempt to access memory that hasn’t been allocated to it.

*Common Language Specification*

The CLR provides built-in support for language interoperability. To ensure that you can develop managed code that can be fully used by developers using any programming language, a set of language features and rules for using them called the Common Language Specification (CLS) has been defined. Components that follow these rules and expose only CLS features are considered CLS-compliant.

*The Class Library*

.NET provides a single-rooted hierarchy of classes, containing over 7000 types. The root of the namespace is called System; this contains basic types like Byte, Double, Boolean, and String, as well as Object. All objects derive from System. Object. As well as objects, there are value types. Value types can be allocated on the stack, which can provide useful flexibility. There are also efficient means of converting value types to object types if and when necessary.

The set of classes is pretty comprehensive, providing collections, file, screen, and network I/O, threading, and so on, as well as XML and database connectivity.

The class library is subdivided into a number of sets (or namespaces), each providing distinct areas of functionality, with dependencies between the namespaces kept to a minimum.

*Languages supported by .Net*

The multi-language capability of the .NET Framework and Visual Studio .NET enables developers to use their existing programming skills to build all types of applications and XML Web services. The .NET framework supports new versions of Microsoft’s old favorites Visual Basic and C++ (as VB.NET and Managed C++), but there are also a number of new additions to the family. Visual Basic .NET has been updated to include many new and improved language features that make it a powerful object-oriented programming language. These features include inheritance, interfaces, and overloading, among others. Visual Basic also now supports structured exception handling, custom attributes and also supports multi-threading. Visual Basic .NET is also CLS compliant, which means that any CLS-compliant language can use the classes, objects, and components you create in Visual Basic .NET. Managed Extensions for C++ and attributed programming are just some of the enhancements made to the C++ language. Managed Extensions simplify the task of migrating existing C++ applications to the new .NET Framework.

C# is Microsoft’s new language. It’s a C-style language that is essentially “C++ for Rapid Application Development”. Unlike other languages, its specification is just the grammar of the language. It has no standard library of its own, and instead has been designed with the intention of using the .NET libraries as its own.

Microsoft Visual J# .NET provides the easiest transition for Java-language developers into the world of XML Web Services and dramatically improves the interoperability of Java-language programs with existing software written in a variety of other programming languages. Active State has created Visual Perl and Visual Python, which enable .NET-aware applications to be built in either Perl or Python. Both products can be integrated into the Visual Studio .NET environment. Visual Perl includes support for Active State’s Perl Dev Kit. Other languages for which .NET compilers are available include

* FORTRAN
* COBOL
* Eiffel
  1. .Net Framework diagram

|  |  |
| --- | --- |
| ASP.NET  XML WEB SERVICES | Windows Forms |
| Base Class Libraries | |
| Common Language Runtime | |
| Operating System | |

C#.NET is also compliant with CLS (Common Language Specification) and supports structured exception handling. CLS is set of rules and constructs that are supported by the CLR (Common Language Runtime). CLR is the runtime environment provided by the .NET Framework; it manages the execution of the code and also makes the development process easier by providing services.

C#.NET is a CLS-compliant language. Any objects, classes, or components that created in C#.NET can be used in any other CLS-compliant language. In addition, we can use objects, classes, and components created in other CLS-compliant languages in C#.NET .The use of CLS ensures complete interoperability among applications, regardless of the languages used to create the application.

*Constructors and Destructors*

Constructors are used to initialize objects, whereas destructors are used to destroy them. In other words, destructors are used to release the resources allocated to the object. In C#.NET the sub finalize procedure is available. The sub finalize procedure is used to complete the tasks that must be performed when an object is destroyed. The sub finalize procedure is called automatically when an object is destroyed. In addition, the sub finalize procedure can be called only from the class it belongs to or from derived classes.

*Garbage Collection*

Garbage Collection is another new feature in C#.NET. The .NET Framework monitors allocated resources, such as objects and variables. In addition, the .NET Framework automatically releases memory for reuse by destroying objects that are no longer in use. In C#.NET, the garbage collector checks for the objects that are not currently in use by applications. When the garbage collector comes across an object that is marked for garbage collection, it releases the memory occupied by the object.

*Overloading*

Overloading is another feature in C#. Overloading enables us to define multiple procedures with the same name, where each procedure has a different set of arguments. Besides using overloading for procedures, we can use it for constructors and properties in a class.

*Multithreading*

C#.NET also supports multithreading. An application that supports multithreading can handle multiple tasks simultaneously, we can use multithreading to decrease the time taken by an application to respond to user interaction.

*Structured exception handling*

C#.NET supports structured handling, which enables us to detect and remove errors at runtime. In C#.NET, we need to use Try…Catch…Finally statements to create exception handlers. Using Try…Catch…Finally statements, we can create robust and effective exception handlers to improve the performance of our application.

*Objectives of .Net framework*

1. To provide a consistent object-oriented programming environment whether object codes is stored and executed locally on Internet-distributed, or executed remotely.

2. To provide a code-execution environment to minimizes software deployment and guarantees safe execution of code.

3. Eliminates the performance problems.

There are different types of application, such as Windows-based applications and Web-based applications.

2.4 ASP.NET OVERVIEW

ASP.NET is a [server-side](https://en.wikipedia.org/wiki/Server-side_scripting) [Web application framework](https://en.wikipedia.org/wiki/Web_application_framework) designed for [Web development](https://en.wikipedia.org/wiki/Web_development) to produce [dynamic Web pages](https://en.wikipedia.org/wiki/Dynamic_Web_pages). It was developed by [Microsoft](https://en.wikipedia.org/wiki/Microsoft) to allow [programmers](https://en.wikipedia.org/wiki/Programmer) to build dynamic [web sites](https://en.wikipedia.org/wiki/Web_site), [web applications](https://en.wikipedia.org/wiki/Web_application) and [web services](https://en.wikipedia.org/wiki/Web_service). It was first released in January 2002 with version 1.0 of the [.NET Framework](https://en.wikipedia.org/wiki/.NET_Framework), and is the successor to Microsoft's [Active Server Pages](https://en.wikipedia.org/wiki/Active_Server_Pages) (ASP) technology. ASP.NET is built on the [Common Language Runtime](https://en.wikipedia.org/wiki/Common_Language_Runtime)(CLR), allowing programmers to write ASP.NET code using any supported [.NET language](https://en.wikipedia.org/wiki/List_of_CLI_languages). User controls are encapsulations of sections of pages which are registered and used as controls in ASP.NET.

Programmers can also build custom controls for ASP.NET applications. Unlike user controls, these controls do not have an ASCX markup file, having all their code compiled into a [dynamic link library (DLL)](https://en.wikipedia.org/wiki/Dynamic-link_library) file. Such custom controls can be used across multiple Web applications and [Visual Studio](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio) projects.

*Session state*

Server-side Session state is held by a collection of user-defined session variables that are persistent during a user session. These variables, accessed using the Session collection, are unique to each session instance. The variables can be set to be automatically destroyed after a defined time of inactivity even if the session does not end. Client-side user session is maintained by either a [cookie](https://en.wikipedia.org/wiki/HTTP_cookie) or by encoding the session ID in the URL itself.

ASP.NET supports three modes of persistence for server-side session variables:

*In-Process Mode*

The session variables are maintained within the ASP.NET [process](https://en.wikipedia.org/wiki/Process_(computing)). This is the fastest way; however, in this mode the variables are destroyed when the ASP.NET process is recycled or shut down.

*ASP State Mode*

ASP.NET runs a separate [Windows service](https://en.wikipedia.org/wiki/Windows_service) that maintains the state variables. Because state management happens outside the ASP.NET process, and because the ASP.NET engine accesses data using .NET Remoting, ASP State is slower than In-Process. This mode allows an ASP.NET application to be load-balanced and scaled across multiple servers. Because the state management service runs independently of ASP.NET, the session variables can persist across ASP.NET process shutdowns. However, since session state server runs as one instance, it is still one point of failure for session state. The session-state service cannot be load-balanced, and there are restrictions on types that can be stored in a session variable.

*SQL Server Mode*

State variables are stored in a [database](https://en.wikipedia.org/wiki/Database), allowing session variables to be persisted across ASP.NET process shutdowns. The main advantage of this mode is that it allows the application to balance load on a server cluster, sharing sessions between servers. This is the slowest method of session state management in ASP.NET.

*Directory structure*

In general, the ASP.NET directory structure can be determined by the developer's preferences. Apart from a few reserved directory names, the site can span any number of directories. The structure is typically reflected directly in the URLs. Although ASP.NET provides means for intercepting the request at any point during processing, the developer is not forced to funnel requests through a central application or front controller.

The special directory names (from ASP.NET 2.0 on) are:

*App\_Code*

This is the "raw code" directory. The ASP.NET server automatically compiles files (and subdirectories) in this folder into an assembly which is accessible in the code of every page of the site. App\_Code will typically be used for data access abstraction code, model code and business code. Also any site-specific http handlers and modules and Web service implementation go in this directory. As an alternative to using App\_Code the developer may opt to provide a separate assembly with precompiled code.

*App\_Data*

The App\_Data ASP.NET Directory is the default directory for any [database](https://en.wikipedia.org/wiki/Database) used by the ASP.NET Website. These databases might include Access (mdb) files or [SQL Server](https://en.wikipedia.org/wiki/Microsoft_SQL_Server) (mdf) files. The App\_Data is the only directory with Write Access enabled for the ASP.NET web application.:[[14]](https://en.wikipedia.org/wiki/ASP.NET" \l "cite_note-14)

*App\_LocalResources*

E.g. a file called CheckOut.aspx.fr-FR.resx holds localized resources for the French version of the CheckOut.aspx page. When the UI culture is set to French, ASP.NET will automatically find and use this file for localization.

*App\_GlobalResources*

Holds resx files with localized resources available to every page of the site. This is where the ASP.NET developer will typically store localized messages etc. which are used on more than one page.

*App\_Themes*

Adds a folder that holds files related to themes which is a new ASP.NET feature that helps ensure a consistent appearance throughout a Web site and makes it easier to change the Web site’s appearance when necessary.

*App\_WebReferences*

holds discovery files and [WSDL](https://en.wikipedia.org/wiki/Web_Services_Description_Language) files for references to [Web services](https://en.wikipedia.org/wiki/Web_service) to be consumed in the site.

*Bin*

Contains compiled code (.dll files) for controls, components, or other code that you want to reference in your application. Any classes represented by code in the Bin folder are automatically referenced in your application.

#### 2.5 ADO.NET OVERVIEW

ADO.NET is an evolution of the ADO data access model that directly addresses customer requirements for developing scalable applications. It was designed specifically for the web with scalability, statelessness, and XML in mind. ADO.NET uses some ADO objects, such as the Connection and Command objects, and also introduces new objects. Key new ADO.NET objects include the Dataset, Data Reader, and Data Adapter.

The role of the managed provider is to connect, fill, and persist the Dataset to and from data stores. The OLE DB and SQL Server .NET Data Providers (System.Data.OleDb and System.Data.SqlClient) that are part of the .Net Framework provide four basic objects: the Command, Connection, DataReader and DataAdapter.

* Connections- For connection to and managing transactions against a database.
* Commands- For issuing SQL commands against a database.
* DataReaders- For reading a forward-only stream of data records from a SQL Server data source.
* DataSets- For storing, remoting and programming against flat data, XML data and relational data.
* DataAdapters. For pushing data into a DataSet, and reconciling data against a database.

2.6 FEATURES OF SQL-SERVER 2008

Microsoft SQL Server is a [relational database management system](http://en.wikipedia.org/wiki/Relational_database_management_system) developed by [Microsoft](http://en.wikipedia.org/wiki/Microsoft). As a database, 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 on another computer across a network (including the Internet). Its primary [query languages](http://en.wikipedia.org/wiki/Query_language) are [T-SQL](http://en.wikipedia.org/wiki/Transact-SQL) and [ANSI SQL](http://en.wikipedia.org/wiki/SQL). SQL Server 2008 aims to make data management [self-tuning](http://en.wikipedia.org/wiki/Self-tuning), self organizing, and self maintaining with the development of SQL Server always on technologies, to provide near-zero downtime. SQL Server 2008 also includes support for [structured](http://en.wikipedia.org/wiki/Structured_data) and semi-structured data, including digital media formats for pictures, audio, video and other multimedia data. In current versions, such multimedia data can be stored as [BLOBs](http://en.wikipedia.org/wiki/Binary_large_object) (binary large objects), but they are generic bit streams. Intrinsic awareness of multimedia data will allow specialized functions to be performed on them.

*Architecture*

The protocol layer implements the external interface to SQL Server. All operations that can be invoked on SQL Server are communicated to it via a Microsoft-defined format, called [Tabular Data Stream](http://en.wikipedia.org/wiki/Tabular_Data_Stream) (TDS). TDS is an application layer protocol, used to transfer data between a database server and a client. Initially designed and developed by Sybase Inc. for their [Sybase SQL Server](http://en.wikipedia.org/wiki/Sybase_SQL_Server) relational database engine in 1984, and later by Microsoft in Microsoft SQL Server, In addition, the SQL Server API is also exposed over [web services](http://en.wikipedia.org/wiki/Web_service).

*Data Storage*

[Data storage](http://en.wikipedia.org/wiki/Computer_data_storage) is a [database](http://en.wikipedia.org/wiki/Database), which is a collection of tables with [typed](http://en.wikipedia.org/wiki/Type_system) columns. Microsoft SQL Server also allows user-defined composite types (UDTs) to be defined and used. It also makes server statistics available as virtual tables and views (called Dynamic Management Views or DMVs). In addition to tables, a database can also contain other objects including [views](http://en.wikipedia.org/wiki/View_(database)), [stored procedures](http://en.wikipedia.org/wiki/Stored_procedure), [indexes](http://en.wikipedia.org/wiki/Index_(database)) and [constraints](http://en.wikipedia.org/wiki/Constraint_(database)), along with a transaction log. A SQL Server database can contain a maximum of 231 objects, and can span multiple OS-level files with a maximum file size of 260 bytes. The data in the database are stored in primary data files with an extension.mdf. Secondary data files, identified with a .ndf extension, are used to store optional [metadata](http://en.wikipedia.org/wiki/Metadata). Log files are identified with the .ldf extension.

*Data Retrieval*

The main mode of retrieving data from an SQL Server database is [querying](http://en.wikipedia.org/wiki/Database_query) for it. The query is expressed using a variant of [SQL](http://en.wikipedia.org/wiki/SQL) called [T-SQL](http://en.wikipedia.org/wiki/T-SQL), a dialect Microsoft SQL Server shares with [Sybase SQL Server](http://en.wikipedia.org/wiki/Adaptive_Server_Enterprise) due to its legacy. The query [declaratively](http://en.wikipedia.org/wiki/Declarative_programming_language) specifies what is to be retrieved. It is processed by the query processor, which figures out the sequence of steps that will be necessary to retrieve the requested data. The sequence of actions necessary to execute a query is called a [query plan](http://en.wikipedia.org/wiki/Query_plan). There might be multiple ways to process the same query. For example, for a query that contains a [join](http://en.wikipedia.org/wiki/Join_(SQL)" \o "Join (SQL)) statement and a [select](http://en.wikipedia.org/wiki/Select_(SQL)) statement, executing join on both the tables and then executing select on the results would give the same result as selecting from each table and then executing the join, but result in different execution plans. In such case, SQL Server chooses the plan that is expected to yield the results in the shortest possible time. This is called [query optimization](http://en.wikipedia.org/wiki/Query_optimization) and is performed by the query processor itself.

*Visual Studio*

[Microsoft Visual Studio](http://en.wikipedia.org/wiki/Microsoft_Visual_Studio) includes native support for data programming with Microsoft SQL Server. It can be used to write and debug code to be executed by SQL CLR. It also includes a data designer that can be used to graphically create, view or edit database schemas. Queries can be created either visually or using code. SSMS 2008 onwards, provides [intellisense](http://en.wikipedia.org/wiki/Intellisense" \o "Intellisense) for SQL queries as well.

*SQL Server Management Studio*

[SQL Server Management Studio](http://en.wikipedia.org/wiki/SQL_Server_Management_Studio) is a [GUI](http://en.wikipedia.org/wiki/GUI) tool included with SQL Server 2008 and later for configuring, managing, and administering all components within Microsoft SQL Server. The tool includes both script editors and graphical tools that work with objects and features of the server. SQL Server Management Studio replaces [Enterprise Manager](http://en.wikipedia.org/wiki/Enterprise_Manager) as the primary management interface for Microsoft SQL Server since SQL Server 2005. A version of SQL Server Management Studio is also available for SQL Server Express Edition, for which it is known as SQL Server Management Studio Express (SSMSE).

A central feature of SQL Server Management Studio is the Object Explorer, which allows the user to browse, select, and act upon any of the objects within the server. It can be used to visually observe and analyze query plans and optimize the database performance, among others. SQL Server Management Studio can also be used to create a new database, alter any existing database schema by adding or modifying tables and indexes, or analyze performance. It includes the query windows which provide a GUI based interface to write and execute queries.

2.7 MICROSOFT VISUAL STUDIO 2010

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop console and graphical user interface applications along with Windows Forms applications, web sites, web applications, and web services in both native code together with managed code for all platforms supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silverlight.

Visual Studio includes a code editor supporting IntelliSense as well as [code refactoring](http://en.wikipedia.org/wiki/Code_refactoring) The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a forms designer for building [GUI](http://en.wikipedia.org/wiki/GUI) applications, [web designer](http://en.wikipedia.org/wiki/Web_designer), [class](http://en.wikipedia.org/wiki/Class_%28computing%29) designer, and [database schema](http://en.wikipedia.org/wiki/Database_schema) designer. It accepts plug-ins that enhance the functionality at almost every level—including adding support for [source-control](http://en.wikipedia.org/wiki/Source_control) systems (like [Subversion](http://en.wikipedia.org/wiki/Subversion_%28software%29) and [Visual SourceSafe](http://en.wikipedia.org/wiki/Visual_SourceSafe)) and adding new toolsets like editors and visual designers for [domain-specific languages](http://en.wikipedia.org/wiki/Domain-specific_language) or toolsets for other aspects of the [software development lifecycle](http://en.wikipedia.org/wiki/Software_development_lifecycle) (like the [Team Foundation Server](http://en.wikipedia.org/wiki/Team_Foundation_Server) client: Team Explorer).

Visual Studio supports different [programming languages](http://en.wikipedia.org/wiki/Programming_language) by means of language services, which allow the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include [C](http://en.wikipedia.org/wiki/C_%28programming_language%29)/[C++](http://en.wikipedia.org/wiki/C%2B%2B) (via [Visual C++](http://en.wikipedia.org/wiki/Visual_C%2B%2B)), [VB.NET](http://en.wikipedia.org/wiki/VB.NET) (via [Visual Basic .NET](http://en.wikipedia.org/wiki/Visual_Basic_.NET)), [C#](http://en.wikipedia.org/wiki/C_Sharp_%28programming_language%29) (via [Visual C#](http://en.wikipedia.org/wiki/Visual_C_Sharp)), and [F#](http://en.wikipedia.org/wiki/F_Sharp_%28programming_language%29) (as of Visual Studio 2010). Support for other languages such as [M](http://en.wikipedia.org/wiki/M_%28programming_language%29), [Python](http://en.wikipedia.org/wiki/IronPython), and [Ruby](http://en.wikipedia.org/wiki/IronRuby) among others is available via language services installed separately. It also supports [XML](http://en.wikipedia.org/wiki/XML)/[XSLT](http://en.wikipedia.org/wiki/XSLT), [HTML](http://en.wikipedia.org/wiki/HTML)/[XHTML](http://en.wikipedia.org/wiki/XHTML), [JavaScript](http://en.wikipedia.org/wiki/JavaScript) and [CSS](http://en.wikipedia.org/wiki/Cascading_Style_Sheets). Individual language-specific versions of Visual Studio also exist which provide more limited language services to the user: Microsoft Visual Basic, Visual J#, Visual C#, and Visual C++.

**3.FEASIBILITY ANALYSIS**

An important outcome of preliminary investigation is the determination that the system request is feasible. This is possible only if it is feasible within limited resource and time. The different feasibilities that have to be analyzed are

* Operational Feasibility
* Economic Feasibility
* Technical Feasibility

###### 3.1 **Operational Feasibility**

Operational Feasibility deals with the study of prospects of the system to be developed. This system operationally eliminates all the tensions of the Admin and helps him in effectively tracking the project progress. This kind of automation will surely reduce the time and energy, which previously consumed in manual work. Based on the study, the system is proved to be operationally feasible.

3.2 Economic Feasibility

Economic Feasibility or Cost-benefit is an assessment of the economic justification for a computer based project. As hardware was installed from the beginning & for lots of purposes thus the cost on project of hardware is low. Since the system is a network based, any number of employees connected to the LAN within that organization can use this tool from at anytime. The Virtual Private Network is to be developed using the existing resources of the organization. So the project is economically feasible.

###### 3.3 **Technical Feasibility**

According to Roger S. Pressman, Technical Feasibility is the assessment of the technical resources of the organization. The organization needs IBM compatible machines with a graphical web browser connected to the Internet and Intranet. The system is developed for platform Independent environment. Java Server Pages, JavaScript, HTML, SQL server and Web Logic Server are used to develop the system. The technical feasibility has been carried out. The system is technically feasible for development and can be developed with the existing facility.

3.4 Request Approval

Not all request projects are desirable or feasible. Some organization receives so many project requests from client users that only few of them are pursued. However, those projects that are both feasible and desirable should be put into schedule. After a project request is approved, it cost, priority, completion time and personnel requirement is estimated and used to determine where to add it to any project list. Truly speaking, the approval of those above factors, development works can be launched.

4.System design and development

4.1 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:’

* What data should be given as input?
* How the data should be arranged or coded?
* The dialog to guide the operating personnel in providing input.
* Methods for preparing input validations and steps to follow when error occur.

4.1.1 OBJECTIVES

1.Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3.When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user

will not be in maize of instant. Thus the objective of input design is to create an input layout .

4.2 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2.Select methods for presenting information.

3.Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

* Convey information about past activities, current status or projections of the
* Future.
* Signal important events, opportunities, problems, or warnings.
* Trigger an action.
* Confirm an action.

4.3 SYSTEM DESIGN

4.3.1 *Introduction*

Systems designs the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. Object-oriented analysis and design methods are becoming the most widely used methods for computer systems design. The UML has become the standard language in object-oriented analysis and design. It is widely used for modeling software systems and is increasingly used for high designing of non-software systems and organizations.The unified modeling language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

* + A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

4.3.2 *User Model View*

* + 1. This view represents the system from the users perspective.
    2. The analysis representation describes a usage scenario from the end-users perspective.

4.3.3 *Structural model view*

* In this model the data and functionality are arrived from inside the system.
* This model view models the static structures.

4.3.4 *Behavioral Model View*

* It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

*4.3.5 Implementation Model View*

In this the structural and behavioral as parts of the system are represented as they are to be built.

*4.3.6 Environmental Model View*

In this the structural and behavioral aspects of the environment in which the system is to be implemented are represented.

UML is specifically constructed through two different domains they are

* UML Analysis modeling, which focuses on the user model and structural model views of the system
* UML design modeling, which focuses on the behavioral modeling, implementation modeling and environmental model views.

Use case Diagrams represent the functionality of the system from a user’s point of view. Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behavior of the system from external point of view.

Actors are external entities that interact with the system. Examples of actors include users like administrator, bank customer …etc., or another system like central database.

4.4 Usecase-diagram

Use case diagrams are used to visualize, specify, construct, and document the (intended) behavior of the system, during requirements capture and analysis. Provide a way for developers, domain experts and end-users to communicate. Serve as basis for testing. Use case diagrams contain use cases, actors, and their relationships. Use cases specify desired behavior. A use case is a description of a set of sequences of actions, including variants; a system performs to yield an observable result of value to an actor. Each sequence represent an interaction of actors with the system.

*4.4.1 Actor:*

An actor is a user of the system and is depicted using a stick figure. the role of the user is written beneath the icon. Actors are not mentioned to humans. If a system communicates with another application, and expects input or delivers output, then that application can also be considered an actor .



*4.4.2 Use-case:*

A use-case is functionally provided by the system, typically described as verb+object. Use-cases are depicted with an ellipse. The name of the use case is written beneath the ellipse.



*4.4.3 Association:*

Associations are used to link Actors with use cases,and indicate that an actor participates in the use case in some form.Associtions are depicted by a line connecting the actor and the use-case.



*4.4.4 Activity Diagram:*

Its basic form, an activity diagram is a simple and intuitive illustration of what happens in a workflow, what activities can be done in parallel, and whether there are alternative paths through the workflow. Activity diagrams represent the business and operational work flows of a system. An Activity diagram is a dynamic diagram that shows the activity and the event that causes the object to be in the particular state. Activity diagrams are similar to state diagrams because activities are the state of doing something. The diagrams describe the state of activities by showing the sequence of activities performed. Activity diagrams can show activities that are conditional or parallel. A State diagram shows the different states an object is in during the lifecycle of its existence in the system, and the transitions in the states of the objects. These transitions depict the activities causing these transitions, shown by arrows. An Activity diagram talks more about these transitions and activities causing the changes in the object states. However, activity diagrams should not take the place of interaction diagrams and state diagrams. Activity diagrams do not give detail about how objects behave or how objects collaborate.

*Initial Activity:* This is first activity of the flow. It is shown as filled circle. Place the initial activity in the top-left corner.

*Final Activity:* The end of activity diagram is shown as bull's eye symbol. This is also called as a final activity. The final activity is optional. Sometimes an activity is simply a dead end, but if this is the case then there is no harm in indicating that the only transition is to an ending point. When someone else reads your diagram, they know that you have considered how to exit these activities.

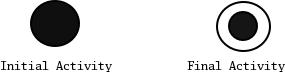


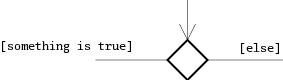
Figure 4.3 Notations for initial and final activities

*Activity States*: Activity state is a state that represents the execution of an atomic action, typically the invocation of an operation or transaction. Activity state depicted as a rectangle with straight top and bottom and with convex arcs on the two sides. Text in the activity box represents an activity (the activity name). State names should reflect the lifecycle stage of an action object.  
A activity has transitions into it but there no transition to other activity. Such activity is called as black hole activity. A black hole activity typically indicates either you have missed one or more transitions. As noted previously if an activity is in fact the final activity in the process you are modeling you should consider adding a transition out of it to an end point. A miracle activity is one that has transitions out of it but none into it, something that should be true only of start points. Once again, this is an indication that you have missed one or more transitions.

Figure 4.2  
Figure 4.4 Activity State

*Composite Activity State:* A concurrent substratethat can be held simultaneously with other sub states contained in the same composite state. This can be represented by drawing two parallel transaction lines from super activity state to concurrent activities.  
Ensure that exit transition guards and activity invariants must form a complete set. An activity invariant is a condition that is always true when your system is processing an activity. Clearly the conditions that are true while processing an activity must not overlap with its exit conditions and the invariants & exit conditions must "adds up" to a complete set.

*Branching:* A diamond represents a decision with alternate paths. The outgoing alternates should be labeled with a condition or guard expression. You can also use guard to one of the paths "else." The guards depicted using the format on the transitions leaving the decision point also help to describe the decision point. Avoid superfluous decision points by naming the activity such that it may include implied decision. To ensure that you have thought through all possibilities for a decision point, each transition leaving a decision point must have a guard. The guards on the transitions leaving a decision point, or an activity, must be consistent with one another. For example guards such as x <0, x = 0, and x > 0 are consistent whereas guard such as x <= 0 and x >= 0 are not consistent because they overlap – it isn’t clear what should happen when x is 0. Once you have reached a decision point it must always be possible to leave it, therefore the guards on its exit transitions must be complete. For example, guards such as x < 0 and x >0 are not complete because it isn’t clear what happens when x is 0.

  
Figure 4.5 Branching or decision box

4.5 Class diagram:

A class diagram describes the types of objects in the system and the various kinds of static relationships that exist among them. It is a graphical representation of a static view on declarative static elements and a central modeling technique that runs through nearly all object-oriented methods. The richest notation in UML. This tutorial covers the following key elements of class diagrams together with example and the implementation using high-level programming language.

#### *Essential Elements:*

##### Classes

##### Attributes

##### Operations

##### Relationships

*4.5.1 Class:*

Classes are the building blocks in object-oriented programming. A class is depicted using a rectangle divided into three sections. It is depicted using a rectangle divided into three sections. the top section is the name of the class. The middle section defines the properties of the class. the bottom section lists the methods of the class.



## *Association*

Association is a relationship between two objects. In other words, association defines the multiplicity between objects. You may be aware of one-to-one, one-to-many, many-to-one, many-to-many all these words define an association between objects. Aggregation is a special form of association. Composition is a special form of aggregation.

http://javapapers.com/wp-content/uploads/2010/06/association.jpg

**Example:**A Student and a Faculty are having an association.

## *Aggregation*

Aggregation is a special case of association. A directional association between objects. When an object ‘has-a’ another object, then you have got an aggregation between them. Direction between them specified which object contains the other object. Aggregation is also called a “Has-a” relationship.

http://javapapers.com/wp-content/uploads/2010/06/aggregation.jpg

## *Composition*

Composition is a special case of aggregation. In a more specific manner, a restricted aggregation is called composition. When an object contains the other object, if the contained object cannot exist without the existence of container object, then it is called composition.

http://javapapers.com/wp-content/uploads/2010/06/composition.jpg

## *Generalization:*

Generalization describes the inheritance relationship of the object oriented world. It is parent and child relationship.

Generalization is represented by an arrow with hollow arrow head as shown below. One end represents the parent element and the other end child element.

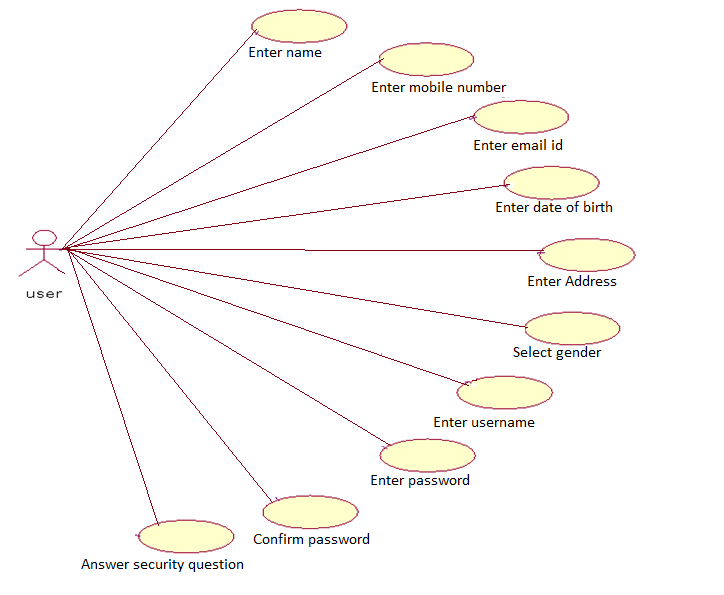
Generalization Notation

Generalization is used to describe parent-child relationship of two elements of a system

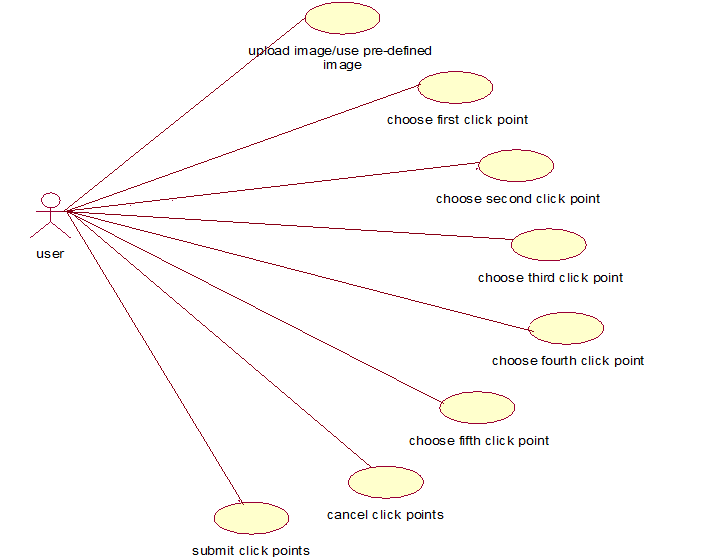
**4.6 DESIGN OF MODULES**

***4.6.1 Pass points module***

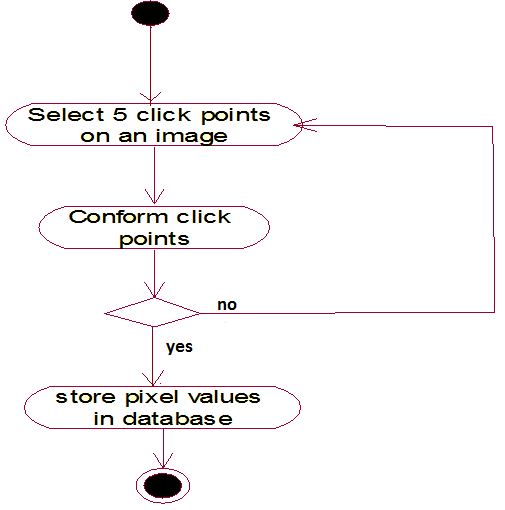
Use case diagram for registration



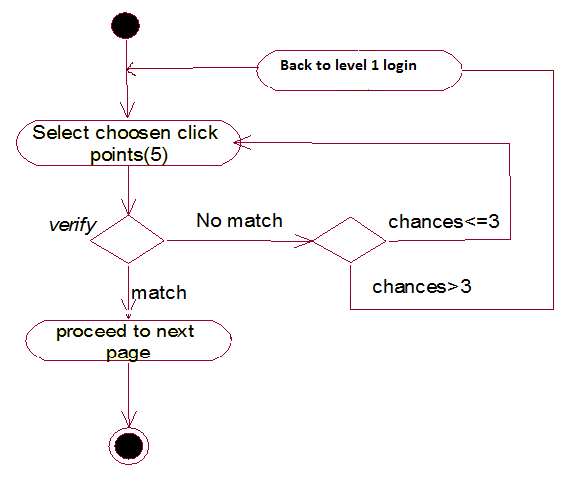
*4.6.2 Use case diagram for pass points technique*



*4.6.3 Activity diagram (registration)*



*4.6.4 Activity diagram (login)*

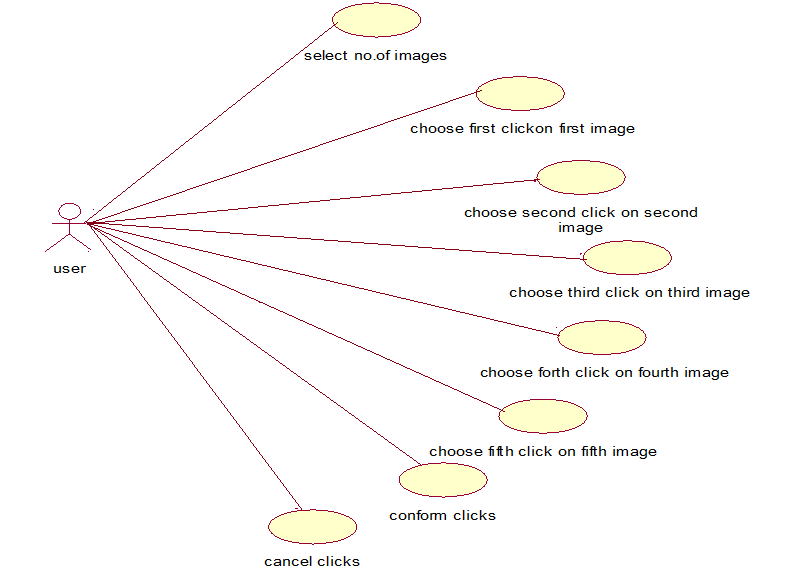


*4.6.5 Sequence diagram*

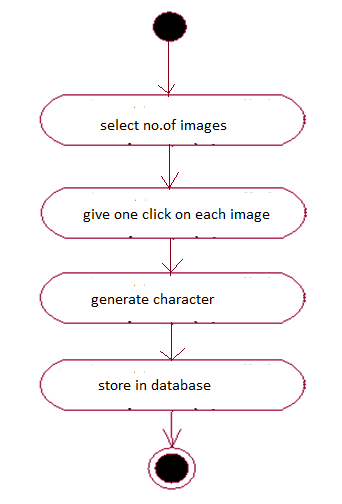


4.7 CUED CLICK POINTS MODULE

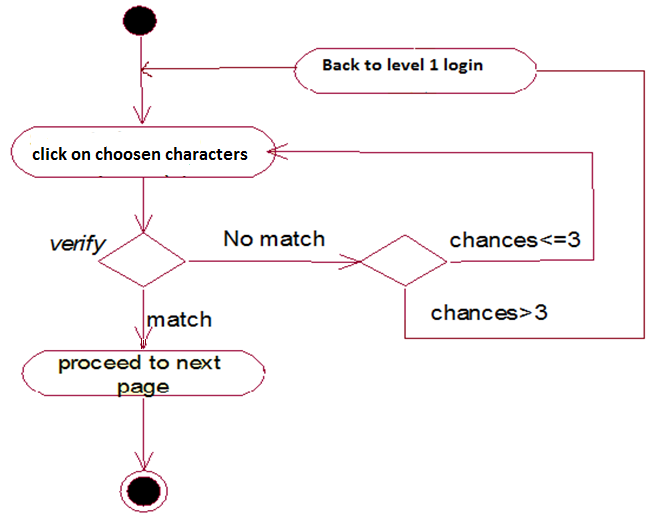
*4.7.1 Use case diagram*



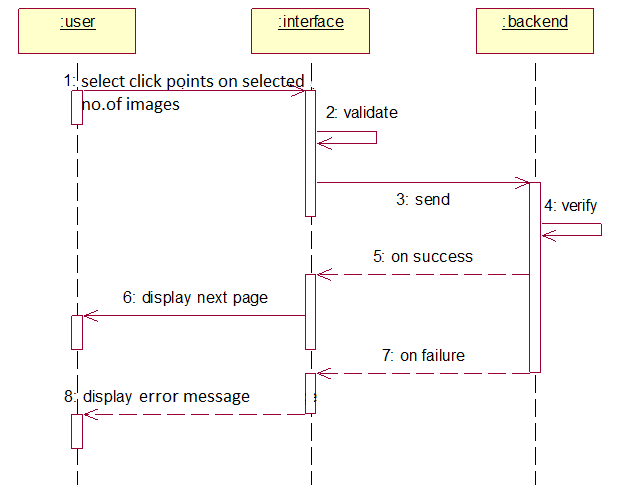
*4.7.2 Activity diagram(registration)*



*4.7.3 Activity Diagram(login)*

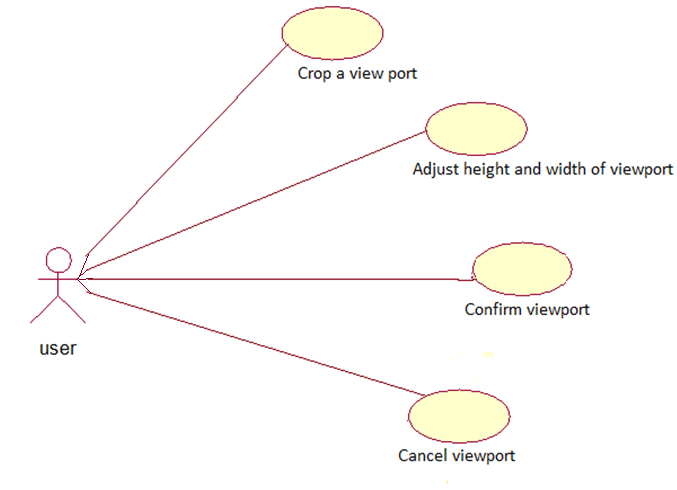


*4.7.4 Sequence diagram*

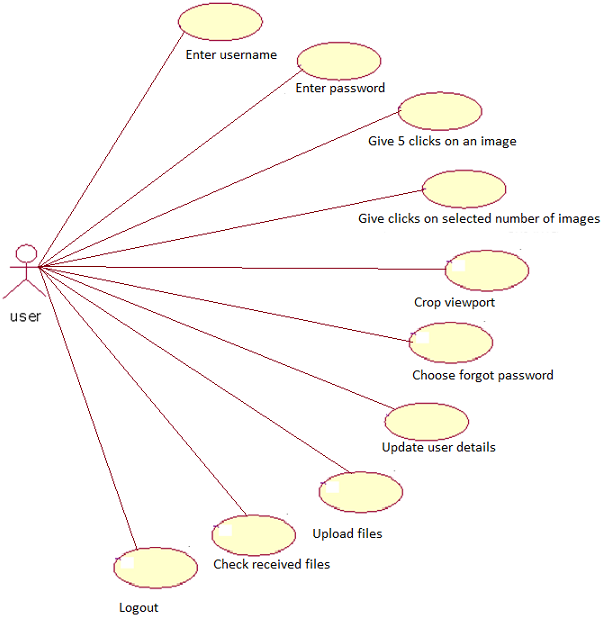


4.8 PERSUASIVE CUED CLICK POINTS MODULE

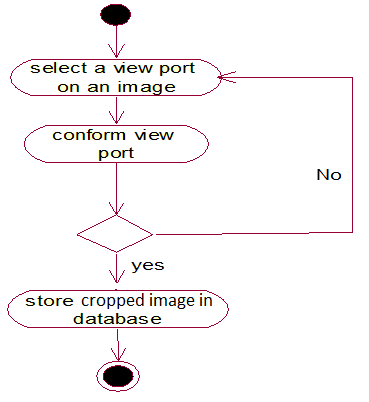
*4.8.1 Use case diagram for persuasive click points technique*



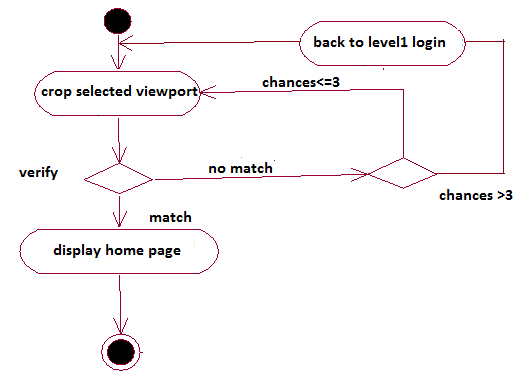
4.8.2 Use case diagram for login



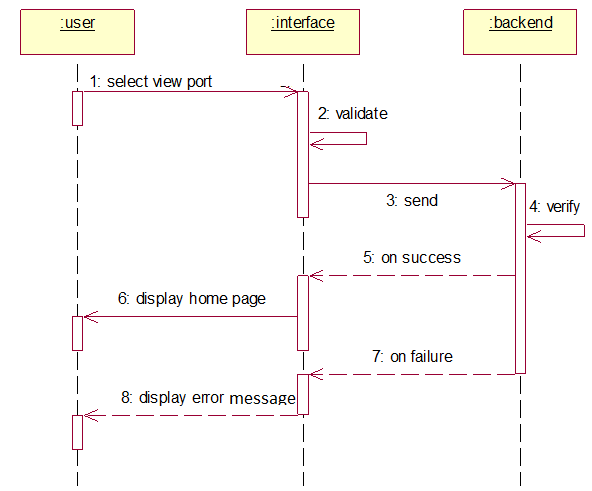
*4.8.3 Activity diagram(registration)*



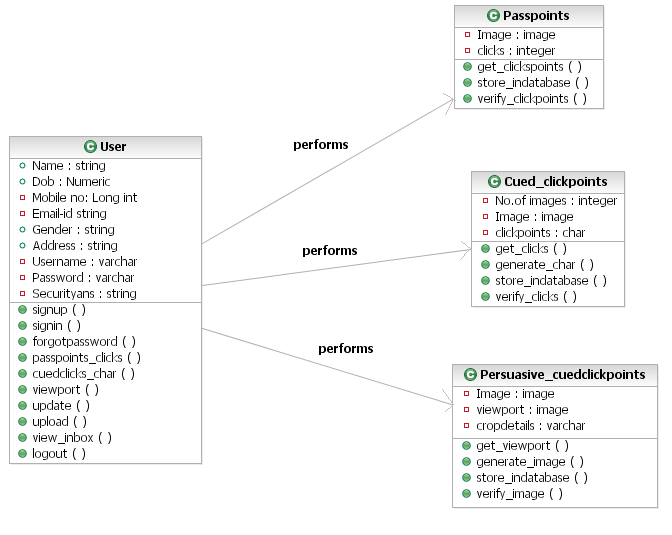
*4.8.4 Activity diagram(login)*



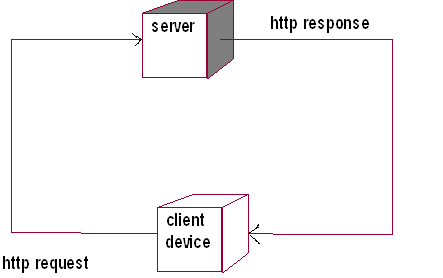
*4.8.5 Sequence diagram*



4.9 CLASS DIAGRAM



4.10 Deployment diagram



5. IMPLEMENTATION AND CODE SNIPPETS:

As mentioned earlier, the main focus of the project is on authentication using three modules. They are,

1. Pass Points module.

2. Cued Click points module.

3. Persuasive Cued Click points modules.

We have tried to integrate all these in our project

5.1 PASS POINTS MODULE:

* Here the user needs to specify 5 click points on a single image

*Register*

* It is implemented by storing the click point coordinates in the database..FIrstly we take an image button tool for the image..The mouser event will give the coordinates of the pixel when the mouse moves over it. It is stored in Label
* One more label is used to store the no. of mouse clicks. Every time the user clicks the image the label is updated with the pixel coordinates .After 5 clicks when the user presses the next button. click points are stored in the fpoint coloumn in the database

*login*

During the login time the click points are verified with the regis table which has already stored the click points. If label data matches with that of the database then we can sign in else incorrect message is displayed

*5.1.1. Sample Code:*

Design Page for Passpoints(.aspx)

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="register2.aspx.cs" Inherits="register2" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<%@ Register Src="~/Heading.ascx" TagName="Heading" TagPrefix="wuc" %>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title>PCCP</title>

<script type="text/javascript" language="javascript">

var x;

var y;

function mouser(event) {

x = event.clientX-92;

y = event.clientY-257;

document.getElementById('Label3').innerHTML = x + 'px' + ',' + y + 'px';

}

</script>

<style type="text/css">

#demo

{

width: 300px;

}

</style>

</head>

<body>

<form id="form1" runat="server">

<div>

<wuc:Heading ID="Heading1" runat="server" />

<table style="position:absolute; top: 197px; left: 89px;">

<tr>

<td>

<asp:Label ID="Label5" runat="server" Font-Bold="True"

Font-Names="DigifaceWide" Font-Size="20px" ForeColor="#9900CC"

Text="Step : 2"></asp:Label>

</td>

</tr>

<tr>

<td>

<asp:Label ID="Label1" runat="server" Font-Bold="True"

Font-Names="Freestyle Script" Font-Size="20px" ForeColor="#006600"

Text="Click 5 Places in the Following Image"></asp:Label>

<asp:Label ID="Label2" runat="server" Font-Bold="True"

Font-Names="Consolas" Font-Size="16px" ForeColor="#663300"

Text="(Note : In this 5 Clicking points are used to your login time)"></asp:Label>

</td>

</tr>

<tr>

<td>

<div id="demo" onmousemove="mouser(event)">

<asp:ImageButton ID="ImageButton1" runat="server" Height="300px" Width="300px" ImageUrl="~/images/5\_21.jpg"

CausesValidation="False" onclick="ImageButton1\_Click" />

</div>

<asp:Label ID="Label3" runat="server"></asp:Label>

<asp:Label ID="Label4" runat="server"></asp:Label>

<asp:Label ID="Label6" runat="server" Visible="false"></asp:Label>

</td>

</tr>

<tr>

<td align="center">

&nbsp;

<asp:ImageButton ID="ImageButton3" runat="server" Height="47px"

ImageUrl="~/images/Cancel.png" onclick="ImageButton3\_Click" Width="71px" />

&nbsp;

<asp:ImageButton ID="ImageButton2" runat="server" Height="44px" Width="81px"

ImageUrl="~/images/NEXT.gif" onclick="ImageButton2\_Click" />

</td>

</tr>

</table>

</div>

</form>

</body>

</html>

Logic Page(.aspx.cs)

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.SqlClient;

using System.Configuration;

public partial class register2 : System.Web.UI.Page

{

SqlConnection con = new SqlConnection(ConfigurationManager.AppSettings["connection"]);

protected void Page\_Load(object sender, EventArgs e)

{

if (!IsPostBack)

{

Label6.Text = "0";

Label4.Text = "";

}

}

protected void ImageButton1\_Click(object sender, ImageClickEventArgs e)

{

if (Convert.ToInt32(Label6.Text) < 5)

{

Label6.Text = Convert.ToString(Convert.ToInt32(Label6.Text)+1);

Label4.Text = Label4.Text + e.X.ToString() + "," + e.Y.ToString() + ",";

}

else

{

RegisterStartupScript("msg", "<script>alert('You Click Only 5 Points.')</script>");

}

}

protected void ImageButton2\_Click(object sender, ImageClickEventArgs e)

{

if (Convert.ToInt32(Label6.Text) == 5)

{

con.Open();

SqlCommand cmd = new SqlCommand("update regis set fpoint='"+Label4.Text+"' where id='"+(string)Session["id"]+"'", con);

cmd.ExecuteNonQuery();

con.Close();

Label6.Text = "0";

Label4.Text = "";

Response.Redirect("register3.aspx");

}

else

{

RegisterStartupScript("msg", "<script>alert('You Only Clicked " + (Convert.ToInt32(Label6.Text)) + " Points. Click Reminder " + (5 - (Convert.ToInt32(Label6.Text))) + " Points...')</script>");

}

}

protected void ImageButton3\_Click(object sender, ImageClickEventArgs e)

{

con.Open();

SqlCommand cmd1 = new SqlCommand("delete from regis where id='"+(string)Session["id"]+"'", con);

cmd1.ExecuteNonQuery();

Response.Redirect("Default.aspx");

con.Close();

}

}

5.2 CUED CLICK POINTS MODULE:

* Cued Click Points (CCP) was developed as an alternative click based graphical password scheme where users select one point per image for five images.
* Here we make use of image map for the image unlike the image button we used in the first module...

Here we divide the IMAGEMAP into 100 regions called RECTANGULAR HOTSPOT regions whose values are uniquely specified and stored in the database..

When the user clicks on the rectangular hotspot region..the postback value for the region is stored in the label..this data is stored in the fopoint coloumn of the db.

This process is repeated for all the five images

*login*

when we click on the image map it navigates to another image map.Process is repeated for 5 images .All click points rectangular hotspot values are compared with the already registered fopoint coloumn values in the db..If they match we can sign in

*Sample Code:*

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="register3.aspx.cs" Inherits="register3" %>

<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit" TagPrefix="cc1" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<%@ Register Src="~/Heading.ascx" TagName="PCCP" TagPrefix="ww1"%>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title>PCCP</title>

<style type="text/css">

.back

{

background-color: #D8D8D8;

filter: alpha(opacity=70);

opacity: 0.7;

}

.style1

{

width: 512px;

}

.style2

{

width: 827px;

}

</style>

</head>

<asp:Button ID="Button2" runat="server" Text="Ok" onclick="Button2\_Click"

style="position:absolute; top: 302px; left: 118px;" />

</asp:Panel>

<table style="position:absolute; top: 197px; left: 89px; height: 423px; width: 1581px;">

<tr>

<td colspan="6">

<asp:Label ID="Label15" runat="server" Font-Bold="True"

Font-Names="Freestyle Script" Font-Size="20px" ForeColor="#006600"

Text="Click any one location in the following Images"></asp:Label>

<br />

<asp:Label ID="Label19" runat="server" Font-Bold="True"

Font-Names="Consolas" Font-Size="16px" ForeColor="#663300"

Text="(Note : One click in an each images and this the password in Login Time)"></asp:Label>

</td>

</tr>

<tr>

<td align="right" class="style1">

<asp:Button ID="Button1" runat="server" ForeColor="Red" Text="Example"

Width="68px" onclick="Button1\_Click" />

</td>

<td align="right" class="style2">

&nbsp;</td>

<td align="right">

&nbsp;</td>

<td align="right">

&nbsp;</td>

<td align="right">

&nbsp;</td>

<td align="right">

&nbsp;</td>

</tr>

<tr>

<td colspan="6">

<asp:ImageMap ID="ImageMap1" runat="server" Height="300px" Width="300px"

ImageUrl="~/images/5\_22.jpg" onclick="ImageMap1\_Click">

<asp:RectangleHotSpot HotSpotMode="PostBack" Top="0" Left="0" Right="30" Bottom="30" AlternateText="A" PostBackValue="1" />

<asp:RectangleHotSpot HotSpotMode="PostBack" Top="0" Left="30" Right="60" Bottom="30" AlternateText="B" PostBackValue="2" />

<asp:RectangleHotSpot HotSpotMode="PostBack" Top="0" Left="60" Right="90" Bottom="30" AlternateText="C" PostBackValue="3" />

<asp:RectangleHotSpot HotSpotMode="PostBack" Top="0" Left="90" Right="120" Bottom="30" AlternateText="D" PostBackValue="4" />

Upto……

<asp:RectangleHotSpot HotSpotMode="PostBack" Top="270" Left="210" Right="240" Bottom="300" AlternateText="Dd" PostBackValue="98" />

<asp:RectangleHotSpot HotSpotMode="PostBack" Top="270" Left="240" Right="270" Bottom="300" AlternateText="Ee" PostBackValue="99" />

<asp:RectangleHotSpot HotSpotMode="PostBack" Top="270" Left="270" Right="300" Bottom="300" AlternateText="Ff" PostBackValue="100" />

</asp:ImageMap>

We repeat the process for the other 4 images taking 4\*100 cases for the 4 image maps. Each case has a uniquely specified value in the database.

*Logic:*

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.SqlClient;

using System.Configuration;

public partial class register3 : System.Web.UI.Page

{

SqlConnection con = new SqlConnection(ConfigurationManager.AppSettings["connection"]);

protected void Page\_Load(object sender, EventArgs e)

{

if (!IsPostBack)

{

Label1.Text = "0";

Label2.Text = "";

ImageMap1.Visible = true;

ImageMap2.Visible = false;

ImageMap3.Visible = false;

ImageMap4.Visible = false;

ImageMap5.Visible = false;

ImageButton2.Visible = false;

}

}

protected void Button2\_Click(object sender, EventArgs e)

{

Panel1.Visible = false;

Button2.Visible = false;

Image2.Visible = false;

}

protected void Button1\_Click(object sender, EventArgs e)

{

Panel1.Visible = true;

Button2.Visible = true;

Image2.Visible = true;

}

protected void ImageMap1\_Click(object sender, ImageMapEventArgs e)

{

Label1.Text = Convert.ToString(Convert.ToInt32(Label1.Text) + 1);

if (Convert.ToInt32(Label1.Text) <= 5)

{

ImageMap1.Visible = false;

ImageMap2.Visible = true;

ImageMap3.Visible = false;

ImageMap4.Visible = false;

ImageMap5.Visible = false;

switch (e.PostBackValue)

{

case "1":

Label2.Text = "1A" + ",";

break;

case "2":

Label2.Text = "1B" + ",";

break;

case "3":

Label2.Text = "1C" + ",";

break;

case "4":

Label2.Text = "1D" + ",";

break;

case "5":

Label2.Text = "1E" + ",";

break;

case "6":

Label2.Text = "1F" + ",";

break;

upto…….

case "100":

Label2.Text = "1ll" + ",";

break;

We repeat the process for the other 4 images taking 4\*100 cases for the 4 image maps. Each case has a uniquely specified value in the database.

5.3 VIEW PORT MODULE OR PERSUASIVE CLICK POINTS MODULES :

* To address the issue of hotspots, Persuasive Cued Click Points (PCCP) was proposed.
* We use a jquery function to implement this module
* Ok, the million dollar question - What is jQuery anyways?
* jQuery is a fast, lightweight JavaScript library that is CSS3 compliant and supports many browsers.

What is the difference between JavaScript and jQuery?

* JavaScript is a language whereas jQuery is a library written using JavaScript.
* As already stated jquery is a library it has a plugin called jcrop.when the jcrop action is performed to the image ..the image is cropped and stored in the db in binary format..the parameters like width ,height,x,y are calculated for the jcrop

*login*

* Again jcrop action is performed and the cropped image's parameters are compared with the db..If same,we can login to the home page
* A few other actions can be performed after logging in like uploading a file to another db..updating the profile..checking the inbox,logging out from ur account

*Sample Code:*

Design:

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="register4.aspx.cs" Inherits="register4" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<%@ Register Src="~/Heading.ascx" TagName="Heading" TagPrefix="wuc" %>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title>PCCP</title>

<link href="jquery.Jcrop.css" rel="stylesheet" type="text/css" />

<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.3/jquery.min.js"></script>

<script type="text/javascript" src="jquery.Jcrop.pack.js"></script>

<script type="text/javascript">

jQuery(document).ready(function() {

jQuery('#imgCrop').Jcrop({

onSelect: storeCoords

});

});

function storeCoords(c) {

jQuery('#X').val(c.x);

jQuery('#Y').val(c.y);

jQuery('#W').val(c.w);

jQuery('#H').val(c.h);

document.getElementById('Label21').innerHTML = 'Width=' + c.w + ' Height=' + c.h + ' X=' + c.x + ' Y=' + c.y;

//"Width=" + w + ", Height=" + h + ", X=" + x + ", Y=" + y;

};

</script>

<style type="text/css">

#demo

{

width: 300px;

}

</style>

</head>

<body>

<form id="form1" runat="server">

<div>

<wuc:Heading ID="Heading1" runat="server" />

<table style="position:absolute; top: 197px; left: 89px; height: 576px;">

<tr>

<td>

<asp:Label ID="Label16" runat="server" Font-Bold="True"

Font-Names="Freestyle Script" Font-Size="20px" ForeColor="#006600"

Text="Confirm Password"></asp:Label>

<br />

<asp:Label ID="Label20" runat="server" Font-Bold="True"

Font-Names="Consolas" Font-Size="16px" ForeColor="#663300"

Text="(Note : In this 5 Clicking points are used to your login time)"></asp:Label>

</td>

</tr>

<tr>

<td>

<div id="demo">

<asp:Panel ID="pnlCrop" runat="server" Width="300px">

<asp:Image ID="imgCrop" runat="server" Height="300px"

ImageUrl="~/images/5\_10.jpg" Width="300px" />

</asp:Panel>

</div>

<asp:Label ID="Label21" runat="server" Font-Bold="True" Font-Names="Calibri"

Font-Size="16px" ForeColor="#CC0000"></asp:Label>

</td>

</tr>

<tr>

<td>

<asp:Panel ID="pnlCropped" runat="server" Visible="false">

<asp:Image ID="imgCropped" runat="server" Visible="false" />

</asp:Panel>

</td>

</tr>

<tr>

<td align="center">

<asp:ImageButton ID="ImageButton3" runat="server" Height="48px"

ImageUrl="~/images/Cancel.png" onclick="ImageButton3\_Click" Width="58px" />

&nbsp;&nbsp;

<asp:ImageButton ID="ImageButton2" runat="server" Height="48px" Width="59px"

ImageUrl="~/images/Submit.png" onclick="ImageButton2\_Click" />

</td>

</tr>

</table>

<asp:HiddenField ID="X" runat="server" />

<asp:HiddenField ID="Y" runat="server" />

<asp:HiddenField ID="W" runat="server" />

<asp:HiddenField ID="H" runat="server" />

</div>

</form>

</body>

</html>

Logic:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Drawing.Imaging;

using System.Drawing;

using System.Configuration;

using System.Data;

using System.Web.Security;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.IO;

using SD = System.Drawing;

using System.Drawing.Drawing2D;

using System.Data.SqlClient;

public partial class register4 : System.Web.UI.Page

{

SqlConnection con = new SqlConnection(ConfigurationManager.AppSettings["connection"]);

String path = HttpContext.Current.Request.PhysicalApplicationPath + "images\\";

string imagenames;

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void ImageButton2\_Click(object sender, ImageClickEventArgs e)

{

string ImageName = "5\_10.jpg";

int w = Convert.ToInt32(W.Value);

int h = Convert.ToInt32(H.Value);

int x = Convert.ToInt32(X.Value);

int y = Convert.ToInt32(Y.Value);

Label21.Text = "Width=" + w + ", Height=" + h + ", X=" + x + ", Y=" + y;

byte[] CropImage = Crop(path + ImageName, w, h, x, y);

using (MemoryStream ms = new MemoryStream(CropImage, 0, CropImage.Length))

{

ms.Write(CropImage, 0, CropImage.Length);

using (SD.Image CroppedImage = SD.Image.FromStream(ms, true))

{

string SaveTo = path + "crop" + (string)Session["names"] + (string)Session["id"]+".jpg";

CroppedImage.Save(SaveTo, CroppedImage.RawFormat);

pnlCrop.Visible = true;

pnlCropped.Visible = false;

imgCropped.ImageUrl = "images/crop" + (string)Session["names"] + (string)Session["id"] + ".jpg";

imagenames="crop" + (string)Session["names"] + (string)Session["id"] + ".jpg";

}

}

con.Open();

SqlCommand cmd = new SqlCommand("update regis set cropimage=@cropimage,names='"+imagenames+"' where id='" + (string)Session["id"] + "'", con);

cmd.Parameters.AddWithValue("@cropimage", CropImage);

cmd.ExecuteNonQuery();

Response.Redirect("Default.aspx");

con.Close();

}

static byte[] Crop(string Img, int Width, int Height, int X, int Y)

{

try

{

using (SD.Image OriginalImage = SD.Image.FromFile(Img))

{

using (SD.Bitmap bmp = new SD.Bitmap(Width, Height))

{

bmp.SetResolution(OriginalImage.HorizontalResolution, OriginalImage.VerticalResolution);

using (SD.Graphics Graphic = SD.Graphics.FromImage(bmp))

{

Graphic.SmoothingMode = SmoothingMode.AntiAlias;

Graphic.InterpolationMode = InterpolationMode.HighQualityBicubic;

Graphic.PixelOffsetMode = PixelOffsetMode.HighQuality;

Graphic.DrawImage(OriginalImage, new SD.Rectangle(0, 0, Width, Height), X, Y, Width, Height, SD.GraphicsUnit.Pixel);

MemoryStream ms = new MemoryStream();

bmp.Save(ms, OriginalImage.RawFormat);

return ms.GetBuffer();

}

}

}

}

catch (Exception Ex)

{

throw (Ex);

}

}

protected void ImageButton3\_Click(object sender, ImageClickEventArgs e)

{

con.Open();

SqlCommand cmd1 = new SqlCommand("delete from regis where id='" + (string)Session["id"] + "'", con);

cmd1.ExecuteNonQuery();

Response.Redirect("Default.aspx");

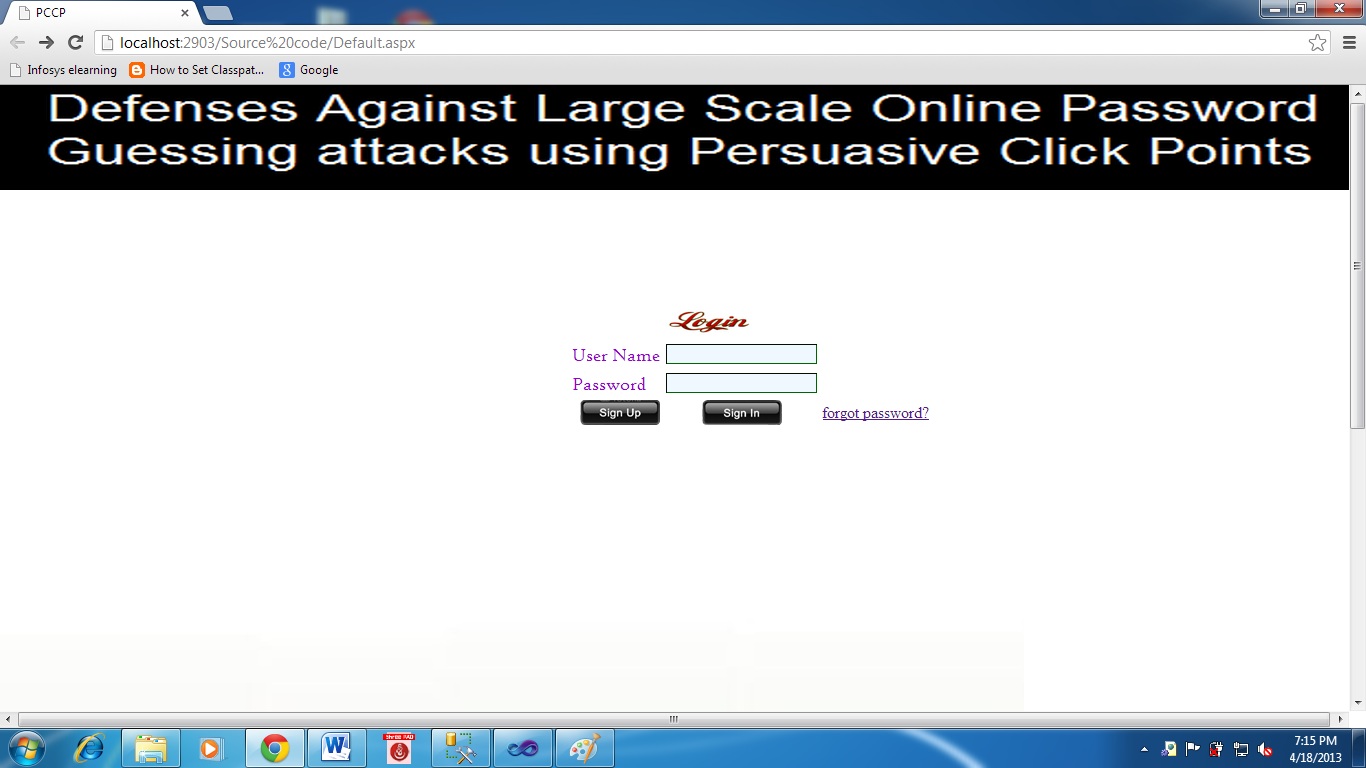
con.Close();

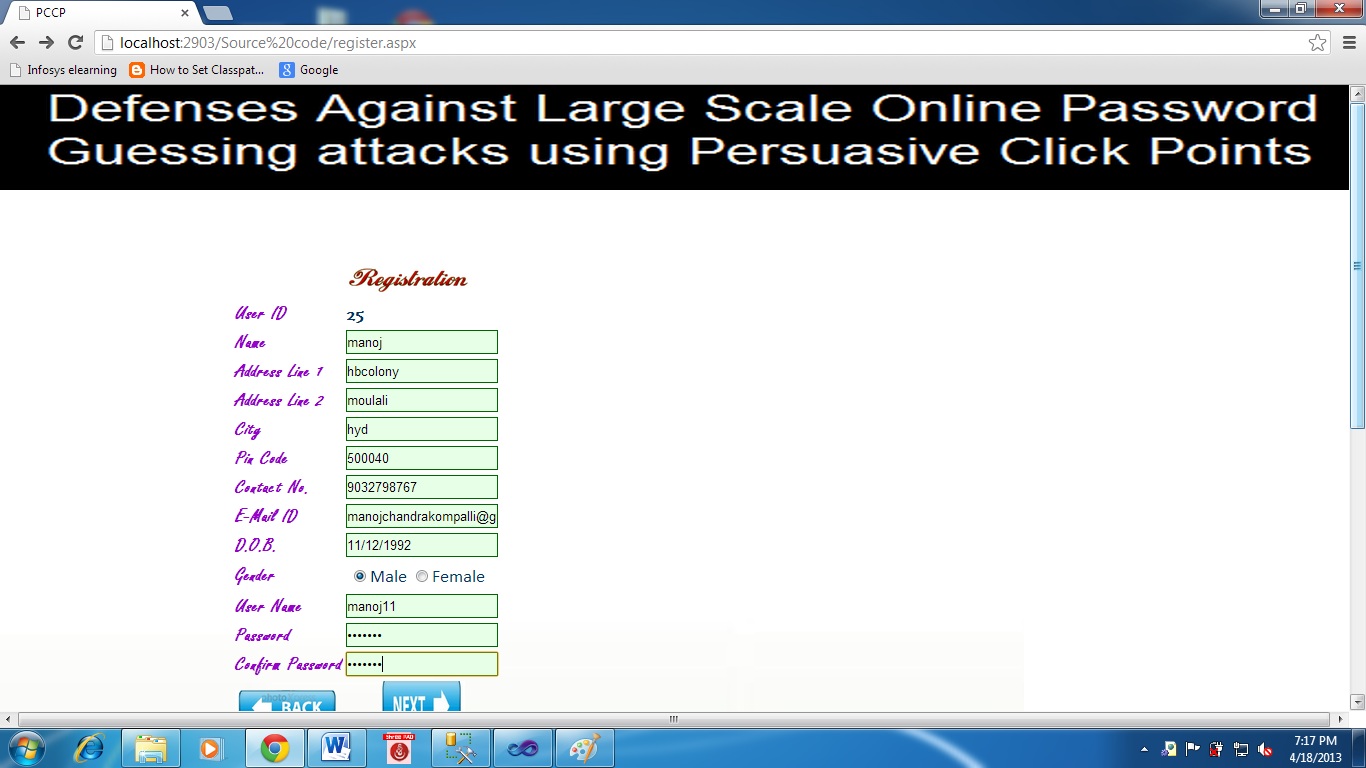
}

}

6. SCREEN SHOTS

6.1.Login Page

6.2.When user clicks on sign up button



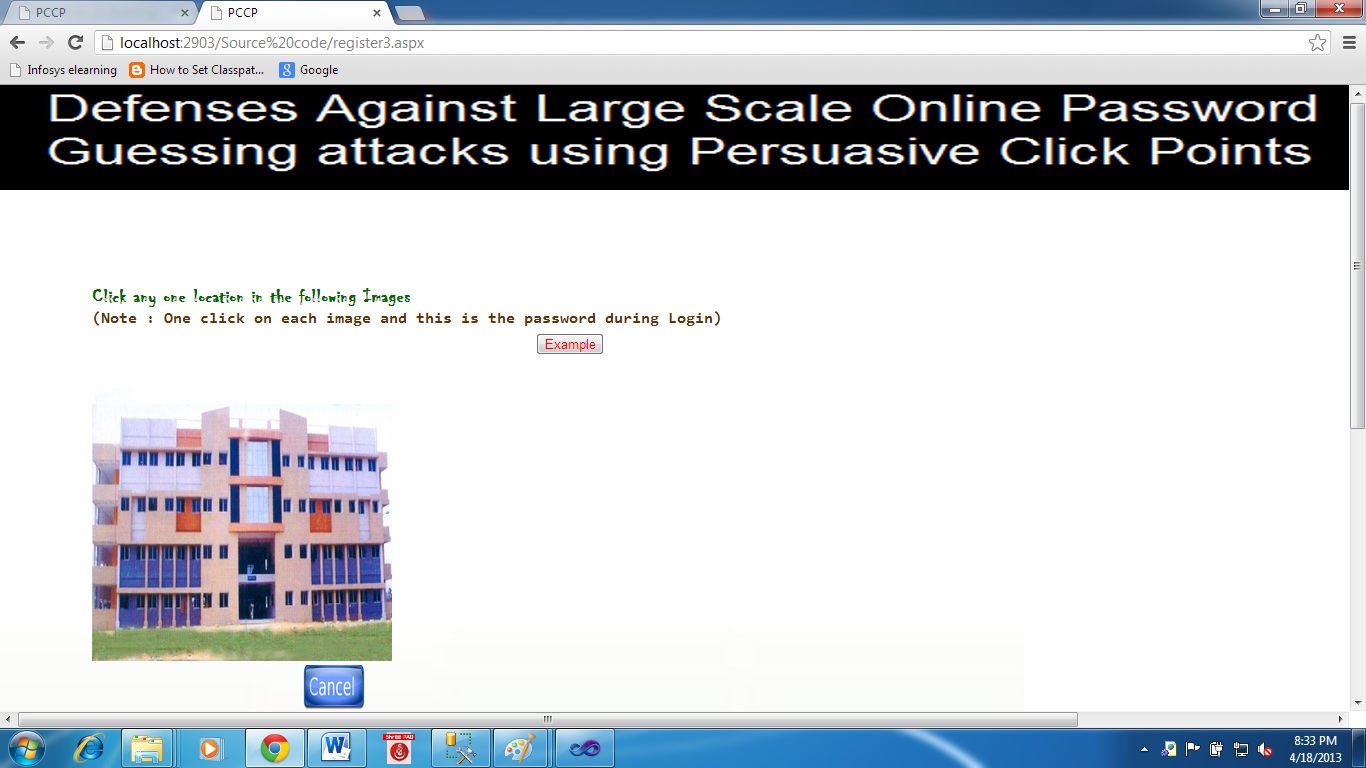
6.3.Registering using pass points module(5 consecutive clicks on single image)



6.4. Registering using click points module using single click on an image(image1)



6.5. Registering using click points module using single click on an image(image2)



6.6 Registering using click points module using single click on an image(image3)



6.7 Registering using click points module using single click on an image(image4)

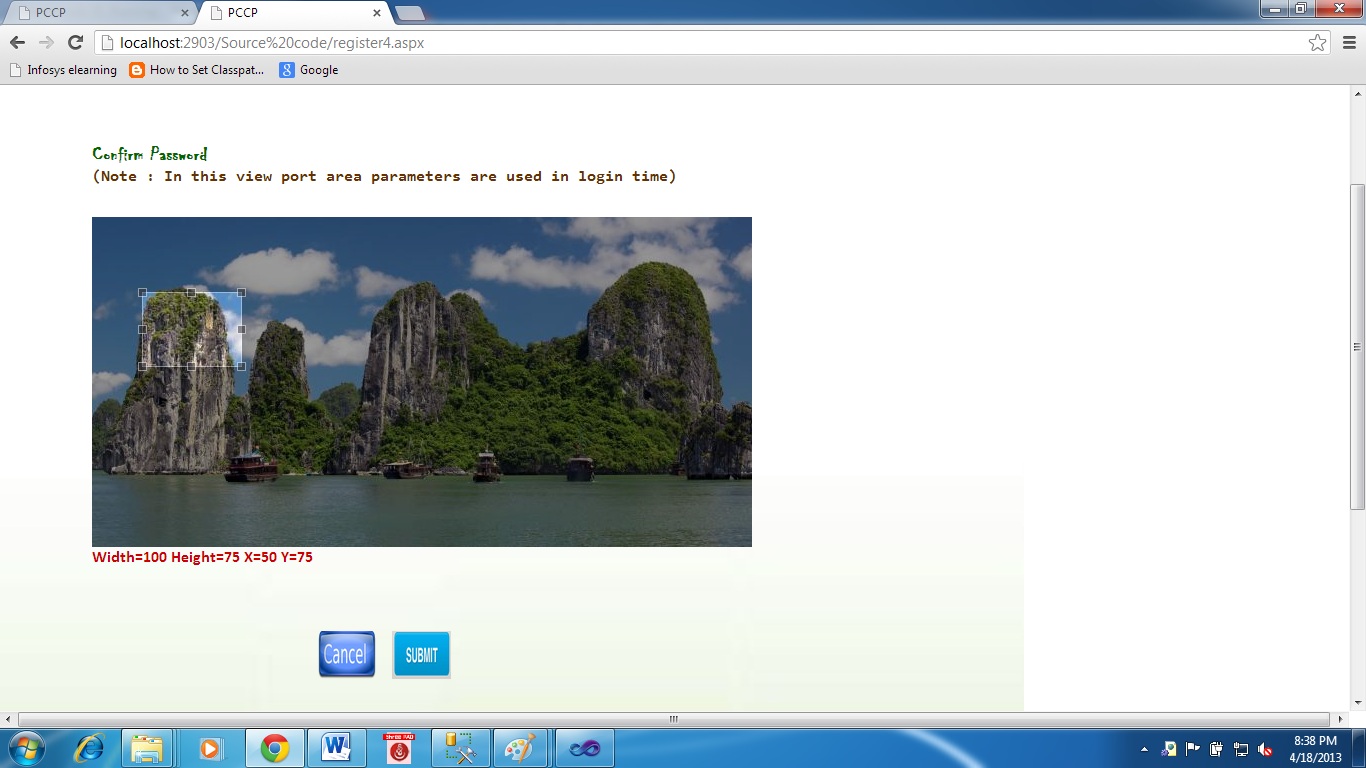
6.8 Registering using click points module using single click on an image(image5)



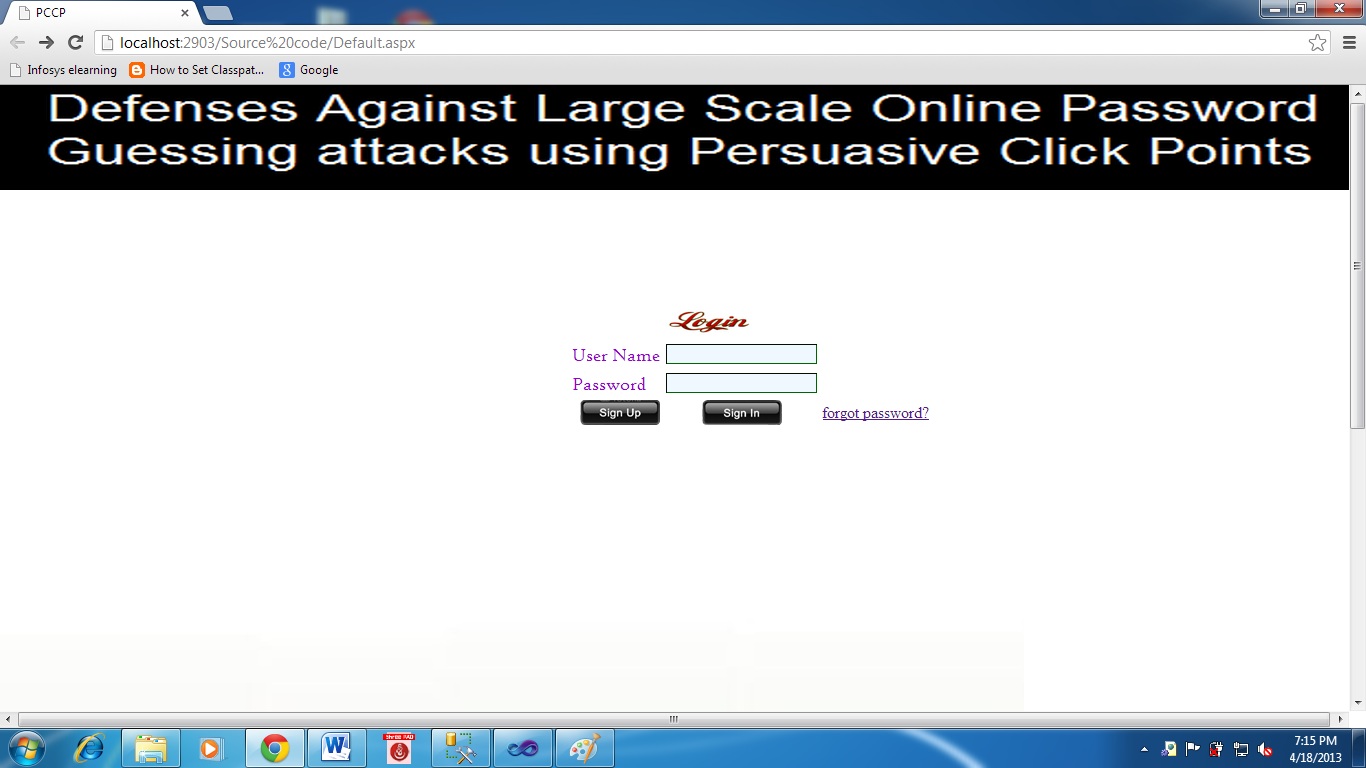
6.9.Displays the first image in the sequence



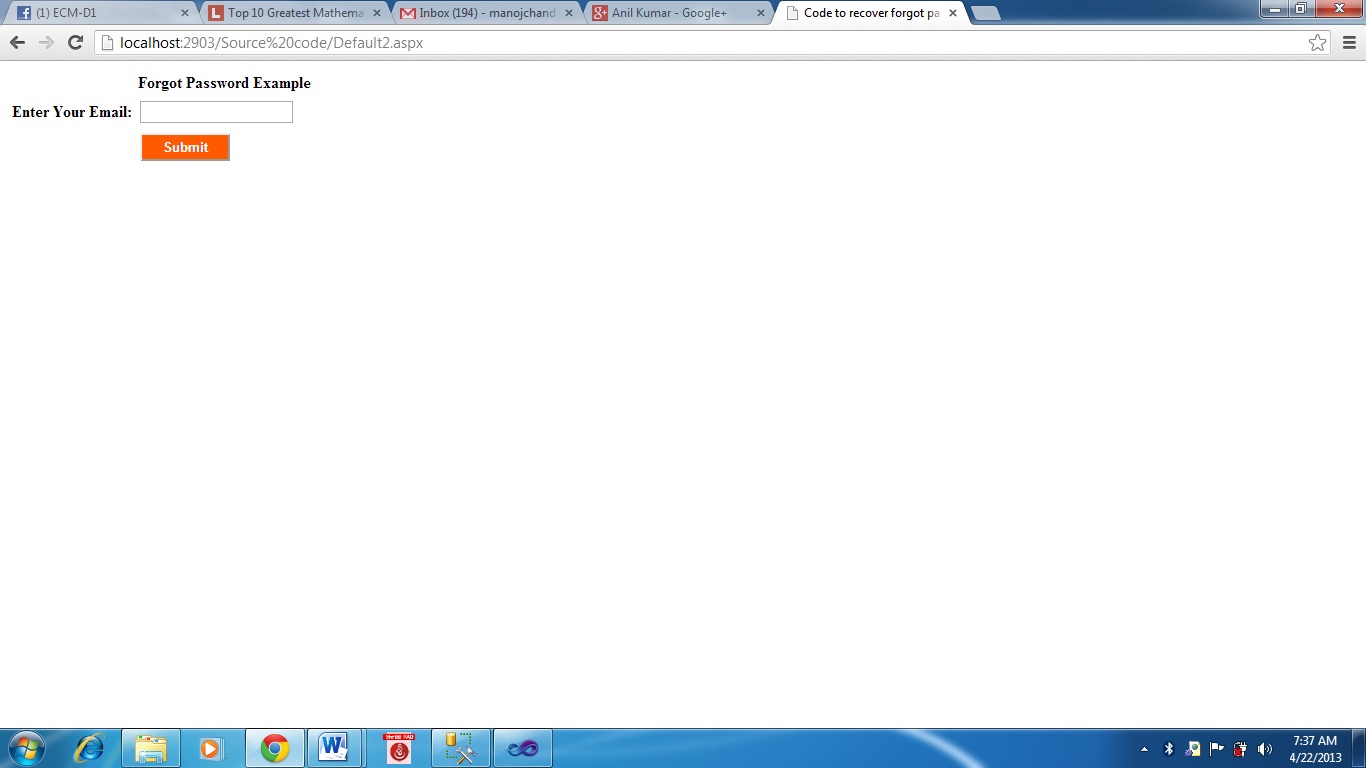
6.10 Selecting the view port area



6.11 Enter the username and password and sign in



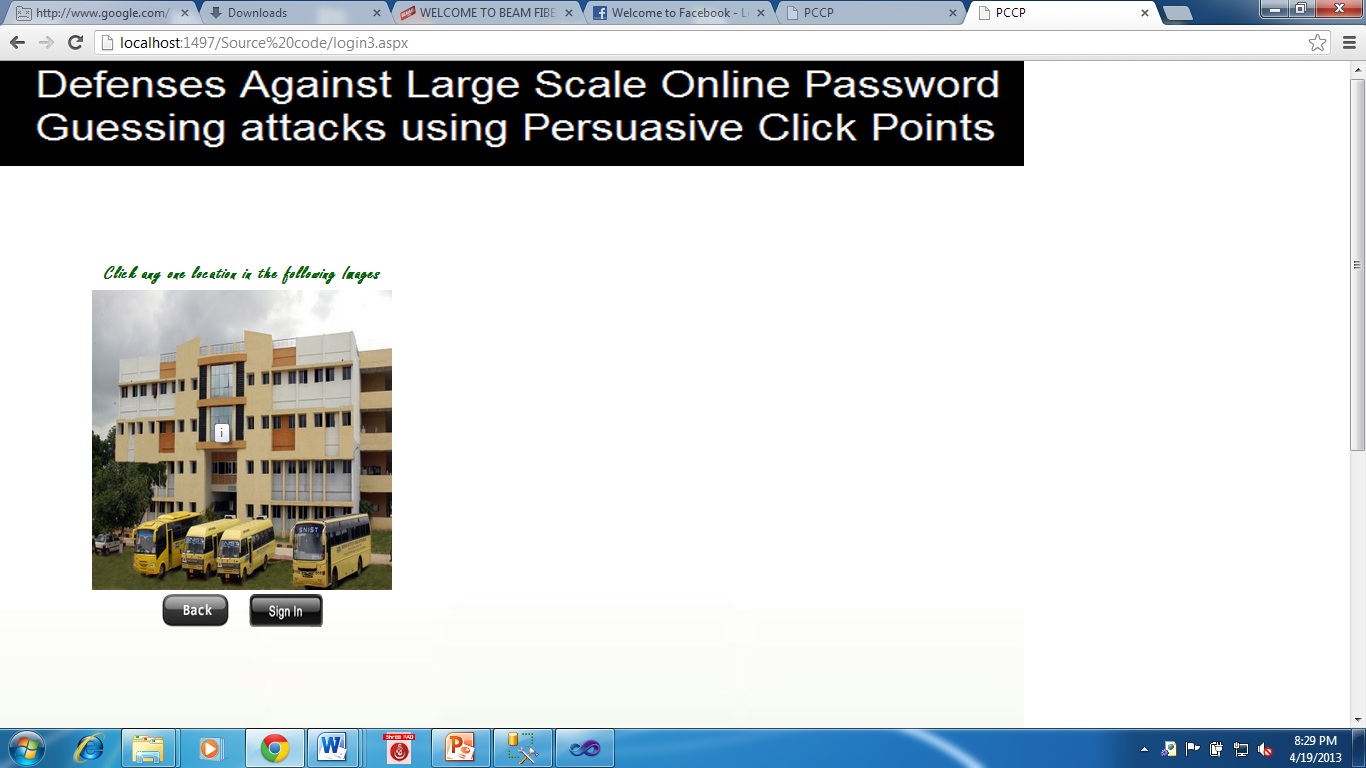
6.12 When a user forgets his password



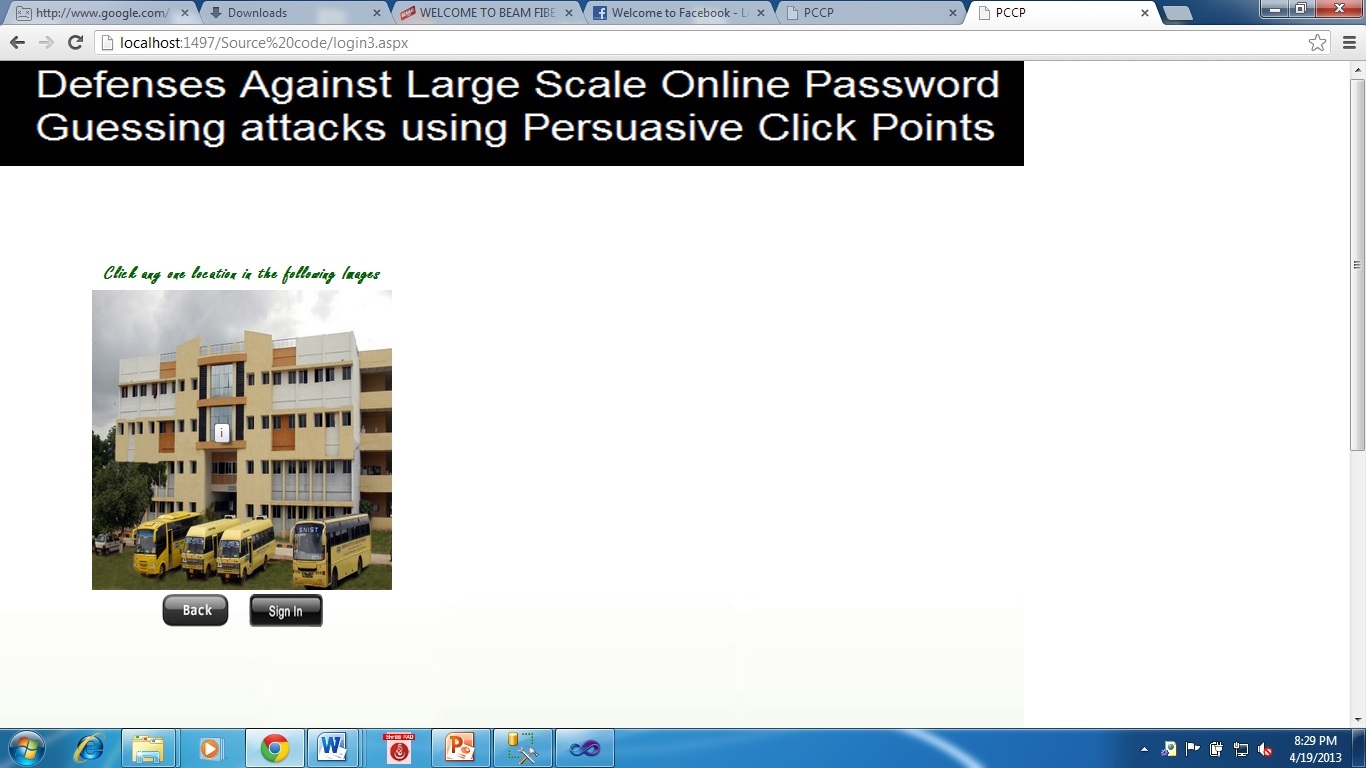
6.13 Signing in clicking on 5 places in the image



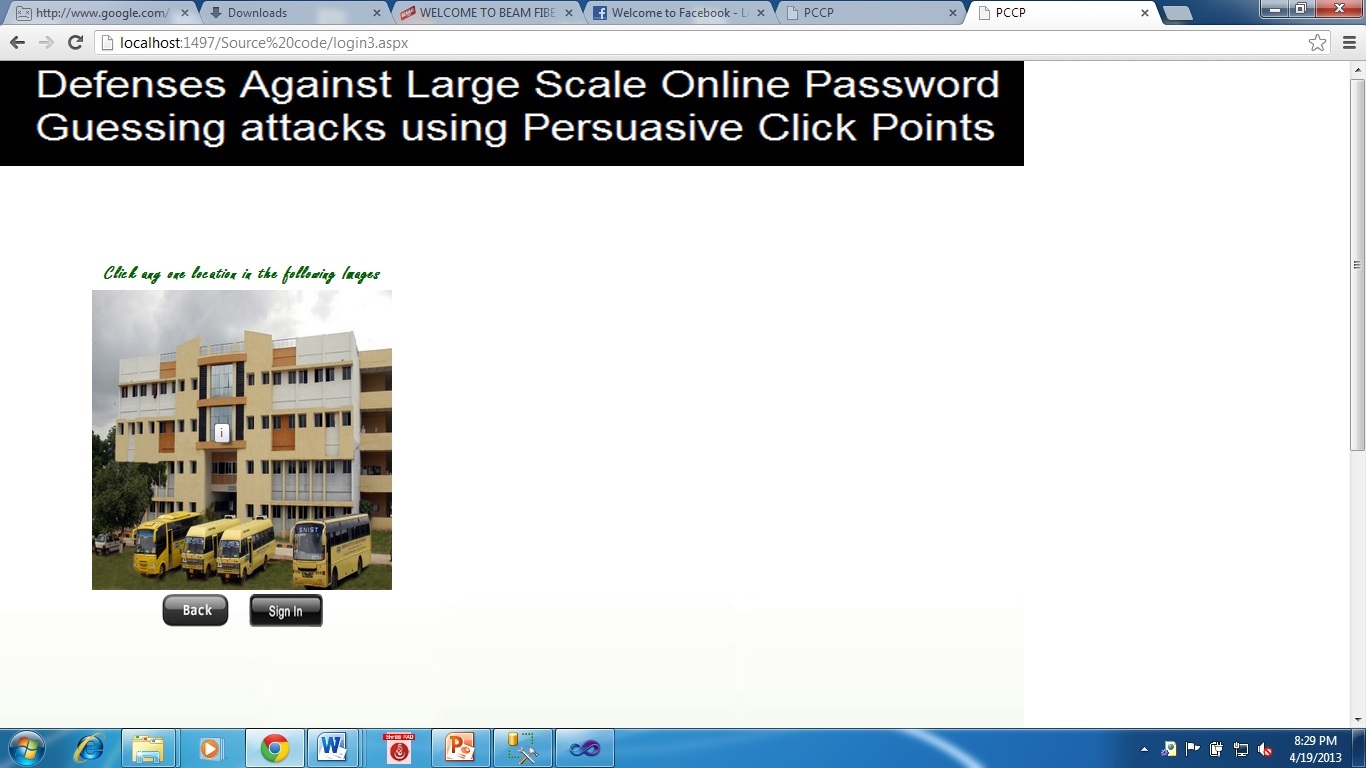
6.14 Clicking on the same click point for the first image in click points module



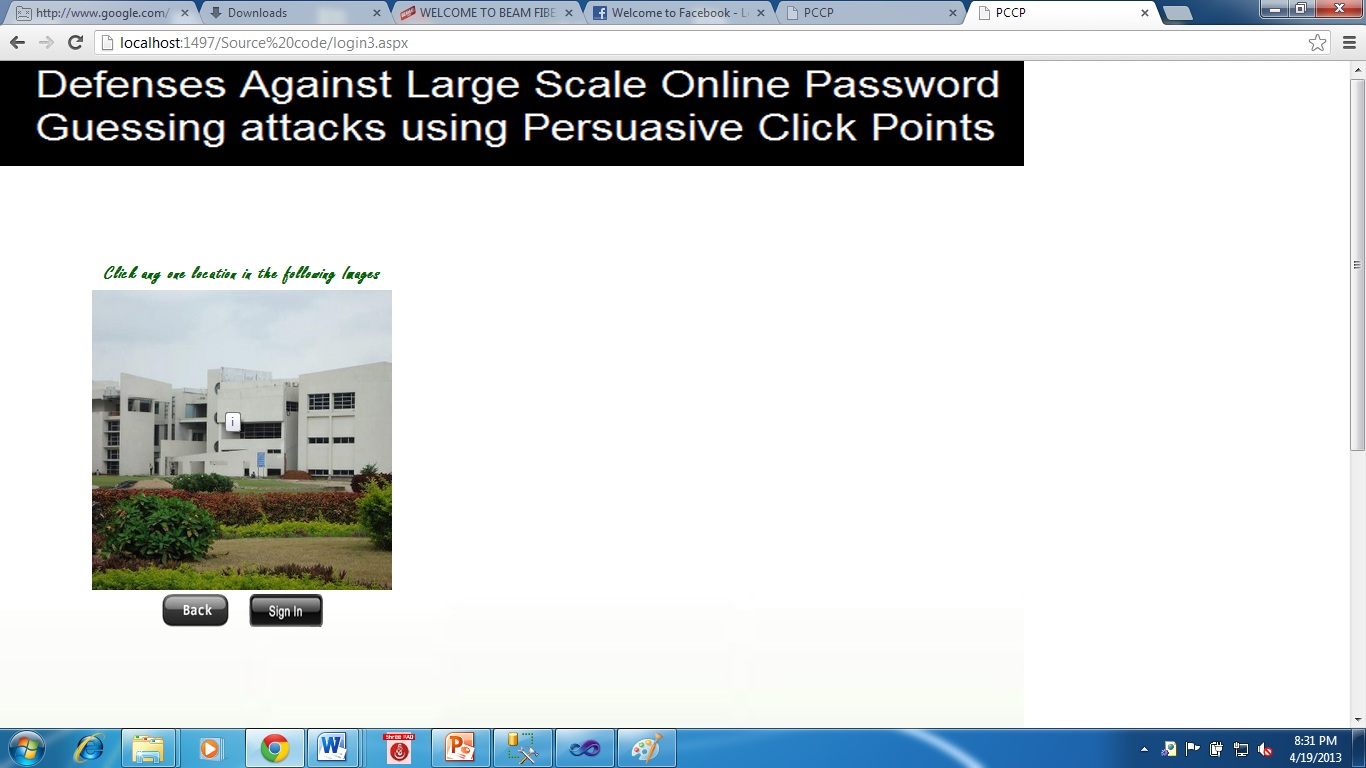
6.15 Clicking on the same click point for the second image in click points module



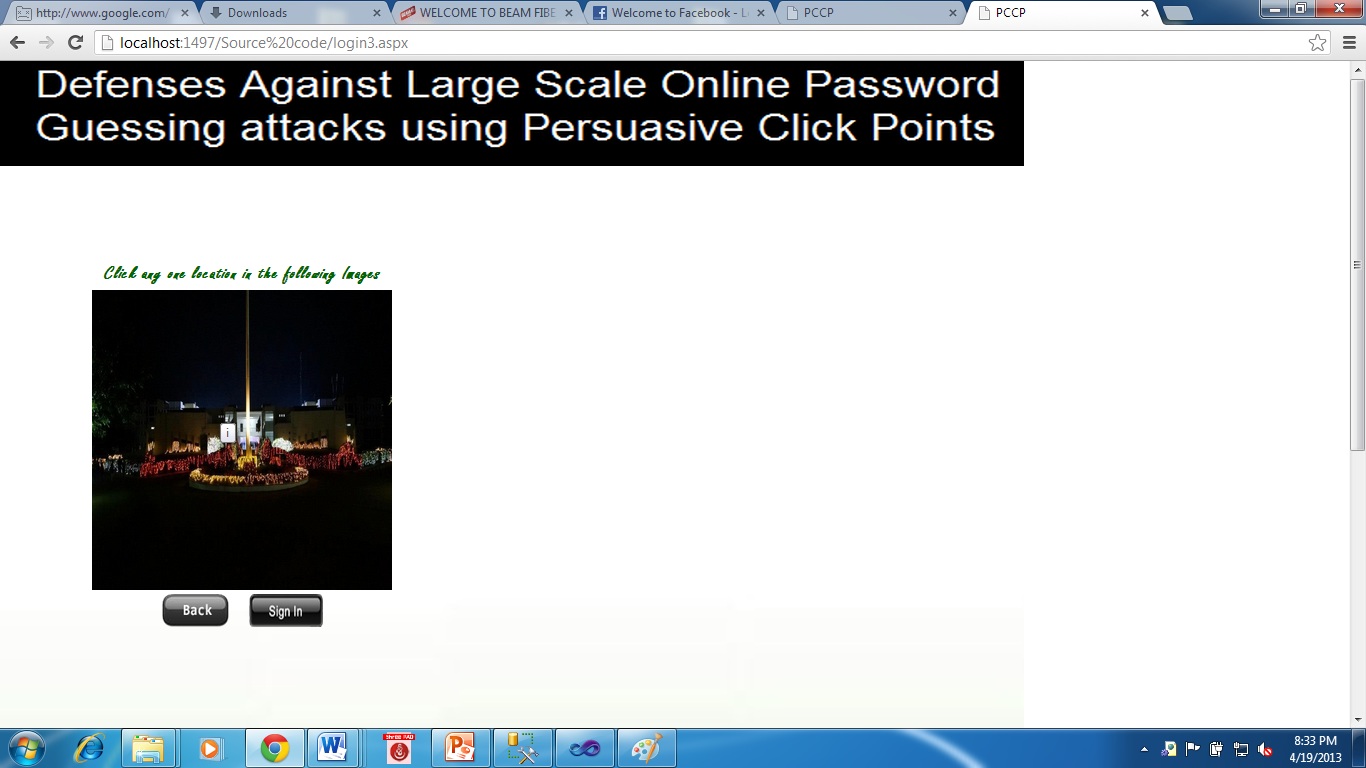
6.16 Clicking on the same click point for the second image in click points module



6.17 Clicking on the same click point for the third image in click points module



6.18 Clicking on the same click point for the fourth image in click points module



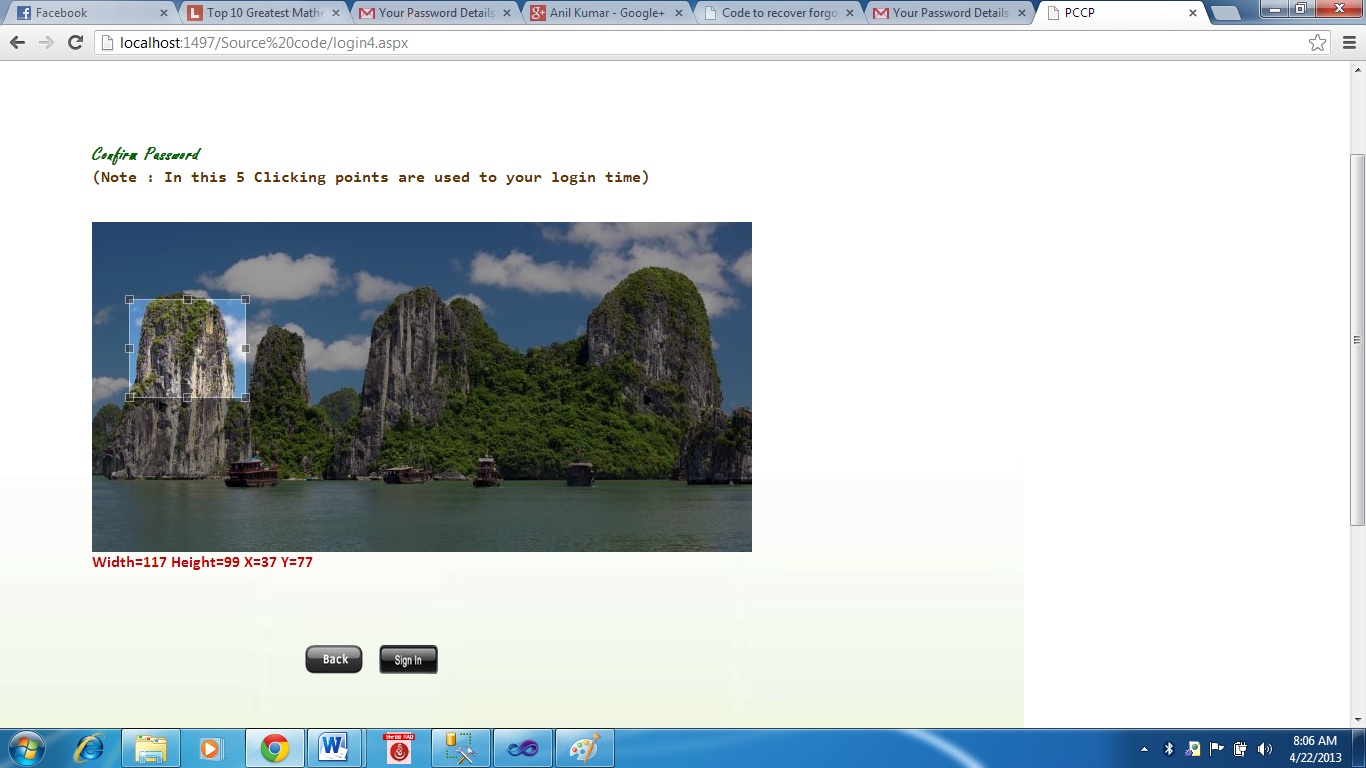
6.19 Clicking on the same click point for the fifth image in click points module



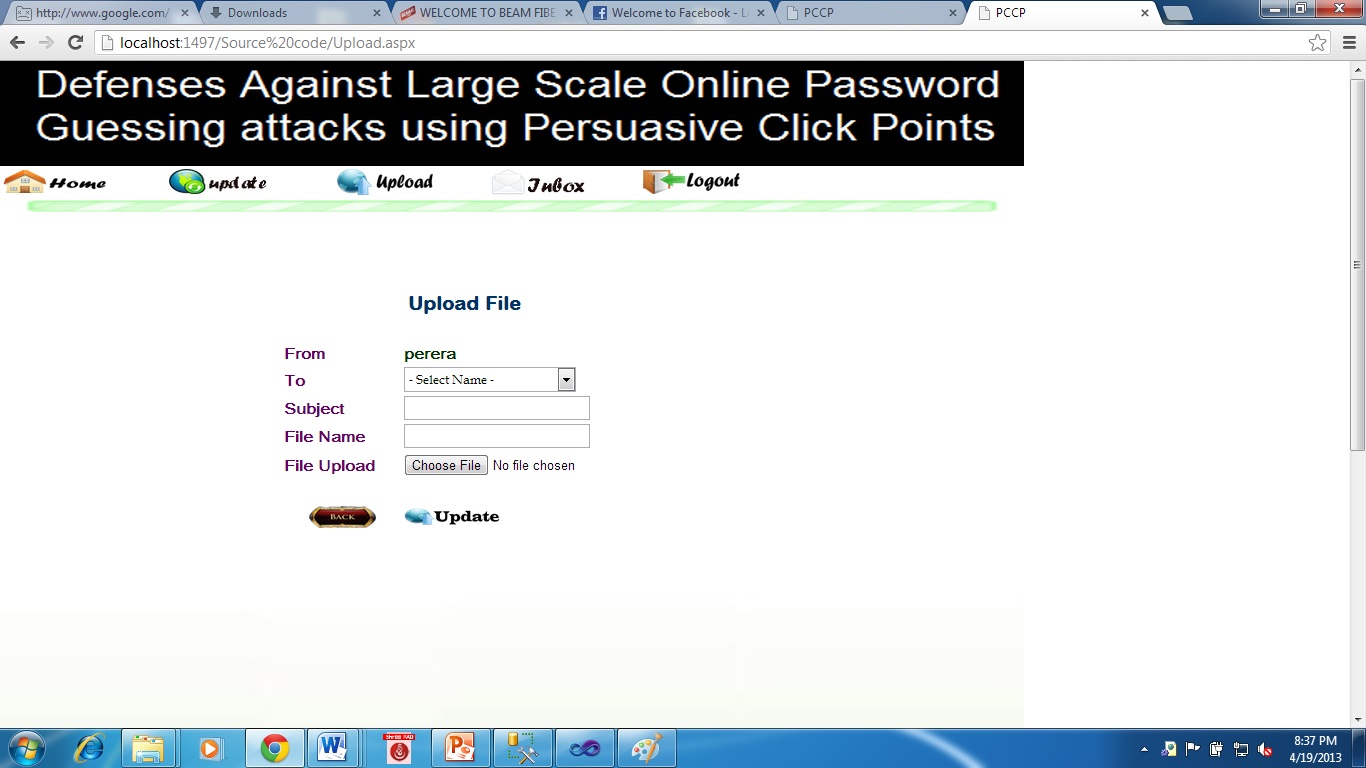
6.20 Displays the first image for confirmation



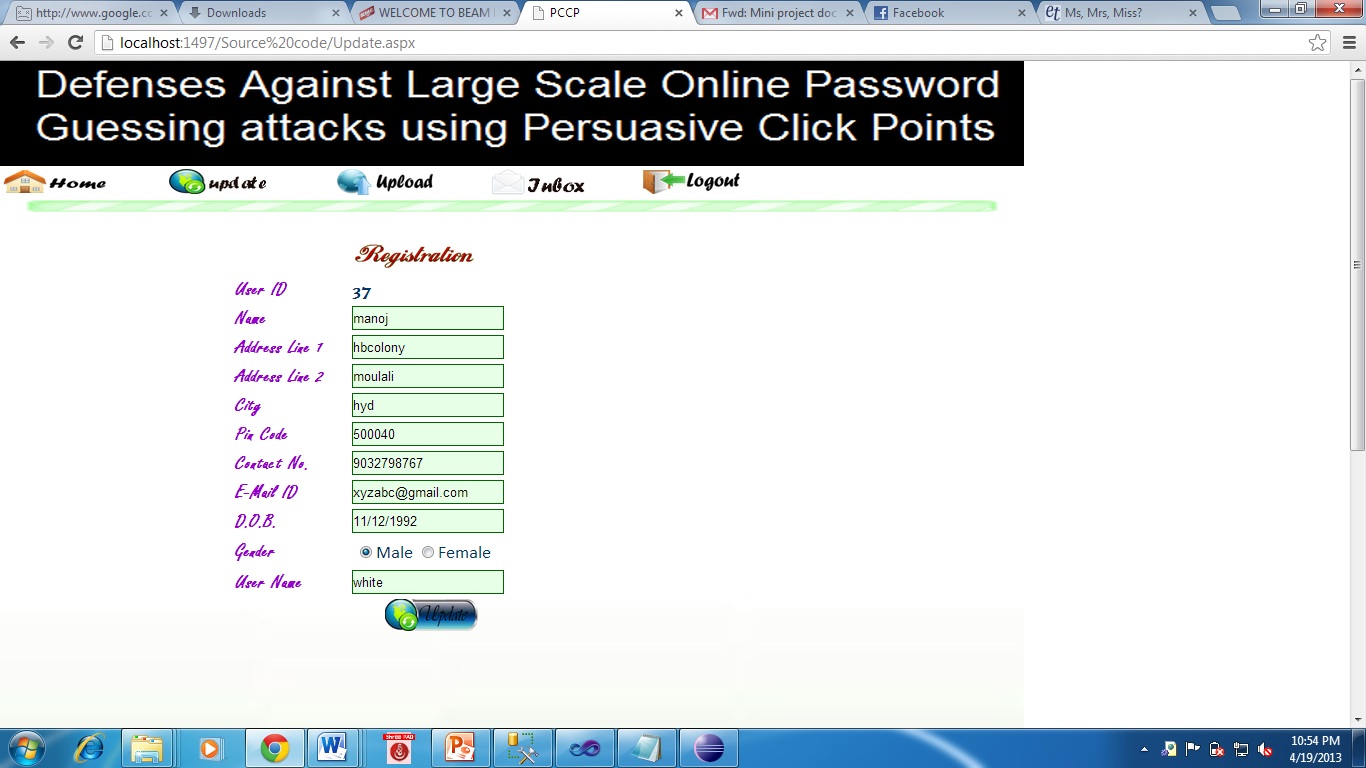
6.21 Select the same view port to finally login to your account



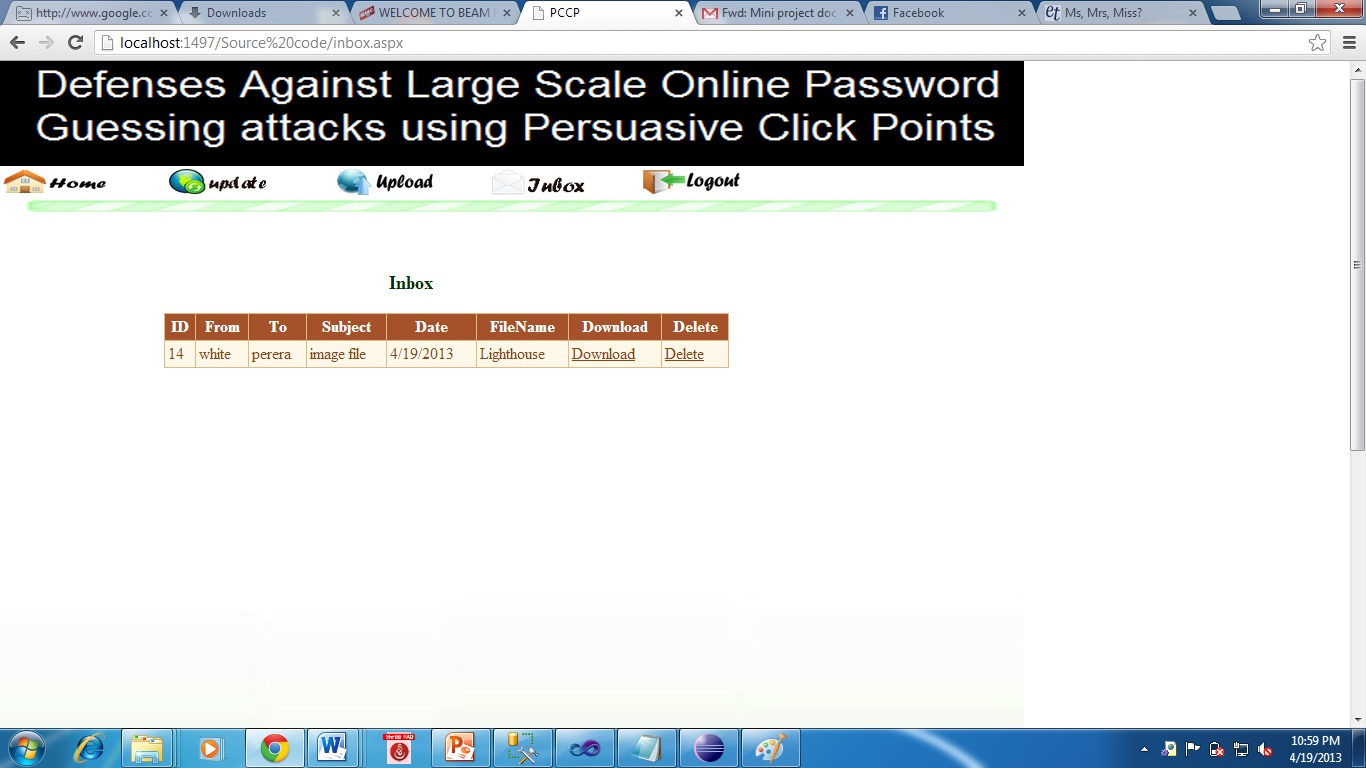
6.22 Upload option allows user to send a file to another user



6.23 Update option allows user to update the registered values



* 1. User can check his inbox which shows files sent by other users.



7. SYSTEM TESTING

7.1. INTRODUCTION

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a wellplanned series of steps that result in the successful construction of software. Testing is the setof activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

7.2. STRATEGIC APPROACH TO SOFTWARE TESTING

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally we arrive at system testing, where the software and other system elements are tested as a whole..

UNIT TESTING

MODULE TESTING

SUB-SYSTEM TESING

SYSTEM TESTING

ACCEPTANCE TESTING

Component Testing

Integration Testing

User Testing

7.3. Unit Testing

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel.

*1. WHITE BOX TESTING*

This type of testing ensures that

* All independent paths have been exercised at least once
* All logical decisions have been exercised on their true and false sides
* All loops are executed at their boundaries and within their operational bounds
* All internal data structures have been exercised to assure their validity.

To follow the concept of white box testing we have tested each form .we have created independently to verify that Data flow is correct, All conditions are exercised to check their validity, All loops are executed on their boundaries.

*2. BASIC PATH TESTING*

Established technique of flow graph with Cyclomatic complexity was used to derive test cases for all the functions. The main steps in deriving test cases were:

Use the design of the code and draw correspondent flow graph.

Determine the Cyclomatic complexity of resultant flow graph, using formula:

V(G)=E-N+2 or

V(G)=P+1 or

V(G)=Number Of Regions

Where V(G) is Cyclomatic complexity,

E is the number of edges,

N is the number of flow graph nodes,

P is the number of predicate nodes.

Determine the basis of set of linearly independent paths.

*3. CONDITIONAL TESTING*

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generate on particular condition is traced to uncover any possible errors.

*4. DATA FLOW TESTING*

This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variable were declared. The definition-use chain method was used in this type of testing. These were particularly useful in nested statements.

*5. LOOP TESTING*

In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops:

* All the loops were tested at their limits, just above them and just below them.
* All the loops were skipped at least once.
* For nested loops test the inner most loop first and then work outwards.
* For concatenated loops the values of dependent loops were set with the help of connected loop.
* Unstructured loops were resolved into nested loops or concatenated loops and tested as above.

Each unit has been separately tested by the development team itself and all the input have been validated.

7.4 Test Cases

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case | Check Item | Test Case Objective | Steps to Execute | Test Data / Input | Expected Result | Actual Result | Result |
| 1 | Sign up button | To check whether registration page is displayed or not | Click on sign up button |  | Registration page must be displayed | Registration page is displayed | Pass |
| 2 | Confirm button | To check whether required fields are filled or not | Click on confirm button | Empty fields | Error messages must be displayed near required fields | Error messages displayed near required fields | Pass |
| 3 | Date of birth verification | To check whether entered date of birth is valid or not | Enter date of birth | Giving invalid date of birth Ex:14-03-2014 | Error: Invalid date of birth must be displayed | Invalid date of birth error is displayed | Pass |
| 4 | Mobile no. verification | To check whether entered mobile number is valid or not | Enter mobile number | Giving invalid number Ex:44444444 | Error: Invalid mobile number must be displayed | Invalid mobile number error is displayed | Pass |
| 5 | E-mail id verification | To check whether entered E-mail id is valid or not | Enter E-mail id | Giving invalid E-mail id Ex:sandy@m.com | Error: Invalid E-mail id must be displayed | Invalid E-mail id error is displayed | Pass |
| 6 | Password verification | To check whether password contains minimum of 5 characters or not | Enter password | Giving characters <5 | Error: Minimum 5 characters must be displayed | Minimum 5 characters | Pass |
| Test Case | Check Item | Test Case Objective | Steps to Execute | Test Data / Input | Expected Result | Actual Result | Result |
| 7 | Confirm button | To check whether next page of registration is displayed or not | click on confirm button | Enter valid data in all fields | Next page of registration must be displayed | Next page of registration is displayed | Pass |
| 8 | Upload button | To check whether chosen image is uploaded or not | Click on upload button | Choose an image >200 kb | Error: choose a smaller size image must be displayed | Choose a smaller size image is displayed | Pass |
| 9 | Upload button | To check whether chosen image is uploaded or not | Click on upload button | Choose image file from system | Image must be displayed on image button | Image is displayed | Pass |
| 10 | Use Pre-defined button | To check whether Pre-defined image is displayed or not. | Click on use pre-defined image |  | Image must be displayed on image button | Image is displayed | Pass |
| 11 | Click points | When no .of click points are more than 5 | Give clicks on image | 6 clicks are given | Error: Required clicks are completed must be displayed | Required clicks are completed message is displayed | Pass |
| 12 | Confirm button | To check whether required clicks are given or not | Click on confirm button | 3 clicks are given | Error:2 more click points are required must be displayed | 2 more click points are required message is displayed. | Pass |
| 13 | Confirm button | Navigate to next level of registration | click on confirm button | 5 clicks are given | Next page of registration must be displayed | Next page of registration is displayed | Pass |
| Test Case | Check Item | Test Case Objective | Steps to Execute | Test Data / Input | Expected Result | Actual Result | Result |
| 14 | Next button | To check whether selected no. of images are displayed or not | select no. of images | 5 images option is selected | 5 images should be displayed in a sequence | 5 images are displayed in a sequence | Pass |
| 15 | Confirm button | To check whether last level of registration is displayed or not | click on confirm button | 5 characters are chosen | Last level registration page should be displayed | Last level registration page is displayed. | Pass |
| 16 | Sign in button | To check whether valid username and password are given or not | Click on sign in button | Enter invalid username and password | Error message must be displayed and forgot password should become visible | Error message is displayed and forgot password link is visible | Pass |
| 17 | Sign in button | Navigate to level 2 login page | Click on sign in button | Enter valid username and password | Level 2 login page should be displayed | Level 2 login page is displayed | Pass |
| 18 | Sign in button | Navigate to level 3 login page | Click on sign in button | Incorrect clicks are given | Error: More 2 chances are there to sign in and forget password label must be displayed | Error message and forgot password label is displayed | Pass |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case | Check Item | Test Case Objective | Steps to Execute | Test Data / Input | Expected Result | Actual Result | Result |
| 19 | Sign in button | Navigate to level 3 login page | Click on sign in button | Incorrect clicks are given more than 3 times | User should be redirected to first stage of login process | User redirected to first stage of login process | Pass |
| 20 | Sign in button | Navigate to level 3 login page | Click on sign in button | Correct clicks are given | Level 3 login page should be displayed | Level 3 login page is displayed | Pass |
| 21 | Sign in button | Navigate to level 4 login page | Click on sign in button | Correct clicks on characters are given | Level 4 login page should be displayed | level 4 login page is displayed | Pass |
| 22 | Sign in button | To enter into the website | Click on sign in button | Incorrect view port is selected | Error: More 2 chances are there to sign in, forget password label and Incorrect view port is selected must be displayed | Error messages and forgot password label are displayed | Pass |
| 23 | Sign in button | To enter into the website | Click on sign in button | Correct view port is selected | Website page should be displayed | Website page is displayed | Pass |
| 24 | Upload option | Navigate to upload page | Click on upload option |  | Upload page should be displayed | Upload page is displayed | Pass |
| Test Case | Check Item | Test Case Objective | Steps to Execute | Test Data / Input | Expected Result | Actual Result | Result |
| 25 | Updateoption | Navigate to Update page | Click on Update option |  | Update page should be displayed | Update page is displayed | Pass |
| 26 | Upload button | To check whether file is uploadedor not | Click on upload button | Fill valid details in upload page | Uploaded successfully message should be displayed | Uploaded successfully message is displayed. | Pass |
| 27 | Inbox option | Navigate to Inbox page | Click on inbox option |  | Received files / no date here message should be displayed | Received files are displayed. | Pass |
| 28 | Download link | To check whether required file is downloaded or not | Click on download link |  | Required file should be downloaded | Required file is downloaded. | Pass |
| 29 | Delete link | To check whether file is deleted or not | Click on delete link |  | File should be deleted | File is deleted. | Pass |
| 30 | Forgot password link | Navigate to forgot password page | Click on forgot password link |  | Forgot password page should be displayed | Forgot password page is displayed. | Pass |
| 31 | Submit button | To check whether password details are sent to the mail or not | Click on submit button option | Incorrect security answer is given | Enter correct answer message should be displayed | Enter correct answer message is displayed. | Pass |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case | Check Item | Test Case Objective | Steps to Execute | Test Data / Input | Expected Result | Actual Result | Result |
| 32 | Submit button | To check whether password details are sent to the mail or not | Click on  Submit button | Correct security answer | Password details sent to your mail message should be displayed. | Password details sent to your mail message is displayed. | Pass |
| 33 | Logout option | To check whether user is logged out or not. | Click on logout option |  | Initial login page should be displayed. | Initial login page is displayed. | Pass |

1. CONCLUSION

A major advantage of Persuasive cued click point scheme is its large password space over alphanumeric passwords. There is a growing interest for Graphical passwords since they are better than Text based passwords, although the main argument for graphical passwords is that people are better at memorizing graphical passwords than text-based passwords. Online password guessing attacks on password-only systems have been observed for decade’s .Present-day attackers targeting such systems are empowered by having control of thousand to million node botnets. In previous ATT-based login protocols, there exists a security-usability trade-off with respect to the number of free failed login attempts (i.e., with no ATTs) versus user login convenience (e.g., less ATTs and other requirements). In contrast, PGRP is more restrictive against brute force and dictionary attacks while safely allowing a large number of free failed attempts for legitimate users. PGRP is apparently more effective in preventing password guessing attacks (without answering ATT challenges), it also offers more convenient login experience, e.g., fewer ATT challenges for legitimate users. PGRP appears suitable for organizations of both small and large number of user accounts.

9. BIBLIOGRAPHY

Good Teachers are worth more than thousand books, we have them in Our Department

References Made From:

1. **Beginning ASP.NET 4: in C# and VB** by Imar Spaanjaars.
2. [**Programming ASP.NET 3.5**](http://www.amazon.com/gp/product/0596529562?ie=UTF8&tag=aspnettelligent-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=0596529562) by Jesse Liberty, Dan Maharry, Dan Hurwitz.
3. [**Beginning ASP.NET 3.5 in C# 2008: From Novice to Professional, Second Edition**](http://www.amazon.com/gp/product/1590598911?ie=UTF8&tag=aspnettelligent-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=1590598911) by Matthew MacDonald.
4. Sonia Chiasson, P.C. van Oorschot, and Robert Biddle, “Graphical Password Authentication Using Cued Click Points” ESORICS, LNCS 4734, pp.359- 374,Springer- Verlag Berlin Heidelberg 2007.
5. Manu Kumar, Tal Garfinkel, Dan Boneh and Terry Winograd, “Reducing Shoulder-surfing by Using Gazebased Password Entry”, Symposium On Usable Privacy and Security (SOUPS) , July 18-20, 2007, Pittsburgh,PA, USA.
6. Zhi Li, Qibin Sun, Yong Lian, and D. D. Giusto, „An association-based graphical password design resistant to shoulder surfing attack‟, International Conference on Multimedia and Expo (ICME), IEEE.2005
7. R. Dhamija and A. Perrig, "Deja Vu: A User Study Using Images for Authentication," in Proceedings of9th USENIX Security Symposium, 2000.
8. S. Akula and V. Devisetty, "Image Based Registration and Authentication System," in Proceedings ofMidwes Instruction and Computing Symposium, 2004.
9. L. Sobrado and J.-C. Birget, "Graphical passwords," The Rutgers Scholar, An Electronic Bulletin forUndergraduate Research, vol. 4, 2002.
10. Sonia Chiasson, Alain Forget, Robert Biddle, P. C. van Oorschot, “User interface design affects security: patterns in click-based graphical passwords”, Springer-Verlag 2009.