1. INTRODUCTION

1.1 MOTIVATION

Our application uploads the file using the ftp details of the cloud account, when user drags the file into the cloud storage. At the same time it's also provides any new files uploaded into the cloud folder of organisation to the admin of the site.

1.2 PROBLEM DEFINATION

If we want to upload the data into cloud server, we need to login to the site and browser and upload the files, user of the site need to login, and check what all the folder created or files uploading in the cloud account of an organization. By analyzing this problem statement, we thought of giving problem solution as follows:

- What if, if we can upload the file directly by dragging the file instead of browsing and uploading.

- What if, if we can provide messaging or mailing system to admin of the site to give any updates when any changes done in the cloud account of organization.

So to provide solution to the problem statement what we have discussed, we are developing an online application known as "Flexible file transfer protocol to amazon s3" where user can upload the file directly by dragging the file and admin can get the instant updates through mail or message.

1.3 OBJECTIVE OF PROJECT

This project is mainly designed to maintain all the work done by each employee in a company on cloud and retrieve the information when ever needed, our application can be used by the users or employees of organization user can easily move the files or folders onto cloud folder from the local system.

1.4 ORGANISATION OF DOCUMENTATION

In this project documentation we have initially put the definition and objective of the project as well as the design of the project which is followed by the implementation and testing phases. Finally the project has been concluded successfully and also the future enhancements of the project were given in this documentation.

2. LITERATURE SURVEY

2.1 INTRODUCTION

Cloud FTP is a model for enabling convenient, where user can upload the file and admin can get the instant updates through mail. Our application uploads the file using ftp details of the cloud storage. At the same time it also provides any new files uploaded into the cloud folder of organization to the admin of site.

This project is mainly designed to drag and drop facility to dump the data into cloud folder to provide easy access, sending the email to the current administrator for the changes done in the cloud by using web services the functionality has been implemented. And will update the progress each instant on the list view.

2.2 EXISTING SYSTEM

In the existing system we used the open the site and update the data on the cloud. We do like creating new folder and upload the files one by one it’s a time consuming process. Instead we need a new way of updating the data on cloud.

2.3 DISADVANTAGES OF EXISTING SYSTEM

- Time consuming process.
- Less flexibility
- Less efficiency

2.4 PROPOSED SYSTEM

In the proposed we prepare a windows tool where we use ftp details of cloud folder which we have created on cloud .And we prepare an interface where we can easily move the files or folders onto cloud folder from the local system.

2.5 ADVANTAGES OF PROPOSED SYSTEM

- Very less time consuming process to upload data.
- Flexible to create, upload and delete the files/folders.
- High-End security for users.
- Easy Maintenance of data by users.
- Durable and cost effective.

2.6 CONCLUSION

This project is mainly used to drag and drop facility to dump the data into cloud folder to provide easy access, sending the email to the current administrator for the changes done in the cloud by using web services the functionality has been implemented.

3. ANALYSIS

3.1 INTRODUCTION

After analyzing the requirements of the task to be performed, the next step is to analyze the problem and understand its context. The first activity in the phase is studying the existing system and other is to understand the requirements and domain of the new system. Both the activities are equally important but the first activity serves as a basis of giving the functional specifications and then successful design of the proposed system. Understanding the properties and requirements of a new system is more difficult and requires creative thinking as well as understanding of existing system is also difficult. Improper understanding of present system can lead diversion from solution.

3.1.1 ANALYSIS MODEL

The model that is basically being followed is water fall model which states that the phases are organized in a linear order. First of all, the feasibility study is done. Once that part is over, the requirement analysis and project planning begins. If system exists as a whole but modification and addition of new module is needed, analysis of present system can be used as basic model. The design starts after the requirement analysis is complete and the coding begins after the design is complete. Once the programming is completed, the testing is done. In this model the sequence of a Requirement Analysis Activities performed in a software development project are:

- Project Planning
- System Design
- Detail Design
- Coding
- Unit Testing
- System Integration & Testing

Here the linear ordering of these activities is critical. At the end of the phase, the output of one phase is the input to other phase. The output of each phase should be consistent with the overall requirement of the system. Some of the qualities of spiral model are also incorporated like after the people concerned with the project review completion of each of the phase the work done. Water fall model has been chosen because all requirements were known before and the objective of our software development is the computerization/automation of an already existing manual working system.

ADVANTAGES:

1. This model is very easy to use and implement.

2. Each phase is completed at a time and processed.

3. This model better works for smaller projects if only the requirements are well understood.

DISADVANTAGES:

1. If the requirements are gathered are inaccurate then the final product is inaccurate and the error is known in the final phase of the model. Any sort of errors that cannot be detected in any previous phase.

2. For long, object-oriented, complex and ongoing projects it’s a poor model.

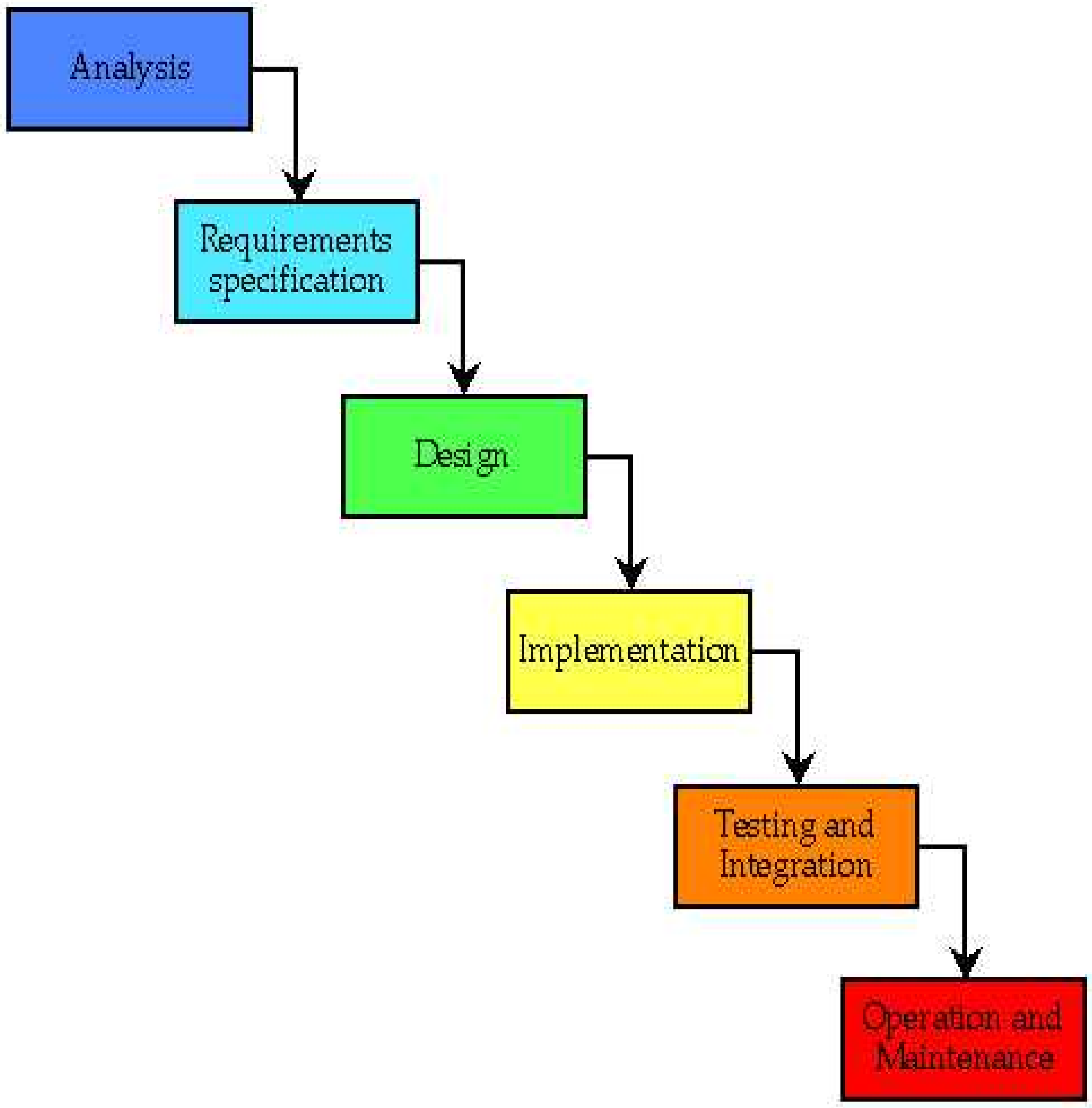


Fig 3.1.1 Waterfall Lifecycle Model

PROTOTYPE MODEL:

In this model the requirements are gathered firstly, and the prototype is deployed according to the requirements. This prototype is a quick design which goes through the coding, design and testing. The phases are not done in detail. By seeing this prototype the client feels like a real system, so that the client understands the entire requirements of the systems.

ADVANTAGES:

1. During the development process the developers are interestingly engaged.

2. The prototype developed that is used by the users for well understanding of the methodology.

3. The user involvement is increased and improved.

4. The flaws and faults are identified early.

5. The user's opinion about the product is known early which leads to an improved system.

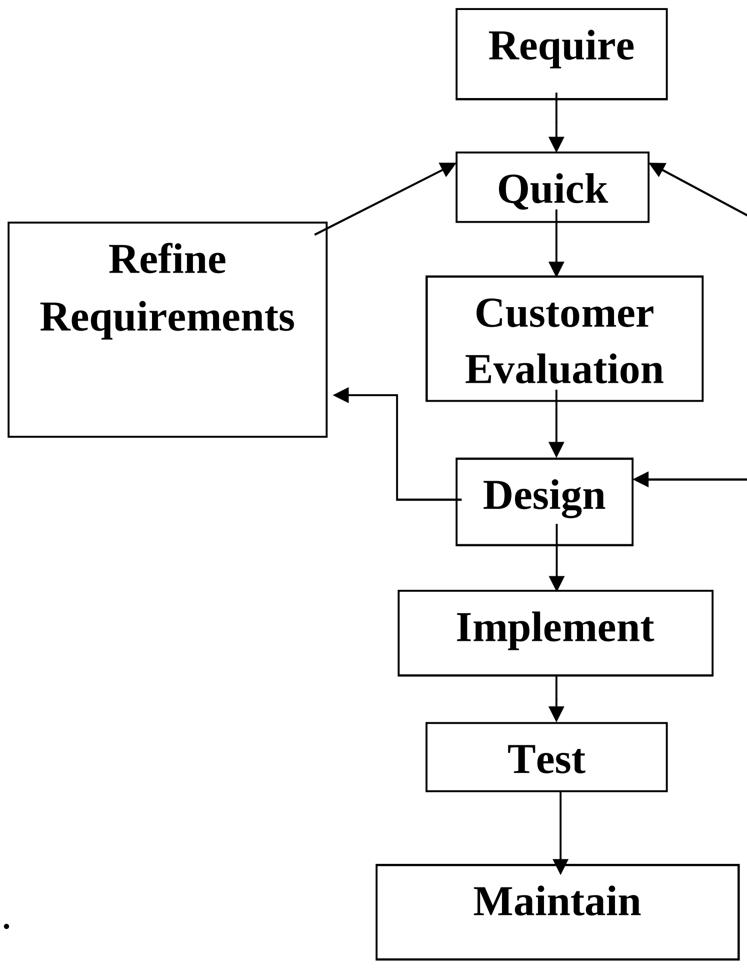
DISADVANTAGES:

1. This model focuses on design quite than functionality.

2. The model is implemented firstly and then errors are evaluated later which becomes a complex process.

3. The model is also known as throw-away prototype.

4. More time spent on development of the prototype that result in delay of the final product.



.

3.1.2 Feasibility Report

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

3.1.2.1: Technical Feasibility

3.1.2.2: Operation Feasibility

3.1.2.3: Economical Feasibility

3.1.2.1 Technical Feasibility**:**

Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, at this point in time, not too many detailed design of the system, making it difficult to access issues like performance, costs on (on account of the kind of technology to be deployed) etc. A number of issues have to be considered while doing a technical analysis.

Understand the different technologies involved in the proposed system:

Before commencing the project, we have to be very clear about what are the technologies that are required for the development of the new system.

3.1.2.2 Operational Feasibility:

Proposed projects are beneficial only if they can be turned into information systems that will meet the organizations operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed. Are there major barriers to Implementation? Here are questions that will help test the operational feasibility of a project:

Is there sufficient support for the project from management from users? If the current system is well liked and used to the extent that persons will not be able to see reasons for change, there may be resistance.

Are the current business methods acceptable to the user? If they are not, Users may welcome a change that will bring about a more operational and useful systems.

Have the user been involved in the planning and development of the project? Early involvement reduces the chances of resistance to the system and in general and increases the likelihood of successful project. Since the proposed system was to help reduce the hardships encountered. In the existing manual system, the new system was considered to be operational feasible.

3.1.3 Economic Feasibility:

Economic feasibility attempts 2 weigh the costs of developing and implementing a new system, against the benefits that would accrue from having the new system in place. This feasibility study gives the top management the economic justification for the new system.

A simple economic analysis which gives the actual comparison of costs and benefits are much more meaningful in this case. In addition, this proves to be a useful point of reference to compare actual costs as the project progresses. There could be various types of intangible benefits on account of automation. These could include increased customer satisfaction, improvement in product quality better decision making timeliness of information, expediting activities, improved accuracy of operations, better documentation and record keeping, faster retrieval of information, better employee morale.

3.2 SOFTWARE REQUIREMENT SPECIFICATION

3.2.1 USER REQUIREMENT

- Computer

3.2.2 SOFTWARE REQUIREMENTS

- Microsoft .Net framework 3.5
- Microsoft Visual Studio 2008
- Microsoft C#.Net language
- Cloud Computing

3.2.3 HARDWARE REQUIREMENTS

• Processor	:	Intel Pentium 4 or more
• Ram	:	1 GB or more
• Hard disk	:	40 GB hard disk recommended for primary partition

3.3 CONCLUSION

In this phase, we understand the software requirement specifications for the project. We arrange all the required components to develop the project in this phase itself so that we will have a clear idea regarding the requirements before designing the project. Thus we will proceed to the design phase followed by the implementation phase of the project.

4. DESIGN

4.1 INTRODUCTION

The design phase begins with the requirements specification for the software to be developed. Design is the first step to moving from the problem domain towards the solution domain. Design is essentially the bridge between requirement specification and the final solution for satisfying the requirements. It is the most critical factor effecting the quality of the software. The design process for software system has two levels.

1. System Design or Top level design

2. Detailed Design or Logical design

1. System Design:

In the system design the focus on the deciding which modules are needed for the system, the specification of these modules and how these modules should be interconnected.

2. Detailed Design:

In detailed design the interconnection of the modules or how the specifications of the modules can be satisfied is decided. Some properties for a software system design are

- Verifiability
- Completeness
- Consistency
- Traceability
- Simplicity / Understandability

4.2 DFD/ER/UML DIAGRAMS:

UML DIAGRAMS:

Modelling is an activity that has been carried out over the years in software development. When writing applications by using the simplest languages to the most powerful and complex languages, you still need to model. Modelling can be as straightforward as drawing a flowchart listing the steps carried out by an application.

Why do we use modeling?

Defining a model makes it easier to break up a complex application or a huge system into simple, discrete pieces that can be individually studied. We can focus more easily on the smaller parts of a system and then understand the "big picture." Hence, the reasons behind modeling can be summed up in two words:

1. Readability:

Brings clarity—ease of understanding. Understanding a system is the first step in either building or enhancing a system. This involves knowing what a system is made up of, how it behaves, and so forth. Modelling a system ensures that it becomes readable and, most importantly, easy to document. Depicting a system to make it readable involves capturing the structure of a system and the behavior of the system.

2. Reusability:

Is the byproduct of making a system readable? After a system has been modelled to make it easy to understand, we tend to identify similarities or redundancy, be they in terms of functionality, features, or structure. The Unified Modelling Language, or UML, as it is popularly known by its TLA (three-letter acronym!), is the language that can be used to model systems and make them readable. This essentially means that UML provides the ability to capture the characteristics of a system by using notations. UML provides a wide array of simple, easy to understand notations for documenting systems based on the object-oriented design principles. These notations are called the nine diagrams of UML. Different languages have been used for depicting systems using object-oriented methodology. The prominent among these were the Rumbaugh methodology, the Booch methodology, and the Jacobson methodology.

The problem was that, although each methodology had its advantages, they were essentially disparate. Hence, if you had to work on different projects that used any of these methodologies, you had to be well versed with each of these methodologies. A very tall order indeed! The Unified Modelling Language is just that. It "unifies" the design principles of each of these methodologies into a single, standard, language that can be easily applied across the board for all object-oriented systems. But, unlike the different methodologies that tended more to the design and detailed design of systems, UML spans the realm of requirements, analysis, and design and, uniquely, implementation as well. The beauty of UML lies in the fact that any of the nine diagrams of UML can be used on an incremental basis as the need arises. Considering all these reasons, it is no wonder that UML is considered "the" language of choice. UML does not have any dependencies with respect to any technologies or languages. This implies that you can use UML to model applications and systems based on either of the current hot technologies; for example, J2EE and .NET. Every effort has been made to keep UML as a clear and concise modelling language without being tied down to any technologies.

INTRODUCTION TO UML:

The Unified Modelling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

GOALS OF UML:

The primary goals in the design of the UML were:

- Provide users with a ready-to-use, expressive visual modeling language so they can develop and exchange meaningful models.
- Provide extensibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development processes.
- Provide a formal basis for understanding the modeling language.

Why we use UML?

As the strategic value of software increases for many companies, the industry looks for techniques to automate the production of software and to improve quality and reduce cost and time-to-market. These techniques include component technology, visual programming, patterns and frameworks. Businesses also seek techniques to manage the complexity of systems as they increase in scope and scale. In particular, they recognize the need to solve recurring architectural problems, such as physical distribution, concurrency, replication, security, load balancing and fault tolerance. Additionally, the development for the World Wide Web, while making some things simpler, has exacerbated these architectural problems. The Unified Modelling Language (UML) was designed to respond to these needs.

UML Diagrams:

The underlying premise of UML is that no one diagram can capture the different elements of a system in its entirety. Hence, UML is made up of nine diagrams that can be used to model a system at different points of time in the software life cycle of a system. The nine UML diagrams are:

1. Use case diagram:

The use case diagram is used to identify the primary elements and processes that form the system. The primary elements are termed as "actors" and the processes are called "use cases." The use case diagram shows which actors interact with each use case.

2. Class diagram:

The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed "methods" of the class. Apart from this, each class may have certain "attributes" that uniquely identify the class.

3. Object diagram:

The object diagram is a special kind of class diagram. An object is an instance of a class. This essentially means that an object represents the state of a class at a given point of time while the system is running.

4. State diagram:

A state diagram, as the name suggests, represents the different states that objects in the system undergo during their life cycle. Objects in the system change states in response to events. In addition to this, a state diagram also captures the transition of the object's state from an initial state to a final state in response to events affecting the system.

5. Activity diagram:

The process flows in the system are captured in the activity diagram. Similar to a state diagram, an activity diagram also consists of activities, actions, transitions, initial and final states, and guard conditions.

6. Sequence diagram:

A sequence diagram represents the interaction between different objects in the system. The important aspect of a sequence diagram is that it is time-ordered. This means that the exact sequence of the interactions between the objects is represented step by step. Different objects in the sequence diagram interact with each other by passing "messages".

7. Collaboration diagram:

A collaboration diagram groups together the interactions between different objects. The interactions are listed as numbered interactions that help to trace the sequence of the interactions. The collaboration diagram helps to identify all the possible interactions that each object has with other objects.

8. Component diagram:

The component diagram represents the high-level parts that make up the system. This diagram depicts, at a high level, what components form part of the system and how they are interrelated. A component diagram depicts the components culled after the system has undergone the development or construction phase.

9. Deployment diagram:

The deployment diagram captures the configuration of the runtime elements of the application. This diagram is by far most useful when a system is built and ready to be deployed. Now that we have an idea of the different UML diagrams, let us see if we can somehow group together these diagrams to enable us to further understand how to use them.

UML Diagram Classification—Static, Dynamic, and Implementation:

Static: The static characteristic of a system is essentially the structural aspect of the system. The static characteristics define what parts the system is made up of.

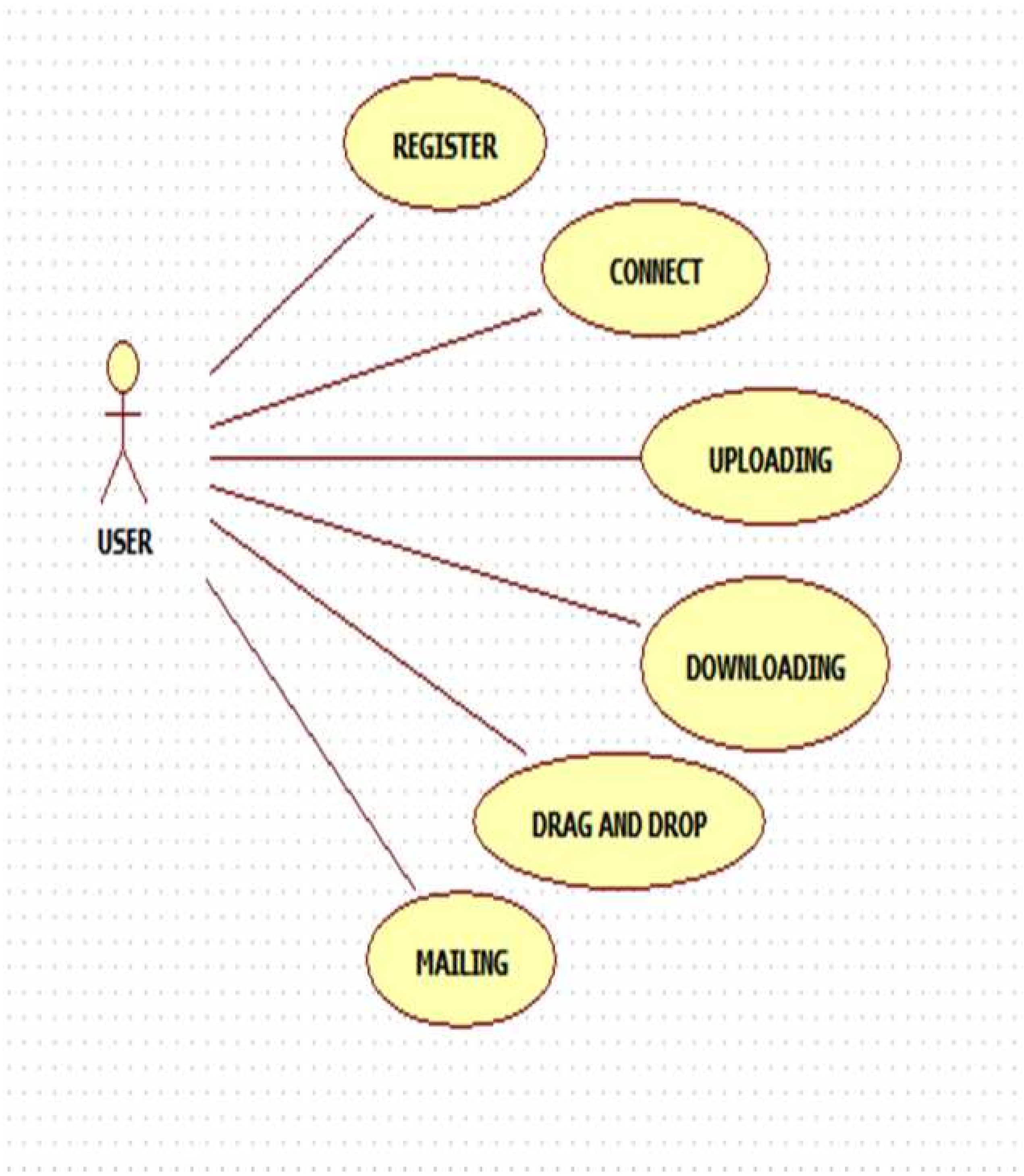
Dynamic: The behavioral features of a system; for example, the ways a system behaves in response to certain events or actions are the dynamic characteristics of a system.

Implementation: The implementation characteristic of a system is an entirely new feature that describes the different elements required for deploying a system.

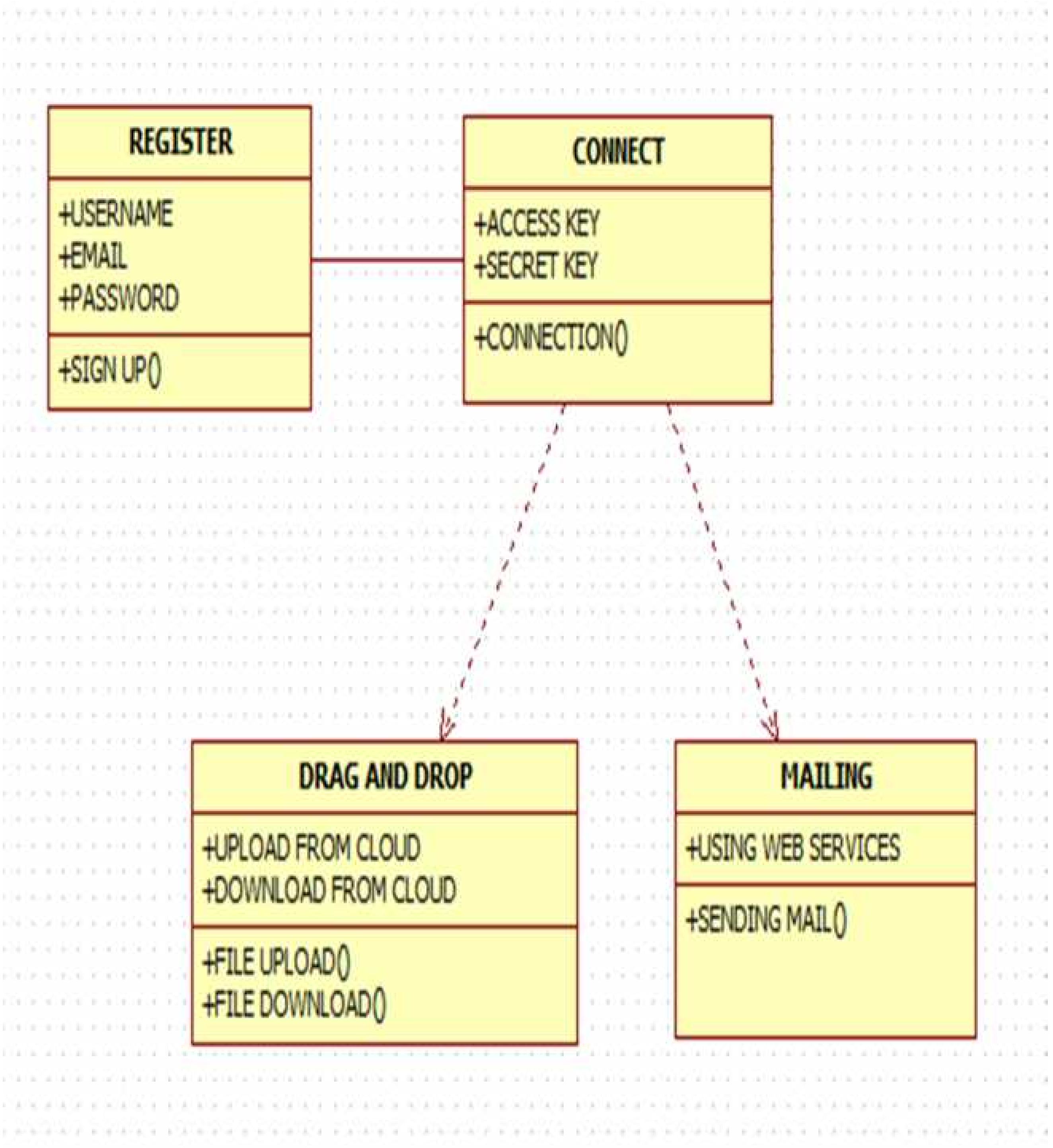
The UML diagrams that fall under each of these categories are:

- **Static:**
  - Use case diagram
  - Class diagram
- **Dynamic:**
  - Object diagram
  - State diagram
  - Activity diagram
  - Sequence diagram
  - Collaboration diagram
- **Implementation:**
  - Component diagram
  - Deployment diagram

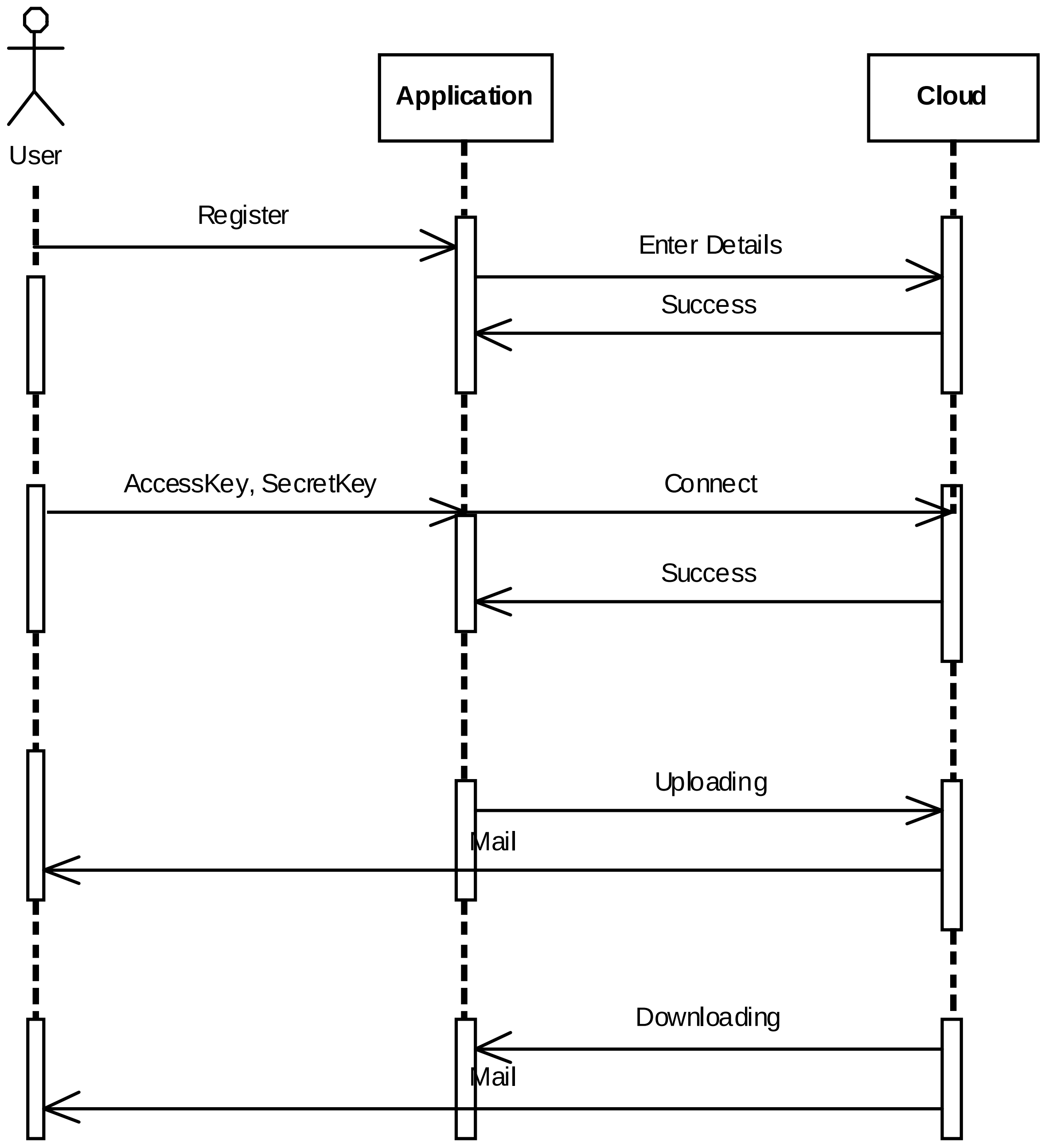
USE CASE DIAGRAM:



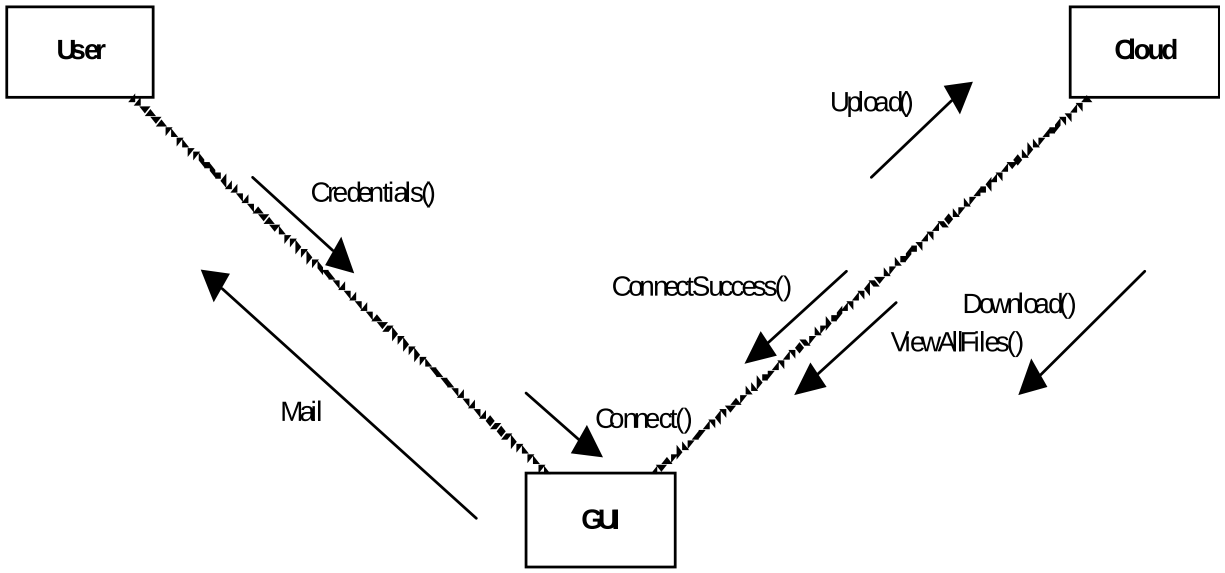
CLASS DIAGRAM:



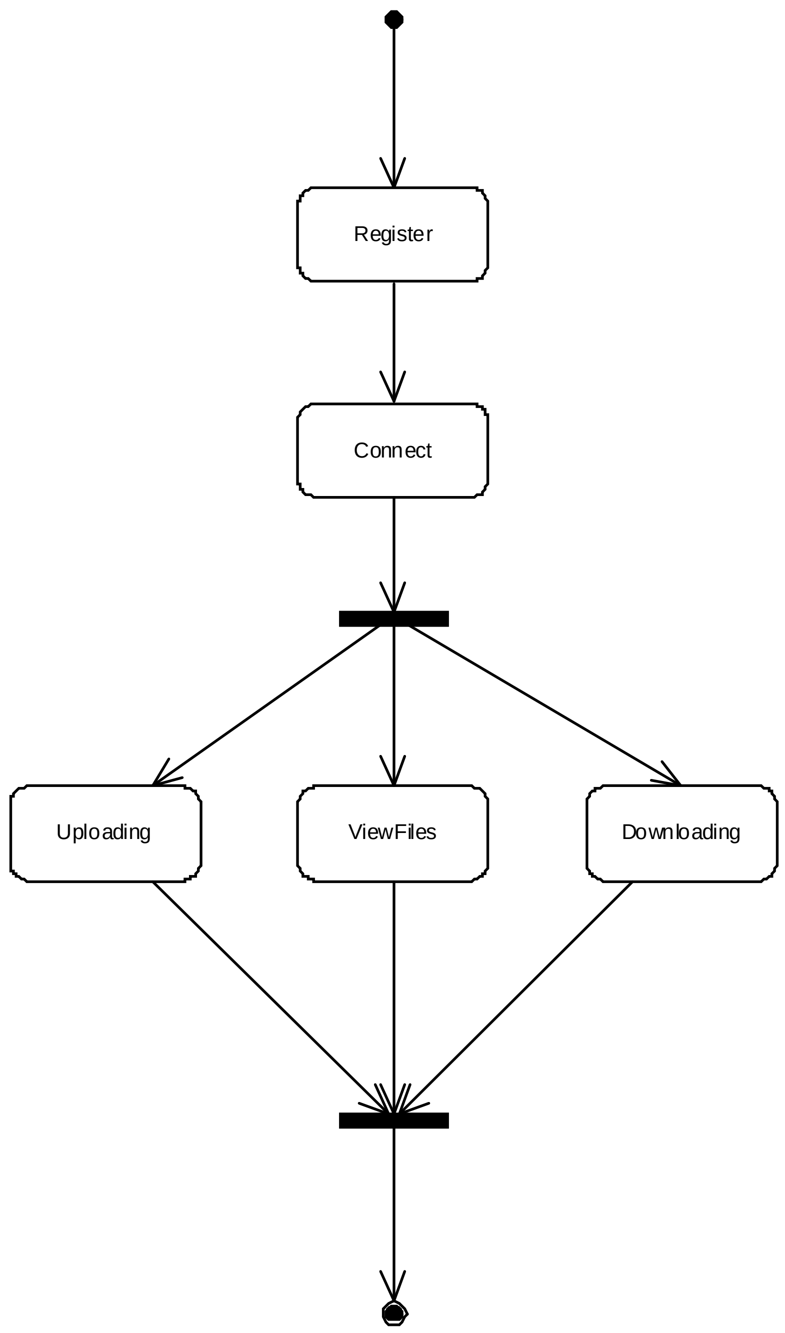
SEQUENCE DIAGRAM:



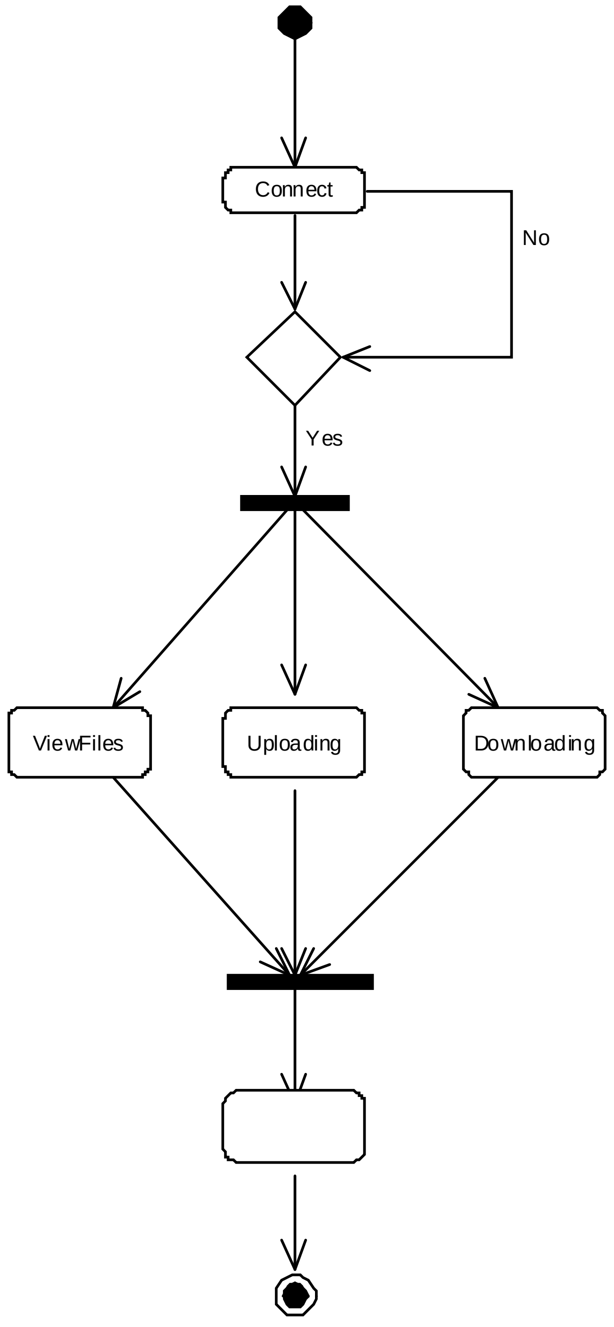
COLLABORATION DIAGRAM:



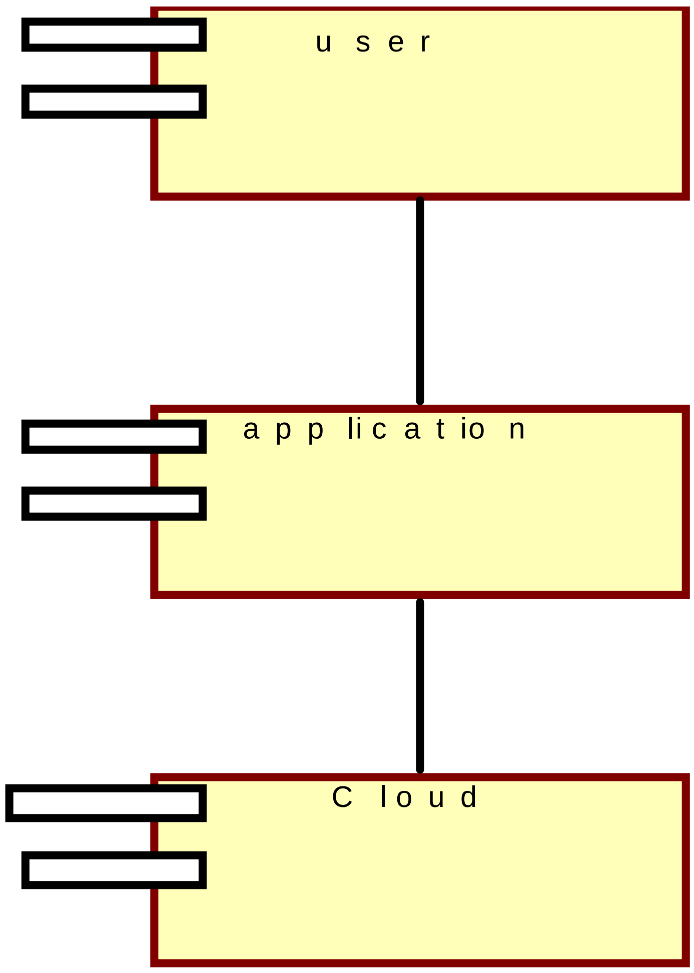
STATECHART DIAGRAM:



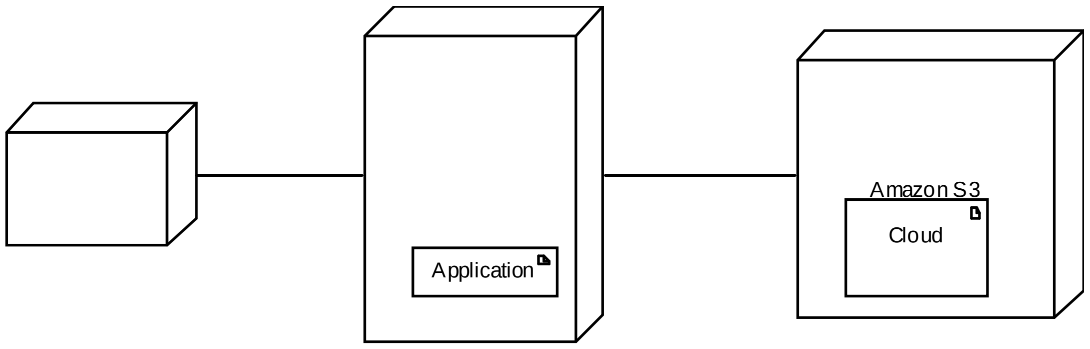
ACTIVITY DIAGRAM:



COMPONENT DIAGRAM:



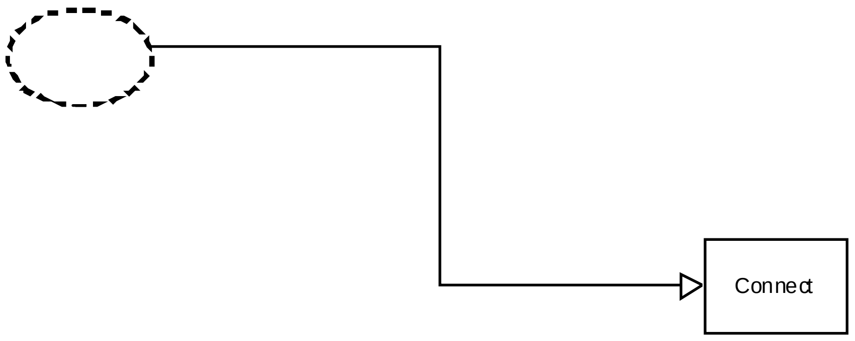
DEPLOYMENT DIAGRAM:



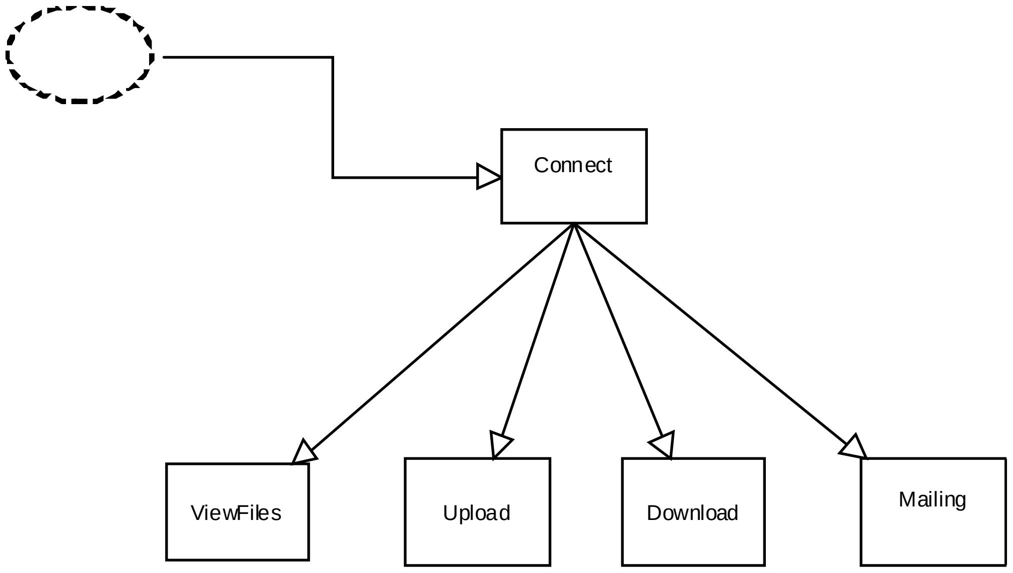
DATA FLOW DIAGRAM

User:

Level 0 :



Level 1:



4.3 MODULE DESIGN AND ORGANISATION

User:

In this module cloud will provide register for different users after login user can manipulate his folders and files i.e., user can upload or download his files and folders which will be directly affected on to cloud. We provide drag and drop facility to dump the data into cloud folder to provide easy access, sending an email to the current administrator for the changes done in the cloud using web services.

4.4 CONCLUSION

In this way we can design the layout of the project which is to be implemented during the construction phase. Thus we will have a clear picture of the project before being coded. Hence any necessary enhancements can be made during this phase and coding can be started.

5. IMPLENTATION & RESULTS

5.1 INTRODUCTION

The implementation part is the most important phase of the project. In this phase, we code the entire project in the chosen software according to the design laid during the previous phase. The code has to be in such a way that the user requirements are satisfied and also not complicated for the user i.e., the user interface or GUI has to be easy to navigate. The code should be efficient in all terms like space, easy to update, etc. In this manner, we can complete the coding part of the project and later it can be sent for testing before being delivered to the customer.

Microsoft .Net Framework

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet. The .NET Framework is designed to fulfill the following objectives:

- To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.
- To provide a code-execution environment that minimizes software deployment and versioning conflicts.
- To provide a code-execution environment that guarantees safe execution of code, including code created by an unknown or semi-trusted third party.
- To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
- To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.

5.2 SAMPLE CODE

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Amazon;
using Amazon.S3;
using System.IO;
using Amazon.S3.Model;
using CryptographicEditor;
using System.Reflection;

namespace CloudFTP
{
    public partial class Form1 : Form
    {

        public Form1()
        {
            InitializeComponent();
            txtAccessKey.Text = "AKIAIVP2GAQDZR5IFIFQ";
            txtSecretKey.Text ="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";
        }

        private void btnConnect_Click(object sender, EventArgs e)
        {
            try
            {
                if (txtAccessKey.Text.Trim() == "" || txtSecretKey.Text.Trim() == "")
                {
                    MessageBox.Show("Fields should not be empty");
                }
                else
                {
                    AmazonS3 s3 = new AmazonS3Client
                            (txtAccessKey.Text.Trim(), txtSecretKey.Text.Trim());
                    ListBucketsResponse req = s3.ListBuckets();
                    lbResult.ForeColor = System.Drawing.Color.Green;
                    lbResult.Items.Insert(0, "Connected Successfully At :"
                                  +  DateTime.Now.ToString());
                    //.ImageList = imageList1;
                    treeView1.ImageList = imageList1;
                    CreateTree();

                }
            }

            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
                lbResult.ForeColor = System.Drawing.Color.Red;
                lbResult.Items.Insert(0, "Connection failed At :"
                              + DateTime.Now.ToString());
            }
        }

        Caxer cax = new Caxer();
        private void Form1_Load(object sender, EventArgs e)
        {
            trvSystemFiles.ImageList = imageList1;

            cax.CreateTree(trvSystemFiles);
        }

        private void trvSystemFiles_BeforeExpand
        (object sender, TreeViewCancelEventArgs e)
        {
            if (e.Node.Nodes[0].Text == "")
                cax.EnumerateDirectory(e.Node, contextMenuStrip1);
        }
```

```
  public DataTable LoadFolders(TreeNode tn)
  {
     DataTable dt = cax.GetFolders(tn);
     return dt;
  }

public DataTable LoadFolders(string s)
  {
     DataTable dt = new DataTable(); // = cax.GetFolders(();
     return dt;
  }
  private void trvSystemFiles_AfterSelect
  (object sender, TreeViewEventArgs e)
  {
     textBox1.Text = trvSystemFiles.SelectedNode.FullPath;
     if (e.Node.Nodes[0].Text == "")
        cax.EnumerateDirectory(e.Node, contextMenuStrip1);
     DataTable dt = LoadFolders(e.Node);
     //dataGridView1.DataSource = dt;
     lvFiles.Columns.Clear();
     lvFiles.Items.Clear();

     foreach (DataColumn c in dt.Columns)
     {
        //adding names of columns as Listview columns
        ColumnHeader h = new ColumnHeader();
        h.Text = c.ColumnName;

        h.Width = 100;
        lvFiles.Columns.Add(h);
     }

     //adding Datarows as listview Grids
     int i = 0;
     foreach (DataRow rr in dt.Rows)
     {
        string[] str = new string[dt.Columns.Count];
        for (int col = 0; col <= dt.Columns.Count - 1; col++)
        {
            str[col] = rr[col].ToString();

        }

        ListViewItem ii;
        ii = new ListViewItem(str);
        if (str[1] == "File Folder")
        {
            ii.ImageIndex = 1;
        }
        else if (str[1] == ".doc" || str[1] == ".docx")
        {
            ii.ImageIndex = 2;
        }
        else
            ii.ImageIndex = 3;
        ii.ToolTipText = ii.Text;
        lvFiles.Items.Add(ii);
    }
}

private void trvSystemFiles_BeforeSelect
(object sender, TreeViewCancelEventArgs e)
{
}

private void trvSystemFiles_BeforeSelect_1
(object sender, TreeViewCancelEventArgs e)
{

    // trvSystemFiles.ImageKey = trvSystemFiles.ImageKey;
}

private void lvFiles_MouseDoubleClick(object sender, MouseEventArgs e)
{
    foreach (ListViewItem lvt in lvFiles.Items)
    {
        if (lvt.Selected == true)
        {
            string s = lvt.Text;

            DataTable dt
            LoadFolders(trvSystemFiles.SelectedNode.FullPath.ToString()+ s);
            //System.Diagnostics.Process.Start(textBox1.Text+"/" + lvt.Text);
        }
    }
}

private void trvSystemFiles_MouseDoubleClick
(object sender, MouseEventArgs e)
{
    //System.Diagnostics.Process.Start(trvSystemFiles.SelectedNode.FullPath);
}

private void lvFiles_SelectedIndexChanged(object sender, EventArgs e)
{

}
int i = 0;
private void lvFiles_ColumnClick(object sender, ColumnClickEventArgs e)
{

    if (i == 0)
    {
        i++;
        lvFiles.Sorting = SortOrder.Ascending;
        lvFiles.Sort();
    }
    else
    {
        lvFiles.Sorting = SortOrder.Descending;
```

```
            lvFiles.Sort();
            i = 0;

        }
    }

   private void lvFiles_AfterLabelEdit(object sender, LabelEditEventArgs e)
    {
        if (File.Exists(path + fname))
        {
            File.Move(path + fname, path + e.Label);

        }
    }
    static string path = string.Empty;
    static string fname = string.Empty;

    private void lvFiles_BeforeLabelEdit(object sender, LabelEditEventArgs e)
    {

        foreach (ListViewItem lvt in lvFiles.Items)
        {
            if (lvt.Selected == true)
            {
                path = textBox1.Text;
                fname = lvt.Text;
                //string s = lvt.Text;
                //System.Diagnostics.Process.Start(textBox1.Text + "/" + lvt.Text);
            }
        }
    }

  public static string GetImage(string value)
    {
        if (value == "Directory")
        {
            StringobjImg=Path.GetDirectoryName(Assembly.
                              GetExecutingAssembly().GetName().CodeBase);
            objImg = objImg.Replace("\\bin\\Debug", "\\image\\file.png");
            return objImg;
        }
        else
        {
            if (value == "mp3")
            {
               string objImg =Path.GetDirectoryName(Assembly.
                                 GetExecutingAssembly().GetName().CodeBase);
                objImg = objImg.Replace("\\bin\\Debug", "\\image\\music.png");
                return objImg;
            }
            else
            {
                string objImg = Path.GetDirectoryName(Assembly.
                                  GetExecutingAssembly().GetName().CodeBase);
                objImg = objImg.Replace("\\bin\\Debug", "\\image\\smallFile.gif");
                return objImg;
            }
        }
    }

    private void contextMenuStrip1_Click(object sender, EventArgs e)
    {
        Upload();
    }

    private void contextMenuStrip2_Click(object sender, EventArgs e)
    {
        Download();
    }

    public void Download()
    {
        AmazonS3 s3 = AWSClientFactory.CreateAmazonS3Client
                          (txtAccessKey.Text, txtSecretKey.Text);
```

```
try
{

    string key = string.Empty;
    string dest = string.Empty;


    string listSelect = listView1.SelectedItems[0].Text;

    string lstSelect2 = listView1.SelectedItems[0].SubItems[1].Text;

    //string lbl = listView1.SelectedItems[0].ToString();
    if (lstSelect2.ToString() == "Directory")
    {
        key = listSelect.ToString();
    }

    else
    {
        key = listSelect.ToString() + "." + lstSelect2.ToString();
    }

    GetObjectRequest objRequest = new GetObjectRequest();
    objRequest.BucketName = textBox2.Text;
    objRequest.Key = key;
    dest = textBox1.Text;

    if (key.IndexOf("/") >= 0)
    {
        string path = key.Substring(0, key.LastIndexOf("/"));
        Directory.CreateDirectory(Path.Combine(dest, path));
        string[] filename = key.Split('/');
        string file = filename[filename.Length - 1].ToString();
        dest = Path.Combine(dest, path);
        dest = Path.Combine(dest, file);

    }
```

```
         else
         {
            dest = textBox1.Text;
         }
         using (S3Response getObjectResponse = s3.GetObject(objRequest))
         {
            if (!File.Exists(dest))
            {
               using (Stream s = getObjectResponse.ResponseStream)
               {
                  using (FileStream fs = new FileStream(dest,
                        FileMode.Create, FileAccess.Write))
                  {
                     byte[] data = new byte[32768];
                     int bytesRead = 0;
                     do
                     {
                        bytesRead = s.Read(data, 0, data.Length);
                        fs.Write(data, 0, bytesRead);
                     }
                     while (bytesRead > 0);
                     fs.Flush();
                  }
               }
            }

            MessageBox.Show("Downloaded Successfully In:" + textBox1.Text);
            AccessWebService();
            lvFiles.BeginUpdate();

          }

      }

      catch (Exception ex)
      {
         MessageBox.Show(ex.Message);
      }
```

```
}

public void Upload()
{
   string status = "";

   string filePath = textBox1.Text;
   string uplSelectedFile = filePath + "/" + lvFiles.SelectedItems[0].Text;
            if (uplSelectedFile != null)
   {
      try
      {
         string fn = lvFiles.SelectedItems[0].Text;

         AmazonS3 s3 = AWSClientFactory.CreateAmazonS3Client
                             (txtAccessKey.Text, txtSecretKey.Text);
         PutObjectRequest requestObject = new PutObjectRequest();
         requestObject.BucketName = textBox2.Text;
         requestObject.Key = fn;

         FileStream fileStream = new FileStream
                                    (uplSelectedFile,FileMode.Open);
         requestObject.InputStream = fileStream;
         PutObjectResponse objectResponse = s3.PutObject(requestObject);
         fileStream.Close();
         MessageBox.Show("Uploaded Successfully In :" + textBox2.Text);
         AccessWebService();
        }

      catch (Exception ex)
      {
         MessageBox.Show(ex.Message);
      }
   }
}

public void CreateTree()
```

```
{
    try
    {
        AmazonS3 client = AWSClientFactory.CreateAmazonS3Client
                                (txtAccessKey.Text, txtSecretKey.Text);

        ListBucketsResponse response = client.ListBuckets();

        foreach (S3Bucket bucket in response.Buckets)
        {
            TreeNode treenode = new TreeNode(bucket.BucketName);
            treenode.ImageIndex = 12;
            treeView1.Nodes.Add(treenode);

        }
    }
    catch (Exception ex)
    {

    }

}

private void treeView1_AfterSelect(object sender, TreeViewEventArgs e)
{
    textBox2.Text = treeView1.SelectedNode.FullPath;
    //if (e.Node.Nodes[0].Text == "")

    DataTable dt = myListview();

    ListViewItem ii;
    listView1.Columns.Clear();
    listView1.Items.Clear();
    foreach (DataColumn c in dt.Columns)
    {
        //adding names of columns as Listview columns
        ColumnHeader h = new ColumnHeader();
        h.Text = c.ColumnName;
        h.Width = 100;
        listView1.Columns.Add(h);
    }

    foreach (DataRow rr in dt.Rows)
    {
        string[] str = new string[dt.Columns.Count];

        for (int col = 0; col <= dt.Columns.Count - 1; col++)
        {
            str[col] = rr[col].ToString();
        }


        ii = new ListViewItem(str);
        if (str[1] == "Directory")
        {
            ii.ImageIndex = 4;
        }

        else if (str[1] == "mp3")
        {
            ii.ImageIndex = 7;
        }
        else if (str[1] == "jpeg" || str[1] == "jpg")
        {
            ii.ImageIndex = 14;
        }
        else if (str[1] == "png")
        {
            ii.ImageIndex = 13;
        }
        else if (str[1] == "docx")
        {
            ii.ImageIndex = 9;
        }
        else
        {
            ii.ImageIndex = 11;
         }

         ii.ToolTipText = ii.Text;

         listView1.Items.Add(ii);
      }
   }

   public DataTable myListview()
   {
      DataTable dt = new DataTable();
      try
      {

         string _extention = string.Empty;


         DataColumn dc1 = new DataColumn("FileName", typeof(string));
         DataColumn dc2 = new DataColumn("Type", typeof(string));
         DataColumn dc3 = new DataColumn("Size", typeof(string));
         DataColumn dc4 = new DataColumn("LastModifiedDate",
                                         typeof(string));

         dt.Columns.Add(dc1);
         dt.Columns.Add(dc2);
         dt.Columns.Add(dc3);
         dt.Columns.Add(dc4);

         AmazonS3 s3 = AWSClientFactory.CreateAmazonS3Client
                     (txtAccessKey.Text, txtSecretKey.Text);

         ListObjectsRequest requestListObjects = new ListObjectsRequest();

         requestListObjects.BucketName = textBox2.Text;
         ListObjectsResponse listObjectsResponse = s3.ListObjects
                                         (requestListObjects);
```

```
        foreach (S3Object s3Object in listObjectsResponse.S3Objects)
        {
            _extention = s3Object.Key;
            string[] ext = _extention.Split('.');
            long size = s3Object.Size;
            string modifieddate = s3Object.LastModified;

            if (ext.Length == 2)
            {

                dt.Rows.Add(ext[0].ToString(), ext[1].ToString(),
                size, modifieddate);

            }
            else
            {
                dt.Rows.Add(ext[0].ToString(), "Directory",
                size.ToString(), modifieddate);
            }
        }

    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    return dt;
}

private void listView1_MouseDown(object sender, MouseEventArgs e)
{

    if (this.listView1.SelectedItems != null)
    {
        this.listView1.DoDragDrop(this.listView1.SelectedItems,
        DragDropEffects.Copy);
        Download();
    }
}

private void listView1_MouseDoubleClick
(object sender, MouseEventArgs e)
{
    Download();
}

private void lvFiles_MouseDown(object sender, MouseEventArgs e)
{
    if (this.lvFiles.SelectedItems != null)
    {
        this.lvFiles.DoDragDrop
        (this.lvFiles.SelectedItems,DragDropEffects.Copy);
        Upload();
    }
}

private void treeView1_BeforeExpand
(object sender, TreeViewCancelEventArgs e)
{
    AmazonS3 s3 = AWSClientFactory.CreateAmazonS3Client
                        (txtAccessKey.Text, txtSecretKey.Text);
    ListObjectsRequest requestListObjects = new ListObjectsRequest();
    requestListObjects.BucketName = treeView1.SelectedNode.ToString();
    ListObjectsResponse listObjectsResponse = s3.ListObjects
                                                    (requestListObjects);

    foreach (S3Object s3Object in listObjectsResponse.S3Objects)
    {
        TreeNode listbuckets = new TreeNode(s3Object.Key);

        string bct = listbuckets.ToString();

        string[] ext = bct.Split('.');

        if (ext.Length >= 2)
```

```
            {

                if (ext[1] == "Directory")
                {
                    listbuckets.ImageIndex = 4;
                }

                else if (ext[1] == "mp3")
                {
                    listbuckets.ImageIndex = 7;
                }
                else if (ext[1] == "jpeg" || ext[1] == "jpg")
                {
                    listbuckets.ImageIndex = 14;
                }
                else if (ext[1] == "docx")
                {
                    listbuckets.ImageIndex = 9;
                }
                else if (ext[1] == "png")
                {
                    listbuckets.ImageIndex = 13;
                }
                else
                {
                    listbuckets.ImageIndex = 11;
                }

            }
            else
            {
                listbuckets.ImageIndex = 1;
            }

        }
    }

    public void AccessWebService()
```

```
        {
            WebReferance.Service1 service = new WebReferance.Service1();
            service.SendingMail();
            MessageBox.Show("Oops modifying");
        }

    }
}
```

5.3 EXPLANATION OF KEY FUNCTIONS

ADO.NET

ADO.NET ARCHITECHTURE:

DATA STORE:

ADO.NET is an extension of the ADO data access model which consist of only the connected architecture. The Microsoft organization has realized the data related operations and have studied and analyzed different data related technologies among which they found ADO to be interesting , later on they extended the features of the ADO and defined own data related technology by refining the ADO and hence given the name as ADO.Net. The Microsoft organization, grouped some set of namespaces which can operates on data , and put together into technology called ADO.NET. The ADO.Net technology enhances the features of the ADO, which consist of only connected architecture where as in ADO.NET they have introduces a disconnected architecture.

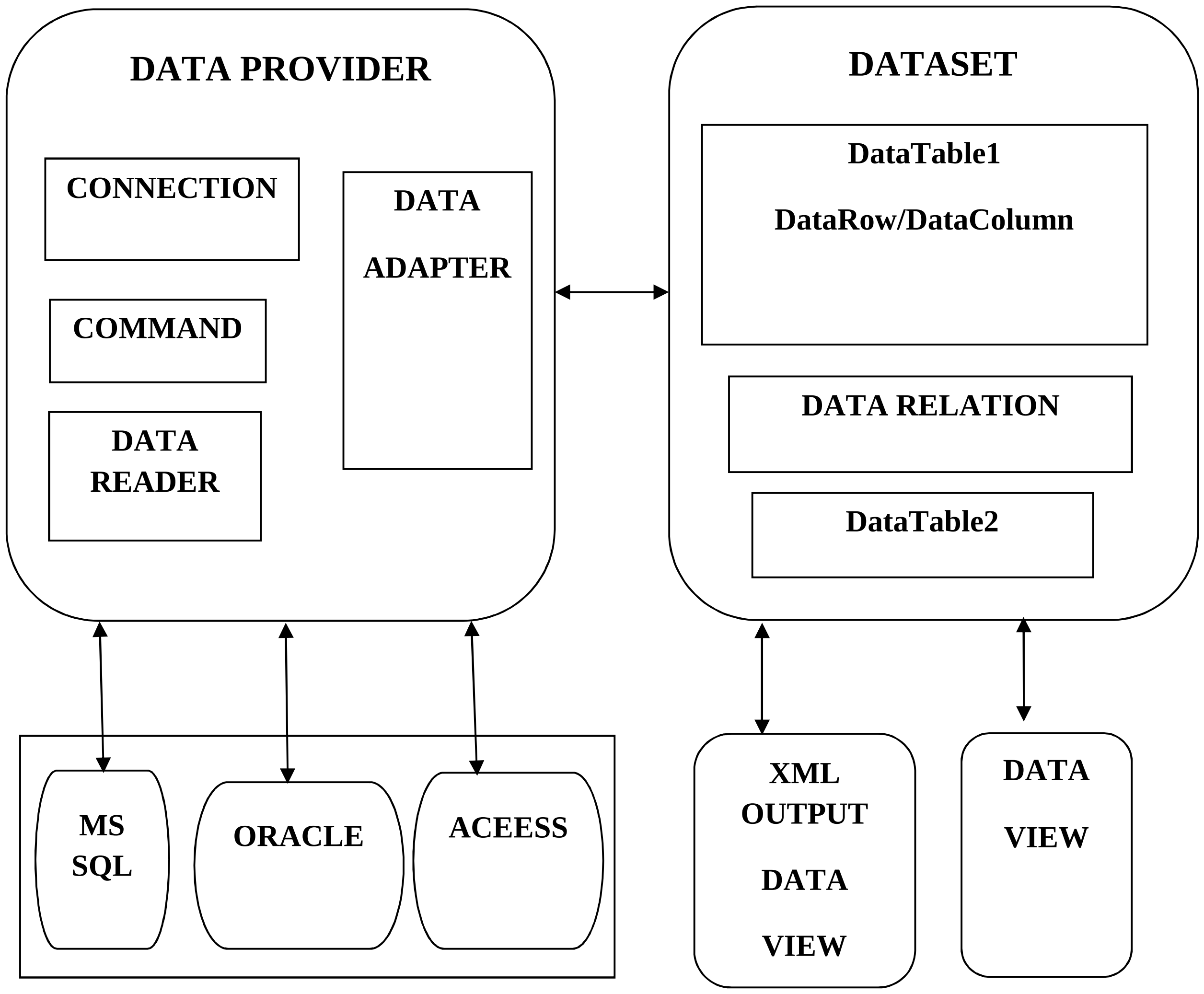
The connected architecture important feature is the data provider, which consists of the four important objects namely connection, command, data reader and data adapter.

The connection object provides the connection to the data store nothing but to the back end database servers. The connection class consists of default constructors and parameterized constructors. The constructor takes arguments, which provides the connection to the back end servers. The arguments that the connection class constructors takes are Data Source , Database and security.

The first parameter Data Source represents the server name to which our application needs to be connected. That's means from the available servers we need to select the particular data base server which can be done through the data source parameter. The second parameter indicates the database to which we are going to connect, that's means in that particular database server, to which data base we want to connect can be done through the data base parameter.

The third parameter security indicates, the security provided for the database server. If the server is running under windows authentication mode , than will use integrated security to be true that's means no need to specify the user name and password explicitly why because the system will takes the prebuilt username and password which has been set for the system.

On the other hand if the back end server Is running under sql authentication mode that will specify the username and password which has be set during the installation of the server, using the security parameter we can connect to the backend database server.



Also sqlconnection class consists of the methods such as open and close. The open method is used to open the connection to the database server. Whereas the close method is used to disconnect the connection from the server. Once the connection is opened while in the application use, the connection should be closed when the application terminates.

The another object of the dataprovider is the command object, using which one can write the queries in order to manipulate data in the database. Once the connection is opened ,the sqlcommand class make use of the connection and will operates on the database. The sqlcommand class will do manipulation using the queries or the stored procedures. Which has to be decided by the programmers whether they want to use the queries or the stored procedures using the method command type? If we want to use queries than we need to select the text query or else we need to select stored procedure option from the command type method.

Command object consists of three methods namely execute non query, execute reader and execute scalar. The execute non query will returns the integer values as an output which indicates how many records have been updated, or modified etc. the second method execute reader returns the complete records been affected by the operations , whereas execute scalar returns the first row first column value remaining will be neglected.

The third object in the connected architecture is the datareader ,which reads the data in a forward only mode , that’s means its retrieves the data from the data base server and forwards it to the application.

The another object is the dataadapter which acts like an interface or bridge between the connected architecture and disconnected architecture.

In the disconnected architecture the important feature is the dataset. It’s a collection of datatables and datarows and the datatables will be linked using the data relations . when the dataset need to be filled ,its request to the data adapter which in turn fills the dataset by making use of fill method. Of dataadapter.

Features of ADO.NET are as follows:

- ADO.NET is the next evolution of ADO for the .Net Framework.
- ADO.NET was created with n-Tier, statelessness and XML in the forefront. Two new objects, the DataSet and DataAdapter, are provided for these scenarios.
- ADO.NET can be used to get data from a stream, or to store data in a cache for updates.
- There is a lot more information about ADO.NET in the documentation.

- Remember, you can execute a command directly against the database in order to do inserts, updates, and deletes. You don't need to first put data into a DataSet in order to insert, update, or delete it.
- Also, you can use a DataSet to bind to the data, move through the data, and navigate data relationships

Asp.net description:

Server-side managed code:

ASP.NET is the hosting environment that enables developers to use the .NET Framework to target Web-based applications. However, ASP.NET is more than just a runtime host; it is a complete architecture for developing Web sites and Internet-distributed objects using managed code. Both Web Forms and XML Web services use IIS and ASP.NET as the publishing mechanism for applications, and both have a collection of supporting classes in the .NET Framework.

XML Web services, an important evolution in Web-based technology, are distributed, server-side application components similar to common Web sites. However, unlike Web-based applications, XML Web services components have no UI and are not targeted for browsers such as Internet Explorer and Netscape Navigator. Instead, XML Web services consist of reusable software components designed to be consumed by other applications, such as traditional client applications, Web-based applications, or even other XML Web services. As a result, XML Web services technology is rapidly moving application development and deployment into the highly distributed environment of the Internet. If Web Forms pages execute in native machine language because, like any other managed application, they take full advantage of the runtime. In contrast, unmanaged ASP pages are always scripted and interpreted.

ASP.NET pages are faster, more functional, and easier to develop than unmanaged ASP pages because they interact with the runtime like any managed application. The .NET Framework also provides a collection of classes and tools to aid in development and consumption of XML Web services applications. XML Web services are built on standards such as SOAP (a remote procedure-call protocol), XML (an extensible data format), and WSDL (the Web Services Description Language). The .NET Framework is built on these standards to promote interoperability with non-Microsoft solutions.

For example, the Web Services Description Language tool included with the .NET Framework SDK can query an XML Web service published on the Web, parse its WSDL description, and produce C# or Visual Basic source code that your application can use to become a client of the XML Web service. The source code can create classes derived from classes in the class library that handle all the underlying communication using SOAP and XML parsing. Although you can use the class library to consume XML Web services directly, the Web Services Description Language tool and the other tools contained in the SDK facilitate your development efforts with the .NET Framework.

If you develop and publish your own XML Web service, the .NET Framework provides a set of classes that conform to all the underlying communication standards, such as SOAP, WSDL, and XML. Using those classes enables you to focus on the logic of your service, without concerning yourself with the communications infrastructure required by distributed software development.

Finally, like Web Forms pages in the managed environment, your XML Web service will run with the speed of native machine language using the scalable communication of IIS.

Active Server Pages.NET

ASP.NET is a programming framework built on the common language runtime that can be used on a server to build powerful Web applications. ASP.NET offers several important advantages over previous Web development models:

- **Enhanced Performance:** ASP.NET is compiled common language runtime code running on the server. Unlike its interpreted predecessors, ASP.NET can take advantage of early binding, just-in-time compilation, native optimization, and caching services right out of the box. This amounts to dramatically better performance before you ever write a line of code.
- **World-Class Tool Support**: The ASP.NET framework is complemented by a rich toolbox and designer in the Visual Studio integrated development environment. WYSIWYG editing, drag-and-drop server controls, and automatic deployment are just a few of the features this powerful tool provides.
- **Scalability and Availability**: ASP.NET has been designed with scalability in mind, with features specifically tailored to improve performance in clustered and multiprocessor environments. Further, processes are closely monitored and managed by the ASP.NET runtime, so that if one misbehaves (leaks, deadlocks), a new process can be created in its place, which helps keep your application constantly available to handle requests.

- **Customizability and Extensibility**: ASP.NET delivers a well-factored architecture that allows developers to "plug-in" their code at the appropriate level. In fact, it is possible to extend or replace any subcomponent of the ASP.NET runtime with your own custom-written component. Implementing custom authentication or state services has never been easier.
- **Security.** With built in Windows authentication and per-application configuration, you can be assured that your applications are secure.

Language Support:

The Microsoft .NET Platform currently offers built-in support for three languages: C#, Visual Basic, and JScript.

What is ASP.NET Web Forms?

The ASP.NET Web Forms page framework is a scalable common language runtime programming model that can be used on the server to dynamically generate Web pages.

Intended as a logical evolution of ASP (ASP.NET provides syntax compatibility with existing pages), the ASP.NET Web Forms framework has been specifically designed to address a number of key deficiencies in the previous model. In particular, it provides:

- The ability to create and use reusable UI controls that can encapsulate common functionality and thus reduce the amount of code that a page developer has to write.
- The ability for developers to cleanly structure their page logic in an orderly fashion (not "spaghetti code").
- The ability for development tools to provide strong WYSIWYG design support for pages (existing ASP code is opaque to tools).

ASP.NET Web Forms pages are text files with an .aspx file name extension. They can be deployed throughout an IIS virtual root directory tree. When a browser client requests .aspx resources, the ASP.NET runtime parses and compiles the target file into a .NET Framework class. This class can then be used to dynamically process incoming requests. (Note that the .aspx file is compiled only the first time it is accessed; the compiled type instance is then reused across multiple requests).

An ASP.NET page can be created simply by taking an existing HTML file and changing its file name extension to .aspx (no modification of code is required). For example, the following sample demonstrates a simple HTML page that collects a user's name and category preference and then performs a form postback to the originating page when a button is clicked:

ASP.NET provides syntax compatibility with existing ASP pages. This includes support for <% %> code render blocks that can be intermixed with HTML content within an .aspx file. These code blocks execute in a top-down manner at page render time.

Code-Behind Web Forms

ASP.NET supports two methods of authoring dynamic pages. The first is the method shown in the preceding samples, where the page code is physically declared within the originating .aspx file. An alternative approach--known as the code-behind method--enables the page code to be more cleanly separated from the HTML content into an entirely separate file.

Introduction to ASP.NET Server Controls

In addition to (or instead of) using <% %> code blocks to program dynamic content, ASP.NET page developers can use ASP.NET server controls to program Web pages. Server controls are declared within an .aspx file using custom tags or intrinsic HTML tags that contain a runat="server" attribute value. Intrinsic HTML tags are handled by one of the controls in the **System.Web.UI.HtmlControls** namespace. Any tag that doesn't explicitly map to one of the controls is assigned the type of **System.Web.UI.HtmlControls.HtmlGenericControl**.

Server controls automatically maintain any client-entered values between round trips to the server. This control state is not stored on the server (it is instead stored within an **<input type="hidden">** form field that is round-tripped between requests). Note also that no client-side script is required.

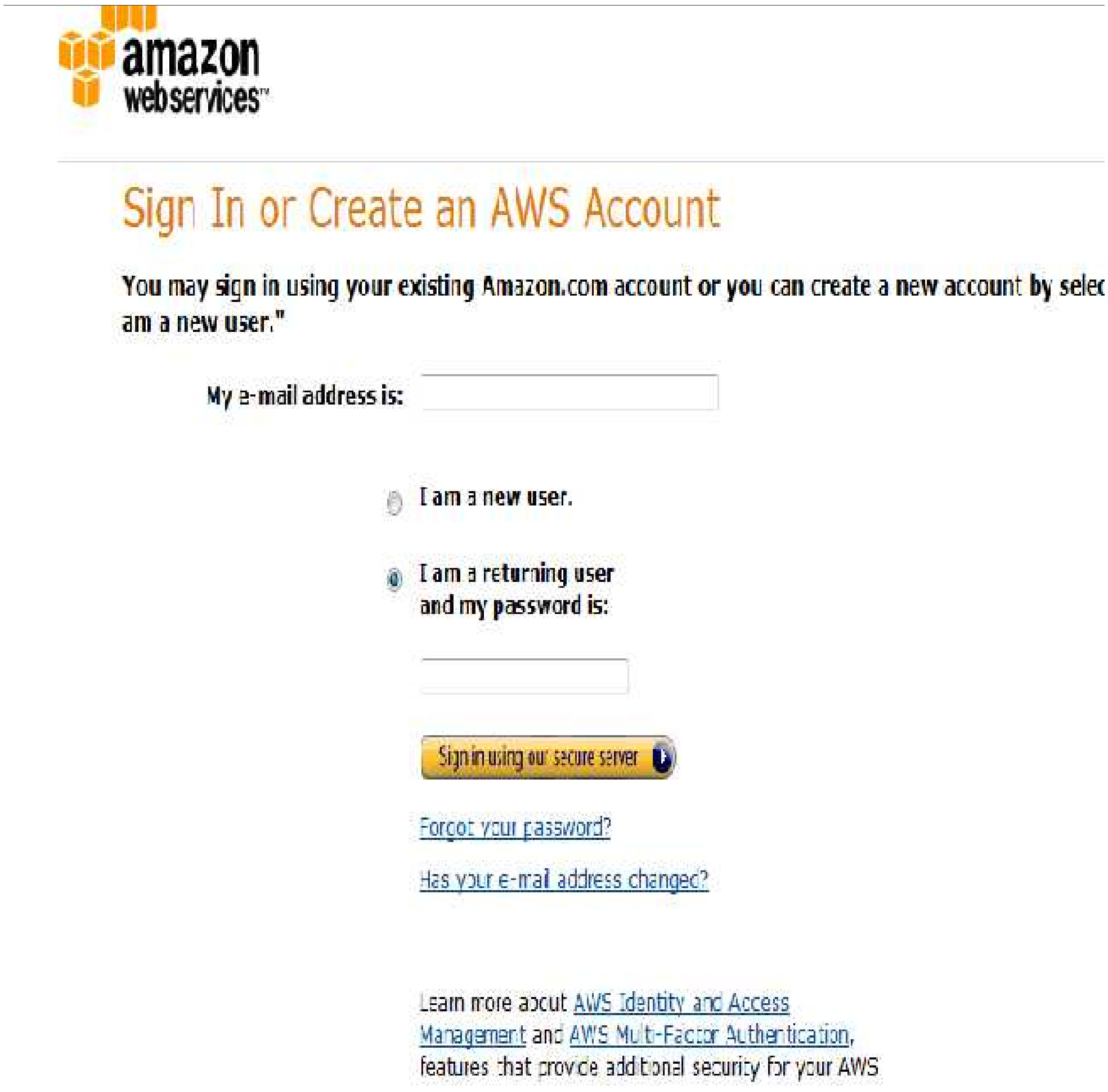
In addition to supporting standard HTML input controls, ASP.NET enables developers to utilize richer custom controls on their pages. For example, the following sample demonstrates how the **<asp:adrotator>** control can be used to dynamically display rotating ads on a page.

- ASP.NET Web Forms provide an easy and powerful way to build dynamic Web UI.
- ASP.NET Web Forms pages can target any browser client (there are no script library or cookie requirements).
- ASP.NET Web Forms pages provide syntax compatibility with existing ASP pages.
- ASP.NET server controls provide an easy way to encapsulate common functionality.
- ASP.NET validation controls provide an easy way to do declarative client or server data validation.

5.4 METHOD OF IMPLEMENTATION

5.4.1 OUTPUT SCREENS:

Cloud Registration:



amazon webservices™

Sign In or Create an AWS Account

You may sign in using your existing Amazon.com account or you can create a new account by selec
am a new user."

My e-mail address is:

I am a new user.

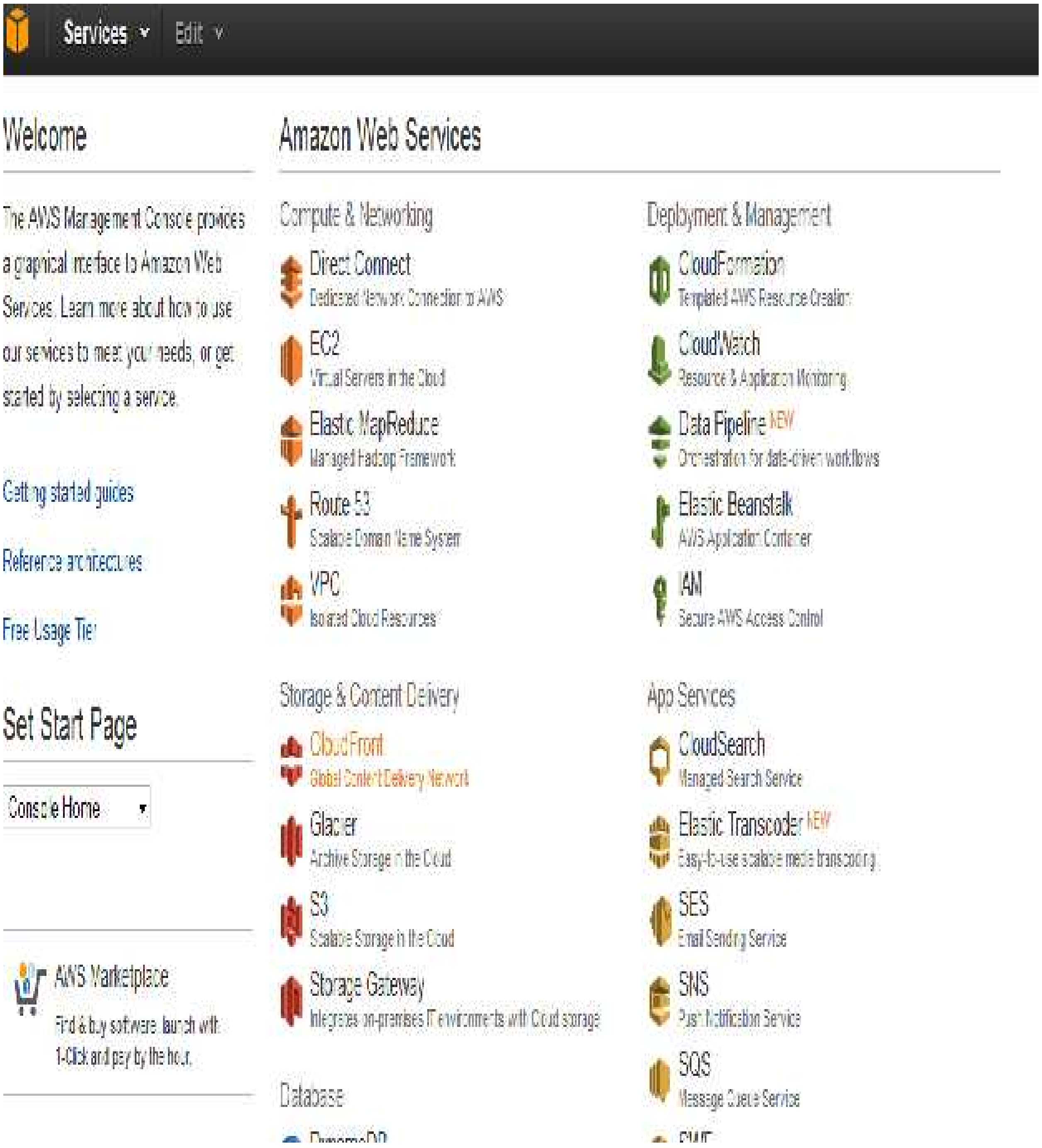
I am a returning user
and my password is:

Forgot your password?

Has your e-mail address changed?

Learn more about AWS Identity and Access Management and AWS Multi-Factor Authentication, features that provide additional security for your AWS

Amazon S3:



Welcome

The AWS Management Console provides a graphical interface to Amazon Web Services. Learn more about how to use our services to meet your needs, or get started by selecting a service.

Getting started guides

Reference architectures

Free Usage Tier

Set Start Page

Console Home

AWS Marketplace

Find & buy software, launch with 1-Click and pay by the hour.

Amazon Web Services

Compute & Networking

Direct Connect
Dedicated Network Connection to AWS

EC2
Virtual Servers in the Cloud

Elastic MapReduce
Managed Hadoop Framework

Route 53
Scalable Domain Name System

VPC
Isolated Cloud Resources

Storage & Content Delivery

CloudFront
Global Content Delivery Network

Glacier
Archive Storage in the Cloud

S3
Scalable Storage in the Cloud

Storage Gateway
Integrates on-premises IT environments with Cloud storage

Database

Deployment & Management

CloudFormation
Templated AWS Resource Creation

CloudWatch
Resource & Application Monitoring

Data Pipeline NEW
Orchestration for data-driven workflows

Elastic Beanstalk
AWS Application Container

IAM
Secure AWS Access Control

App Services

CloudSearch
Managed Search Service

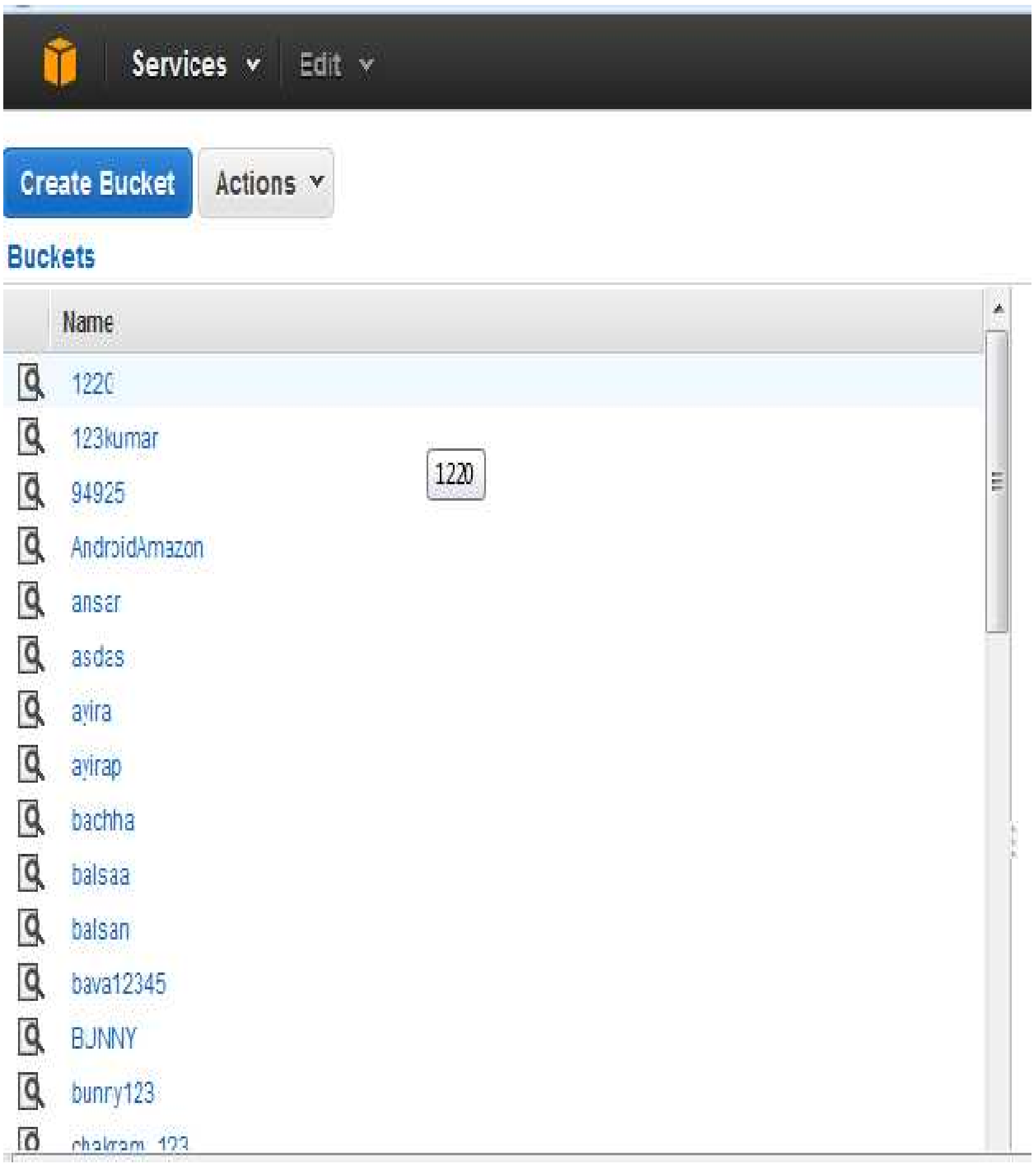
Elastic Transcoder NEW
Easy-to-use scalable media transcoding

SES
Email Sending Service

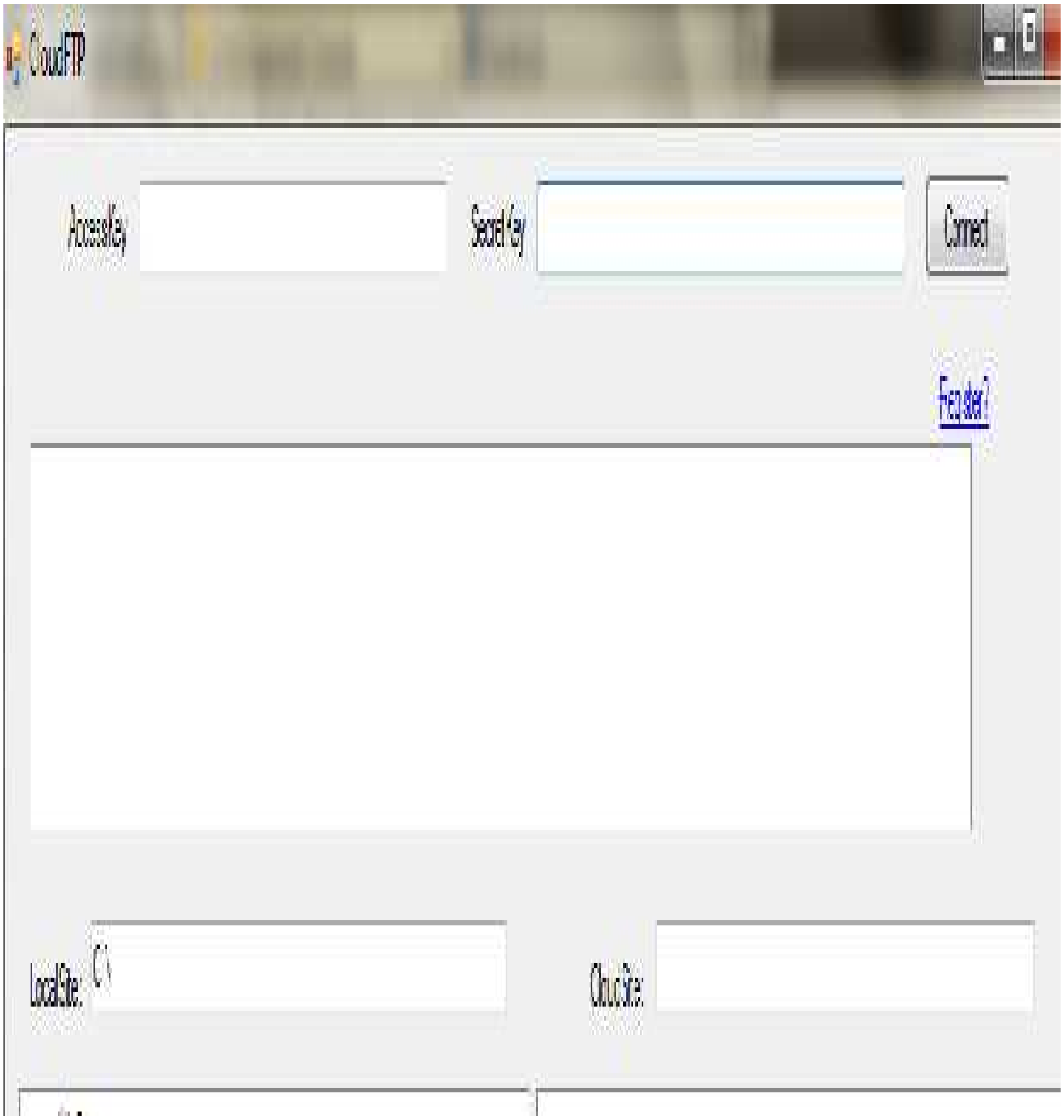
SNS
Push Notification Service

SQS
Message Queue Service

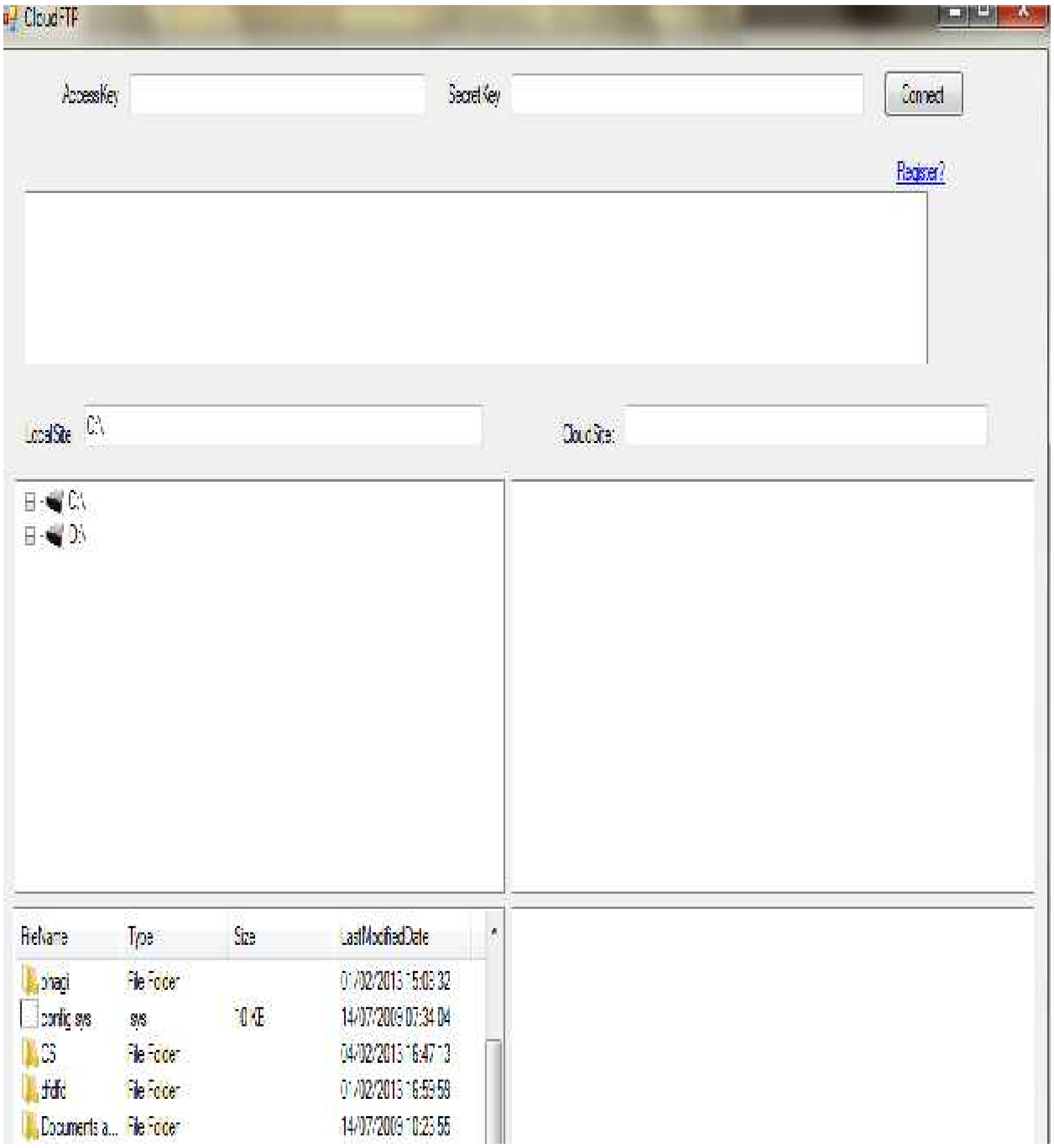
Cloud Folders:



Credentials:

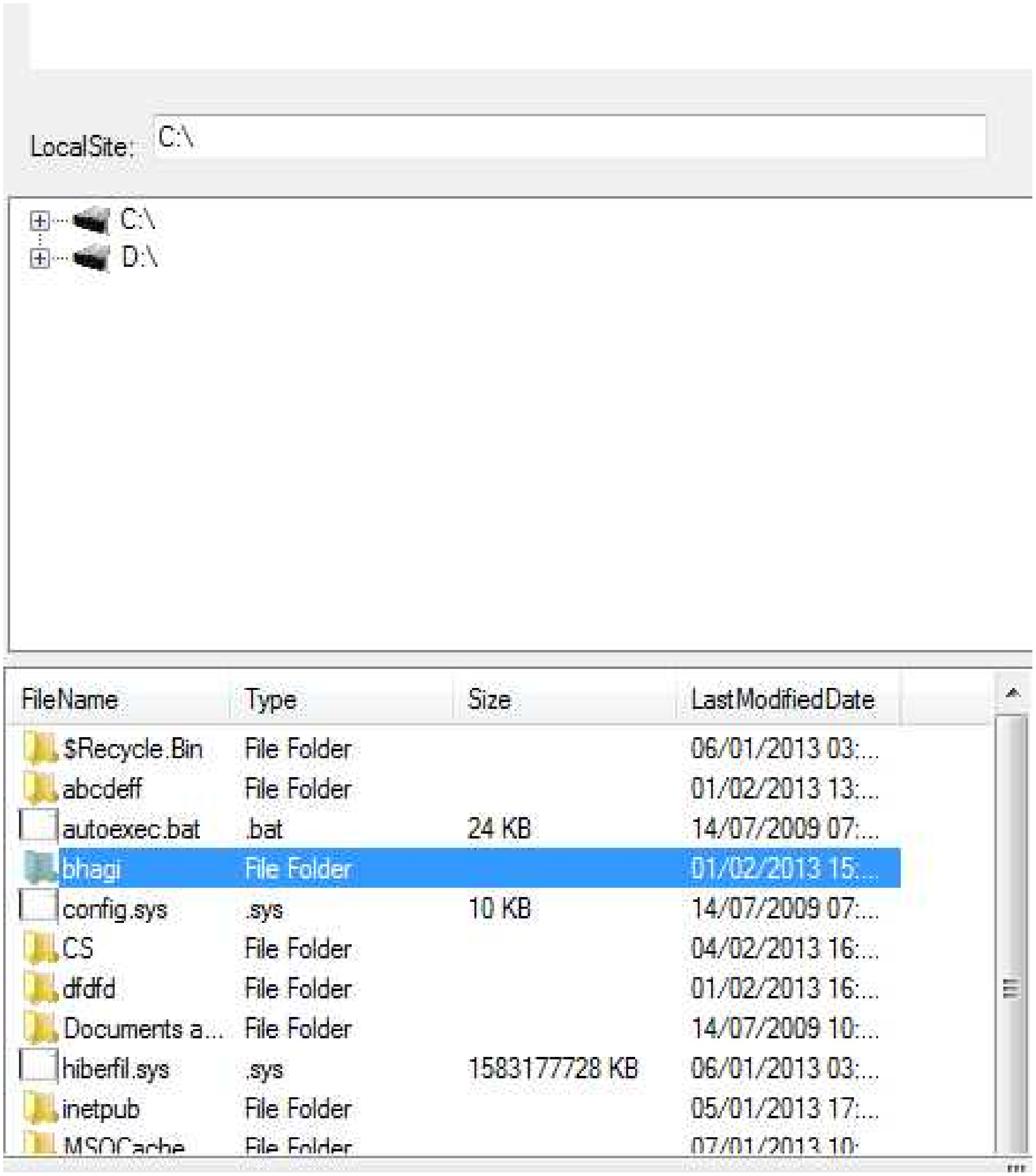


Application:



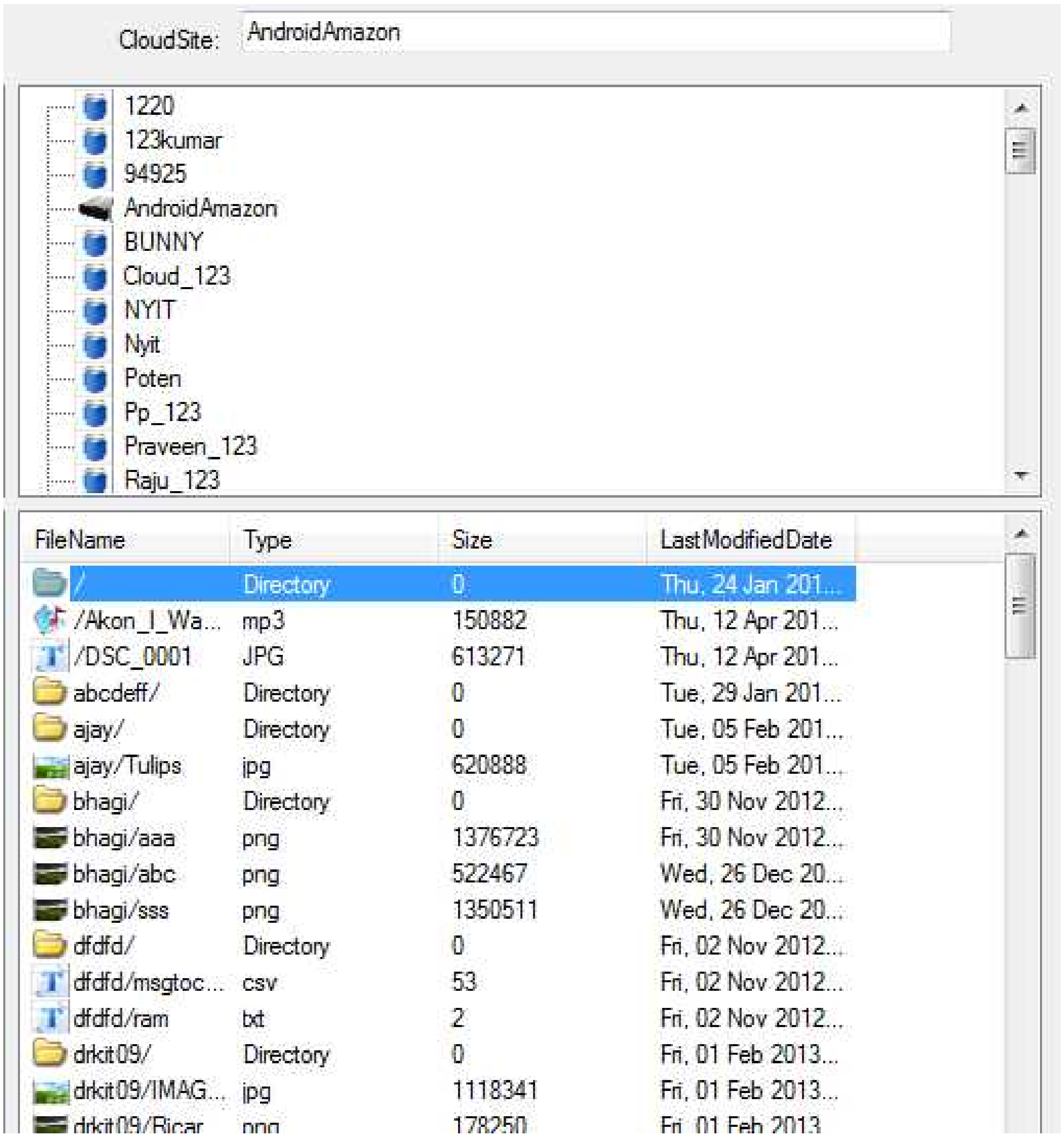
FileName	Type	Size	LastModifiedDate
bhagi	File Folder		01/02/2013 15:03:32
config.sys	sys	10 KB	14/07/2009 07:34:04
CS	File Folder		04/02/2013 16:47:13
dfdfd	File Folder		01/02/2013 16:59:58
Documents a...	File Folder		14/07/2009 10:25:55

Local Site:



FileName	Type	Size	LastModifiedDate
$Recycle.Bin	File Folder		06/01/2013 03:...
abcdeff	File Folder		01/02/2013 13:...
autoexec.bat	.bat	24 KB	14/07/2009 07:...
bhagi	File Folder		01/02/2013 15:...
config.sys	.sys	10 KB	14/07/2009 07:...
CS	File Folder		04/02/2013 16:...
dfdfd	File Folder		01/02/2013 16:...
Documents a...	File Folder		14/07/2009 10:...
hiberfil.sys	.sys	1583177728 KB	06/01/2013 03:...
inetpub	File Folder		05/01/2013 17:...
MSOCache	File Folder		07/01/2013 10:

Cloud Site:

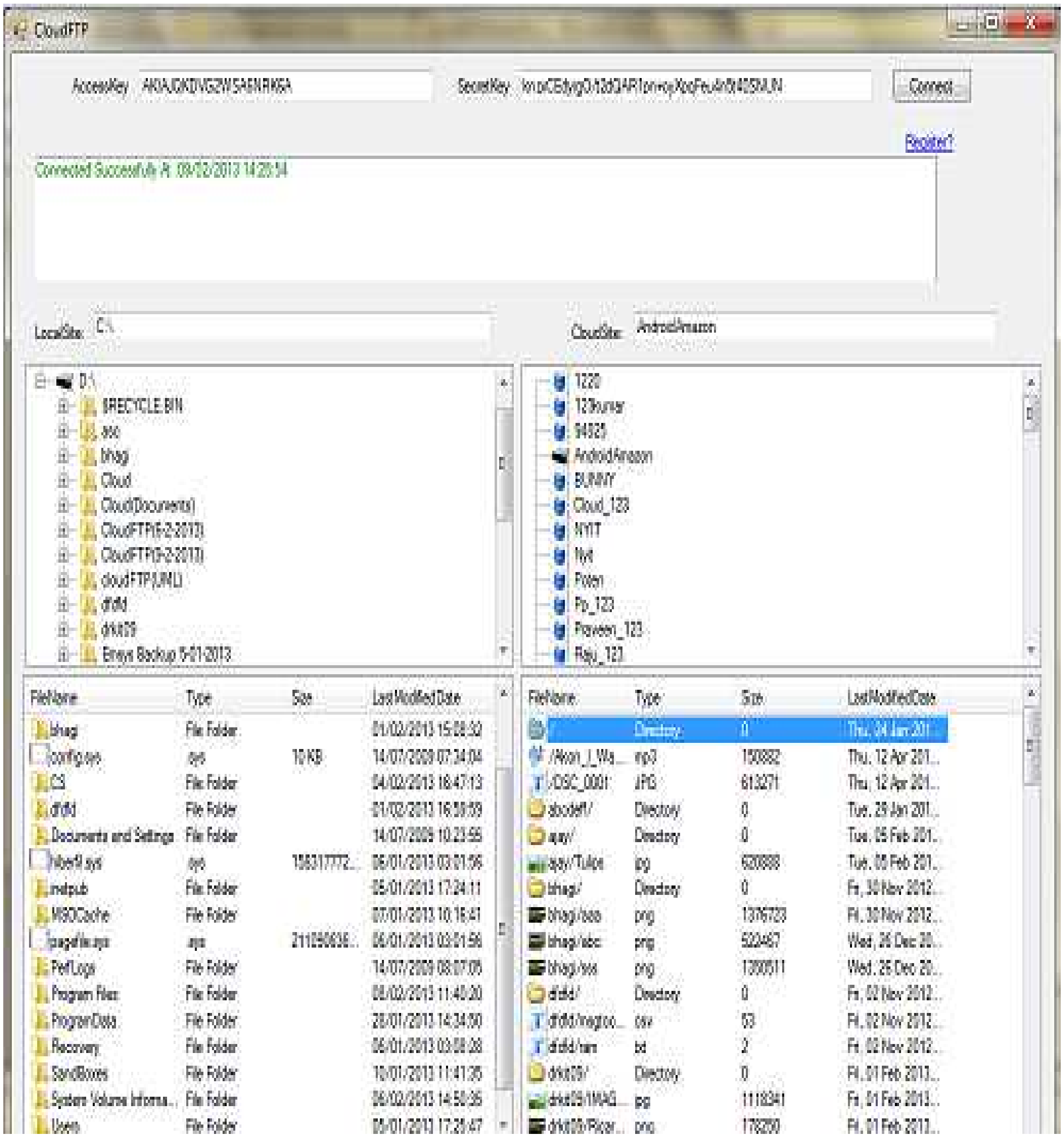


CloudSite: AndroidAmazon

- 1220
- 123kumar
- 94925
- AndroidAmazon
- BUNNY
- Cloud_123
- NYIT
- Nyit
- Poten
- Pp_123
- Praveen_123
- Raju_123

FileName	Type	Size	LastModifiedDate
/	Directory	0	Thu, 24 Jan 201...
/Akon_I_Wa...	mp3	150882	Thu, 12 Apr 201...
/DSC_0001	JPG	613271	Thu, 12 Apr 201...
abcdeff/	Directory	0	Tue, 29 Jan 201...
ajay/	Directory	0	Tue, 05 Feb 201...
ajay/Tulips	jpg	620888	Tue, 05 Feb 201...
bhagi/	Directory	0	Fri, 30 Nov 2012...
bhagi/aaa	png	1376723	Fri, 30 Nov 2012...
bhagi/abc	png	522467	Wed, 26 Dec 20...
bhagi/sss	png	1350511	Wed, 26 Dec 20...
dfdfd/	Directory	0	Fri, 02 Nov 2012...
dfdfd/msgtoc...	csv	53	Fri, 02 Nov 2012...
dfdfd/ram	txt	2	Fri, 02 Nov 2012...
drkit09/	Directory	0	Fri, 01 Feb 2013...
drkit09/IMAG...	jpg	1118341	Fri, 01 Feb 2013...
drkit09/Ricar	png	178250	Fri, 01 Feb 2013

Viewing Cloud Files:



AccessKey AKIAJDKDVG2WSA6NRK6A

SecretKey kn pCEdygO/t2dQ4R1on+vjXpqFeuAr8/405MJN

Connected Successfully At 09/02/2013 14:20:54

LocalSite: C:\

CloudSite: AndroidAmazon

```
D:\
  $RECYCLE.BIN
  aaa
  bhagi
  Cloud
  Cloud(Documents)
  CloudFTP(6-2-2013)
  CloudFTP(9-2-2013)
  cloudFTP(UML)
  dfdfd
  dkk09
  Emeys Backup 5-01-2013
```

```
1220
123kumar
94025
AndroidAmazon
BUNNY
Cloud_123
NYIT
Nyit
Poten
Pb_123
Praveen_123
Raju_123
```

FileName	Type	Size	LastModifiedDate
bhagi	File Folder		01/02/2013 15:08:32
config.sys	.sys	10 KB	14/07/2009 07:34:04
CS	File Folder		04/02/2013 18:47:13
dfdfd	File Folder		01/02/2013 16:59:09
Documents and Settings	File Folder		14/07/2009 10:23:55
hiberfil.sys	.sys	156317772...	06/01/2013 03:01:56
inetpub	File Folder		05/01/2013 17:24:11
MSOCache	File Folder		07/01/2013 10:16:41
pagefile.sys	.sys	21105063...	06/01/2013 03:01:56
PerfLogs	File Folder		14/07/2009 08:07:05
Program Files	File Folder		08/02/2013 11:40:20
ProgramData	File Folder		28/01/2013 14:34:00
Recovery	File Folder		06/01/2013 03:08:28
SandBoxes	File Folder		10/01/2013 11:41:35
System Volume Informa...	File Folder		09/02/2013 14:50:35
Users	File Folder		05/01/2013 17:25:47

FileName	Type	Size	LastModifiedDate
/	Directory	0	Thu, 24 Jan 201...
/Akon_I_Wa...	mp3	150932	Thu, 12 Apr 201...
/DSC_0001	JPG	613271	Thu, 12 Apr 201...
abcdef/	Directory	0	Tue, 29 Jan 201...
ajay/	Directory	0	Tue, 05 Feb 201...
ajay/Tulips	jpg	620888	Tue, 05 Feb 201...
bhagi/	Directory	0	Fri, 30 Nov 2012...
bhagi/aaa	png	1376723	Fri, 30 Nov 2012...
bhagi/abc	png	522467	Wed, 26 Dec 20...
bhagi/sss	png	1390511	Wed, 26 Dec 20...
dfdfd/	Directory	0	Fri, 02 Nov 2012...
dfdfd/negtco...	csv	53	Fri, 02 Nov 2012...
dfdfd/ram	txt	2	Fri, 02 Nov 2012...
dkk09/	Directory	0	Fri, 01 Feb 2013...
dkk09/IMAG...	jpg	1118341	Fri, 01 Feb 2013...
dkk09/Picor...	png	178250	Fri, 01 Feb 2013...

5.5 CONCLUSION

In this way we implemented the project successfully with the help of .NET framework for an easy interaction of the user with the interfaces. We proceed to the next phase i.e., testing which is very important before delivering the project.

6. TESTING & VALIDATION

6.1 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 well-planned series of steps that result in the successful construction of software. Testing is the set of 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.

The following are the Testing Objectives:

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

Scope:

A primary purpose for testing is to detect software failures so that defects may be uncovered and corrected. This is a non-trivial pursuit. Testing cannot establish that a product functions properly under all conditions but can only establish that it does not function properly under specific conditions. The scope of software testing often includes examination of code as well as execution of that code in various environments and conditions as well as examining the aspects of code: does it do what it is supposed to do and do what it needs to do. In the current culture of software development, a testing organization may be separate from the development team. There are various roles for testing team members. Information derived from software testing may be used to correct the process by which software is developed.

Defects and failures:

Not all software defects are caused by coding errors. One common source of expensive defects is caused by requirements gaps, e.g., unrecognized requirements,that result in errors of omission by the program designer. A common source of requirements gaps is non-functional requirements such as testability, scalability, maintainability, usability, performance, and security.

Software faults occur through the following process. A programmer makes an error (mistake), which results in a defect (fault, bug) in the software source code. If this defect is executed, in certain situations the system will produce wrong results, causing a failure. Not all defects will necessarily result in failures. For example, defects in dead code will never result in failures. A defect can turn into a failure when the environment is changed.

Examples of these changes in environment include the software being run on a new hardware platform, alterations in source data or interacting with different software. A single defect may result in a wide range of failure symptoms.

Compatibility:

A frequent cause of software failure is compatibility with another application, a new operating system, or, increasingly, web browser version. In the case of lack of backward compatibility, this can occur (for example...) because the programmers have only considered coding their programs for, or testing the software upon, "the latest version of" this-or-that operating system. The unintended consequence of this fact is that: their latest work might not be fully compatible with earlier mixtures of software/hardware, or it might not be fully compatible with another important operating system. In any case, these differences, whatever they might be, may have resulted in (unintended...) software failures, as witnessed by some significant population of computer users.

Input combinations and preconditions:

A very fundamental problem with software testing is that testing under *all* combinations of inputs and preconditions (initial state) is not feasible, even with a simple product. This means that the number of defects in a software product can be very large and defects that occur infrequently are difficult to find in testing. More significantly, non-functional dimensions of quality (how it is supposed to *be* versus what it is supposed to *do*) -- for example, usability, scalability, performance, compatibility, reliability -- can be highly subjective; something that constitutes sufficient value to one person may be intolerable to another.

Static vs. dynamic testing:

There are many approaches to software testing. Reviews, walkthroughs or inspections are considered as static testing, whereas actually executing programmed code with a given set of test cases is referred to as dynamic testing. The former can be, (and unfortunately in practice often is...) omitted, whereas the latter takes place when programs begin to be used for the first time - which is normally considered the beginning of the testing stage. This may actually begin before the program is 100% complete in order to test particular sections of code (modules or discrete functions). For example, Spreadsheet programs are, by their very nature, tested to a large extent "on the fly" during the build process as the result of some calculation or text manipulation is shown interactively immediately after each formula is entered.

Software verification and validation:

Software testing is used in association with verification and validation:

- Verification: Have we built the software right (i.e., does it match the specification?)? It is process based.

- Validation: Have we built the right software (i.e., is this what the customer wants?)? It is product based.

The software testing team:

Software testing can be done by software testers. Until the 1950s the term "software tester" was used generally, but later it was also seen as a separate profession. Regarding the periods and the different goals in software testing, there have been established different roles: test lead/manager, test designer, tester, test automater /automation developer, and test administrator.

Software Quality Assurance (SQA):

Though controversial, software testing may be viewed as an important part of the software quality assurance (SQA) process. In SQA, software process specialists and auditors take a broader view on software and its development. They examine and change the software engineering process itself to reduce the amount of faults that end up in the delivered software: the so-called *defect rate.*

What constitutes an "acceptable defect rate" depends on the nature of the software. For example, an arcade video game designed to *simulate* flying an airplane would presumably have a much higher tolerance for defects than mission critical software such as that used to control the functions of an airliner that *really is* flying!

Although there are close links with SQA, testing departments often exist independently, and there may be no SQA function in some companies.

Software Testing is a task intended to detect defects in software by contrasting a computer program's expected results with its actual results for a given set of inputs. By contrast, QA (Quality Assurance) is the implementation of policies and procedures intended to prevent defects from occurring in the first place.

6.2 DESIGN OF TEST CASES AND SCENARIOUS

Function	**Pre – conditions**	**Expected result**	**Actual result**	**Result (pass/fail)**
Authentication	Enter Secret Key, Access Key	Connects to specified Cloud Storage if the credentials match.	Connects to Cloud and can view the files	Pass
Authentication	Enter Secret Key, Access Key	If the credentials do not match it shows the status of Connection Failed.	It shows the status of Connection Failed.	Fail
View Files Upload Download	Upload Download using Drag And Drop Functionality	Upload Download successfully.	Upload Download successfully.	Pass

6.3 VALIDATION:

The terms verification and validations are used interchangeably we will describe both these methods. Verification is the process of determining whether or not the products of given phase of software development fulfill the specifications established in the previous phase. These activities include proving and reviews. Validation is the process of evaluating the software at the end of software development process, we find how well the software satisfies the requirement specifications.

The requirement of the software starts with requirement document and requirement specifications without errors and specifying client's requirements correctly. The validation process of evaluating the developed system at the end is to ensure that it must satisfy all the necessary requirement specification. Requirement verification also checks the factors as completeness, consistency and testability of the requirements**.** As we all know that testing plays a crucial role in evaluation of the system.

That is in order to know whether the system working properly or not. In other words we can say that in order to know whether the system which we have developed will give the expected output or not can be know by doing the testing. Testing phase comes after coding phase. Usually organizations or the software developing companies use different types of testing strategies in order to evaluate the performance of a system.

Also it gives the output which provides clear information regarding the project or system , whether the project which we have developed will going to give the expected output or not , that is whether the system fails or succeed in the market.

We have many types of testing such as unit testing, integration testing, system testing, black box testing, white box testing and regression analysis testing and so on. In our project Secure Cryptographic messaging we are using unit testing, integration testing , and system testing. Unit testing is the one in which each entity or objects in the module will be tested . Once the entity is evaluated to be tested successfully than will move further with the another kind of testing. That's is once unit testing is done with all modules, than integration testing will be done, on the every module or on group of two or three modules. Finally system testing will be done , in which all the modules of a system will be tested at once , there by getting the overall performance of a system that means we can conclude the result on the entire system whether our system is working as per our requirements or as per our expectations or not.

The advantage of developing or testing modules wise is that, we can reduce the effort, cost and time. Because if we are testing module wise than we can know clearly which module is working fine and which module is not working, thereby the module which is not working perfectly can be evaluated once again by going necessary modifications unlike the system being tested on a whole, where if any errors comes in than the entire system need to be tested or evaluated which consumes more effort , time and cost.

6.4 CONCLUSION

In this way we also completed the testing phase of the project and ensured that the system is ready to go live. Thus we developed a new technology so that every user can receive and send the mail the mails.

7. CONCLUSION

This project develops an application framework that is concentrated on enhancing and facilitating the communication among the different participating entities involved in the cloud storage. The system is intended for site Users.

The project is designed and coded in such a way that any further modification that are needed in future can be easily implemented without affecting the functionality of the system. The technical documentation provided in the project report helps the application developers to understand the internal architecture of the system and thus assist them in enhancing the system.

This project is purely user-friendly and platform independent, so user can run this tool in any environment. Finally this plays a vital role in any big organization. .NET platform adds powerful automated controls with built in functionality and provides the developers a complete line of support they even desire and this is the main reason that we achieved the goal of completing this application framework.

REFERENCES

- Jeff Poise (2002). Programming Microsoft .NET. Microsoft Press.
- Asp.net general information and tutorials: http://www.asp.net
- Building an ASP.NET Intranet, Wrox publication
- Microsoft developer's references and help: http://msdn.microsoft.com/asp.net/
- Design Patterns C# by Steven metsker.
- The Unified Modeling Language- By Grady Booch
- Software Engineering- By R.S Pressman