Seed Block Algorithm: A Remote Smart Data Back-up Technique for Cloud Computing

A PROJECT REPORT

SUBMITTED BY

Nuthan N **488CS16028**

In partial fulfillment for the award of the diploma

DIPLOMA IN COMPUTER SCIENCE PROGRAMME

IN

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VIDYAVARDHAKA POLYTECHNIC

DEPARTMENT OF TECHNICAL EDUCATION

BENGALURU-560001

Year of Submission : APRIL 2019

A Project Report on

Seed Block Algorithm: A Remote Smart Data Back-up Technique for Cloud Computing

Submitted for partial fulfillment of the requirements for the award of

DIPLOMA IN COMPUTER SCIENCE AND ENGINEERING

BY

BATCH

Manjunath M 488CS16016

Nuthan N 488CS16028

Raghu M 488CS16036

Shreyas S 488CS16049

Under the Guidance of

Miss Sangeetha G B.E,M.Tech

Lecturer

Department of

Computer Science and engineering

Department of computer science

Vidyavardhaka Polytechnic

Gokulam 3rd stage

2018-2019

CANDIDATE'S DECLARATION

I, **Nuthan N** the student of Diploma in Computer Science and Engineering Department bearing Register Number **488CS16028** of **VidyaVardhaka Polytechnic**, hereby declare that, I owe full responsibility for the information,results and conclusions provided in this project work titled "**Seed Block Algorithm: A Remote Smart Data Back-up Technique for Cloud Computing**" submitted to **Board of Technical Examinations, Government of Karnataka** for the award of Diploma in Computer Science and Engineering . To the best of my knowledge, this project work has not been submitted in part or full elsewhere in any other institution/organization for the award of any certificate/diploma/degree. I have completely taken care in acknowledging the contribution of others in this academic work. I further declare that in case of any violation of intellectual property rights and particulars declared, found at any stage, I, as the candidate will be solely responsible for the same.

Date:

Place: **Signature of candidate**

Name:_____

Reg No: _____

Vidyavardhaka Polytechnic

Department of Computer Science

BONAFIDE CERTIFICATE

Certified that this project report "**Seed Block Algorithm: A Remote Smart Data Back-up Technique for Cloud Computing**" is the bonafide work of "**Nuthan N**" bearing Register No "**488CS16028**" of this institution who carried out the project work under my supervision.

SIGNATURE

Sangeetha G
Guide
DeptOf CS,VVP,
MysoreMysore

SIGNATURE

Shridhar P J
Head of Department
Dept Of CS,VVP,

DEPARTMENT OF TECHNICAL EDUCATION

Vidyavardhaka Polytechnic

Gokulam 3rd Stage, Mysore-570002

Department of Computer Science

CERTIFICATE

Certified that this project report entitled "**Seed Block Algorithm: A Remote Smart Data Back-up Technique for Cloud Computing**" which is being submitted by "**Nuthan N**" Reg. NO "**488CS16028**", a bonafide student of **VIDYAVARDHAKA POLYTECHNIC** in partial fulfilment for the award of Diploma in **Computer Science Engineering** during the year **2018-2019** is record of students own work carried out under my guidance. It is certified that all corrections/suggestions indicated for internal Assessment have been incorporated in the Report and one copy of it being deposited in the polytechnic library.

The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said diploma.

It is further understood that by this certificate the undersigned do not endorse or approve any statement made, opinion expressed or conclusion drawn there in but approve the project only for the purpose for which it is submitted.

Sangeetha G	Shridhar P.J	Shri.Krishna
Guide	**Head of Department**	**Principal**

Name and signature Examiner

1 _____

2 _____

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mentioning of the people whose constant guidance and encouragement made it possible. We take pleasure in presenting before you, our project, which is result of studied blend of both research and knowledge.

We express our earnest gratitude to our internal guide, Lecturer **Miss Sangeetha G and Department of CS**, our project guide, for his constant support, encouragement and guidance. We are grateful for his cooperation and his valuable suggestions.

Finally, we express our gratitude to all other members who are involved either directly or indirectly for the completion of this project.

DECLARATION

We, the undersigned, declare that the project entitled **'Seed Block Algorithm: A Remote Smart Data Back-up Technique for Cloud Computing'**, being submitted in partial fulfillment for the award of Diploma in Computer Science and Engineering, affiliated to Board of technical education, is the work carried out by us.

Nuthan N **488CS16028**

Log sheet of 5th Sem for project work-1

SL No	Date	Task Of Students	Progress Of Task	Initial Of Staff	Evaluation
1.		Discussion about various technologies to develop project	Awareness regarding various technologies		**Completed**
2.		Seminar on various technologies	Awareness regarding various technologies		**Completed**
3.		PPT of web based windows based and mobile based project	Awareness regarding different types of projects		**Completed**
4.		Submissionof the actual project synopsis of 5thsem	Awareness regarding synopsis		**Completed**
5.		Presentation with seminar on synopsis by each member in the team	Awareness regarding presentation		**Completed**

CONTENTS

LIST OF FIGURES

ABSTRACT

In cloud computing, data generated in electronic form are large in amount. To maintain this data efficiently, there is a necessity of data recovery services. The proposed system user a Seed Block Algorithm, which is a smart remote data back up algorithm. To procure this, in this paper this propose a smart remote data backup algorithm, Seed Block Algorithm. The objective of proposed algorithm is twofold; first as it help the users to collect information from any remote location in the absence of network connectivity and secondly it allows to recover the files in case of the file deletion or if the cloud gets destroyed due to any reason. The time related issues are also being solved by proposed SBA such that it will take minimum time for the recovery process. Proposed SBA also focuses on the security concept for the back-up files stored at remote server, without using any of the existing encryption techniques.

CHAPTER 1

INTRODUCTION

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name comes from the common use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation. Cloud computing consists of hardware and software resources made available on the Internet as managed third-party services. These services typically provide access to advanced software applications and high-end networks of server computers.

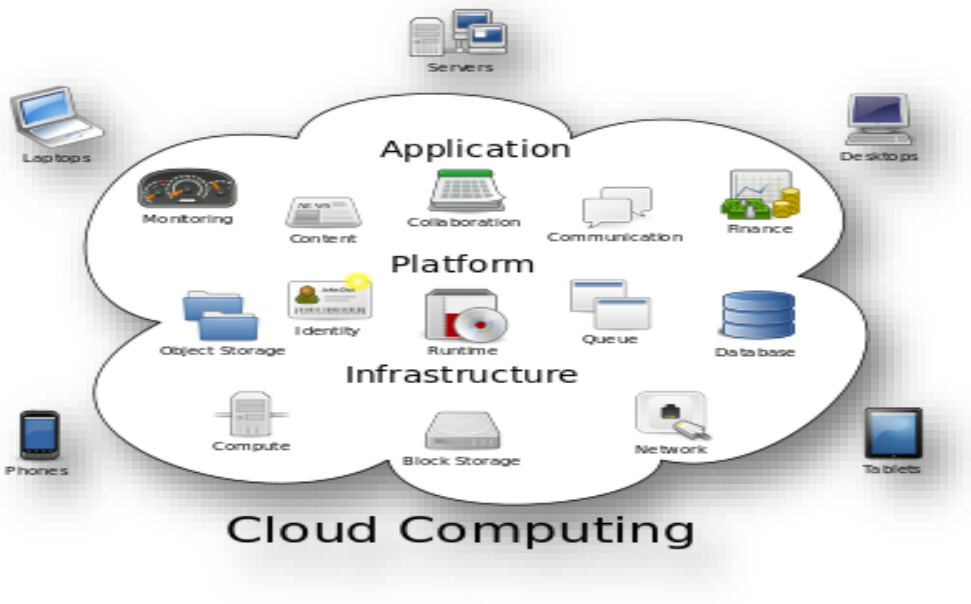


Fig1.1:-Structure of cloud computing

Cloud Computing working:

The goal of cloud computing is to apply traditional supercomputing, or high-performance computing power, normally used by military and research facilities, to perform tens of trillions of computations per second, in consumer-oriented applications such as financial portfolios, to deliver personalized information, to provide data storage or to power large, immersive computer games.

The cloud computing uses networks of large groups of servers typically running low-cost consumer PC technology with specialized connections to spread data-processing chores across them. This shared IT infrastructure contains large pools of systems that are linked together. Often, virtualization techniques are used to maximize the power of cloud computing.

Characteristics and Services Models:

The salient characteristics of cloud computing based on the definitions provided by the National Institute of Standards and Terminology (NIST) are outlined below:

- **On-demand self-service**: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service's provider.
- **Broad network access**: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).
- **Resource pooling**: The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different

physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location-independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or data center). Examples of resources include storage, processing, memory, network bandwidth, and virtual machines.

- **Rapid elasticity**: Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.
- **Measured service**: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be managed, controlled, and reported providing transparency for both the provider and consumer of the utilized service.

Fig1.2:- Five Essential Characteristics of cloud computing

Services Models:

Cloud Computing comprises three different service models, namely Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). The three service models or layer are completed by an end user layer that encapsulates the end user perspective on cloud services. The model is shown in figure below. If a cloud user accesses services on the infrastructure layer, for instance, she can run her own applications on the resources of a cloud infrastructure and remain responsible for the support, maintenance, and security of these applications herself. If she accesses a service on the application layer, these tasks are normally taken care of by the cloud service provider.

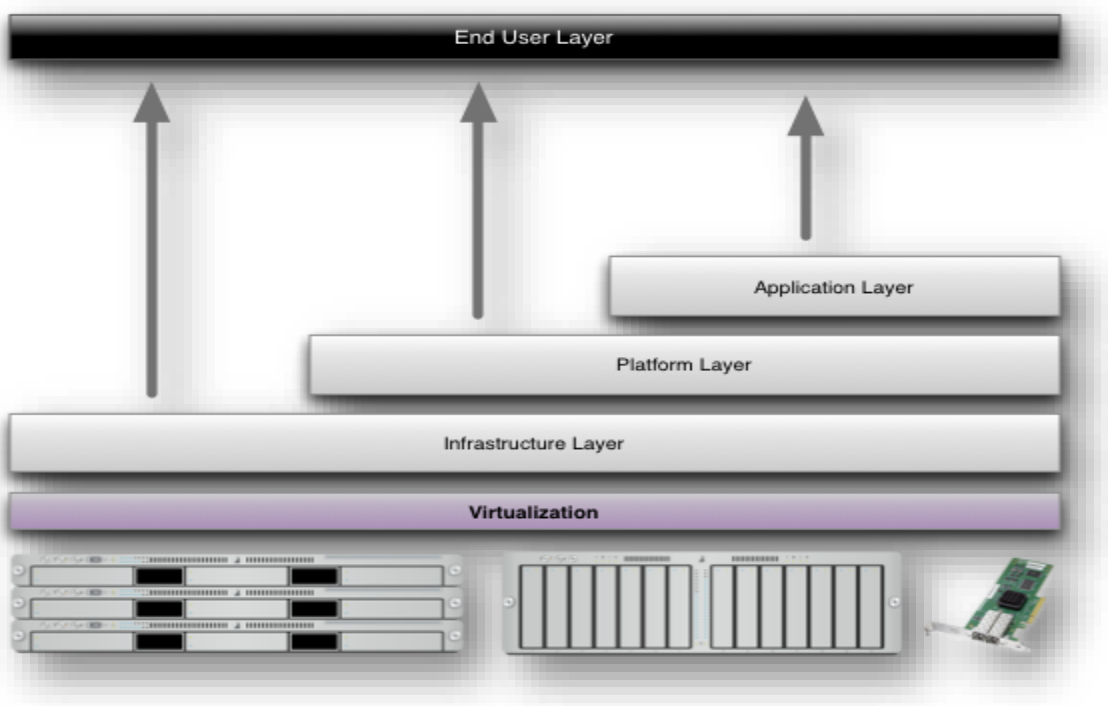


Fig1.3:-Structure of service models

Benefits of cloud computing:

1. **Achieve economies of scale** – increase volume output or productivity with fewer people. Your cost per unit, project or product plummets.
2. **Reduce spending on technology infrastructure.** Maintain easy access to your information with minimal upfront spending. Pay as you go (weekly, quarterly or yearly), based on demand.
3. **Globalize your workforce on the cheap.** People worldwide can access the cloud, provided they have an Internet connection.
4. **Streamline processes.** Get more work done in less time with less people.
5. **Reduce capital costs.** There's no need to spend big money on hardware, software or licensing fees.
6. **Improve accessibility.** You have access anytime, anywhere, making your life so much easier!
7. **Monitor projects more effectively.** Stay within budget and ahead of completion cycle times.
8. **Less personnel training is needed.** It takes fewer people to do more work on a cloud, with a minimal learning curve on hardware and software issues.
9. **Minimize licensing new software.** Stretch and grow without the need to buy expensive software licenses or programs.
10. **Improve flexibility.** You can change direction without serious "people" or "financial" issues at stake.

Advantages:

1. **Price:**Pay for only the resources used.
2. **Security**: Cloud instances are isolated in the network from other instances for improved security.

3. **Performance:** Instances can be added instantly for improved performance. Clients have access to the total resources of the Cloud's core hardware.
4. **Scalability:** Auto-deploy cloud instances when needed.
5. **Uptime:** Uses multiple servers for maximum redundancies. In case of server failure, instances can be automatically created on another server.
6. **Control:** Able to login from any location. Server snapshot and a software library lets you deploy custom instances.
7. **Traffic:** Deals with spike in traffic with quick deployment of additional instances to handle the load.

Seed Block Algorithm:-

This algorithm focuses on simplicity of the back-up and recovery process. It basically uses the concept of Exclusive– OR (XOR) operation of the computing world. For ex: - Suppose there are two data files: A and B. When we XOR A and B it produced X i.e. X = A ⊕B. If suppose A data file get destroyed and we want our A data file back then we are able to get A data file back, then it is very easy to get back it with the help of B and X data file .i.e. A =X ⊕B .

CHAPTER 2

LITERATURE SURVEY

[1.] Distributed index for matching multimedia objects Authors: A. Abdelsadek

This paper presents the design and evaluation of DIMO, a distributed system for matching high-dimensional multimedia objects. DIMO provides multimedia applications with the basic function of computing the K nearest neighbors on large-scale datasets. It also allows multimedia applications to define application-specific functions to further process the computed nearest neighbors. DIMO presents a novel method for partitioning, searching, and storing high-dimensional datasets on distributed infrastructures that support the MapReduce programming model. We have implemented DIMO and extensively evaluated it on Amazon clusters with number of machines ranging from 8 to 128. We have experimented with large datasets of sizes up to 160 million data points extracted from images, and each point has 128 dimensions. Our experimental results show that DIMO: (i) results in high precision when compared against the ground-truth nearest neighbors, (ii) can elastically utilize varying amounts of computing resources, (iii) does not impose high network overheads, (iv) does not require large main memory even for processing large datasets, and (v) balances the load across the used computing machines. In addition, DIMO outperforms the closest system in the literature by a large margin (up to 20%) in terms of the achieved average precision of the computed nearest neighbors. Furthermore, DIMO requires at least three orders of magnitudes less storage than the other system, and it is more computationally efficient.

[2.] Distributed Kd-Trees for retrieval from very large image collections Authors: M. Aly, M. Munich, and P. Perona

Distributed Kd-Trees is a method for building image retrieval systems that can handle hundreds of millions of images. It is based on dividing the Kd-Tree into a "root subtree" that resides on a root machine, and several "leaf subtrees", each residing on a leaf ma-chine. The root machine handles incoming queries and farms out feature matching to an appropriate small subset of the leaf machines. Our implementation employs the Map Reduce architecture to efficiently build and distribute the Kd-Tree for millions of images. It can run on thousands of machines, and provides orders of magnitude more through-put than the state-of-the-art, with better recognition performance. We show experiments with up to 100 million images running on 2048 machines, with run time of a fraction of a second for each query image.

[3.] Multidimensional binary search trees used for associative searching Authors: J. Bentley

This paper develops the multidimensional binary search tree (or k-d tree, where k is the dimensionality of the search space) as a data structure for storage of information to be retrieved by associative searches. The k-d tree is defined and examples are given. It is shown to be quite efficient in its storage requirements. A significant advantage of this structure is that a single data structure can handle many types of queries very efficiently. Various utility algorithms are developed; their proven average running times in an n record file are: insertion, O(log n); deletion of the root, O(n(k-1)/k); deletion of a random node, O(log n); and optimization (guarantees logarithmic performance of searches), O(n log n). Search algorithms are given for partial match queries with t keys specified [proven maximum running time of O(n(k-t)/k)] and for nearest neighbor queries [empirically observed average running time of O(log n).] These performances

far surpass the best currently known algorithms for these tasks. An algorithm is presented to handle any general intersection query. The main focus of this paper is theoretical. It is felt, however, that k-d trees could be quite useful in many applications, and examples of potential uses are given.

[4.] Map Reduce: Simplified data processing on large clusters Authors: J. Dean and S. Ghemawat

Map Reduce is a programming model and an associated implementation for processing and generating large datasets that is amenable to a broad variety of real-world tasks. Users specify the computation in terms of a map and a reduce function, and the underlying runtime system automatically parallelizes the computation across large-scale clusters of machines, handles machine failures, and schedules inter-machine communication to make efficient use of the network and disks. Programmers find the system easy to use: more than ten thousand distinct Map Reduce programs have been implemented internally at Google over the past four years, and an average of one hundred thousand Map Reduce jobs are executed on Google's clusters every day, processing a total of more than twenty peta bytes of data per day.

[5.] Watermarking techniques for intellectual property protection Authors: A. Kahng, J. Lach, W. Mangione-Smith, S. Mantik, I. Markov, M. Potkonjak, P. Tucker, H. Wang, and G. Wolfe

Digital system designs are the product of valuable effort and know-how. Their embodiments, from software and HDL program down to device-level netlist and mask data, represent carefully guarded intellectual property (IP). Hence, design methodologies based on IP reuse require new mechanisms to protect the rights of IP producers and owners. This paper establishes principles of watermarking-based IP protection, where a watermark is a mechanism for identification that is

(i) nearly invisible to human and machine inspection, (ii) difficult to remove, and (iii) permanently embedded as an integral part of the design. We survey related work in cryptography and design methodology, then develop desiderata, metrics and example approaches — centering on constraint-based techniques — for watermarking at various stages of the VLSI design process.

CHAPTER 3

SYSTEM REQUIREMENTS:

HARDWARE REQUIREMENTS:

- System : Pentium IV 2.4 GHz.
- Hard Disk : 40 GB.
- Floppy Drive : 1.44 Mb.
- Monitor : 15 VGA Colour.
- Mouse : Logitech.
- Ram : 512 Mb.

SOFTWARE REQUIREMENTS:

- Operating system : - Windows XP.
- Coding Language : ASP.NET, C#
- Data Base : MS SQL SERVER 2005

CHAPTER 4

SYSTEM ANALYSIS

4.1 EXISTING SYSTEM:

The recent back-up and recovery techniques that have been developed in cloud computing domain such as HSDRT, PCS, ERGOT, Linux Box, Cold/Hot backup strategy etc. Detail review shows that none of these techniques are able to provide best performances under all uncontrolled circumstances such as cost, security, low implementation complexity, redundancy and recovery in short span of time.

DISADVANTAGES OF EXISTING SYSTEM:

1. High Implementation cost
2. Low security
3. Reduced Privacy
4. High implementation complexity
5. Poor Recovery
6. Redundancy
7. Time Complexity

4.2 PROPOSED SYSTEM:

The objective of proposed algorithm is two types; first it help the users to collect information from any remote location in the absence of network connectivity and second to recover the files in case of the file deletion or if the cloud gets destroyed due to any reason.

ADVANTAGES OF PROPOSED SYSTEM:

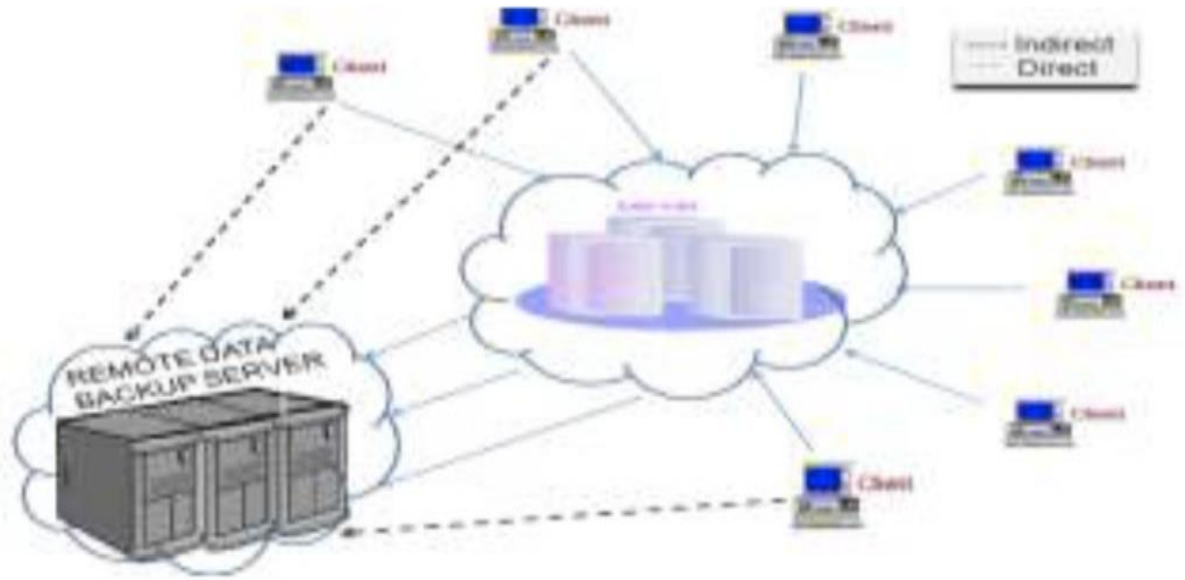
1. Recover same size data
2. Low cost
3. Good Privacy
4. Reduced Time Complexity
5. High Security
6. Good Storage Facilities
7. Backup and Recovery in short span of time.

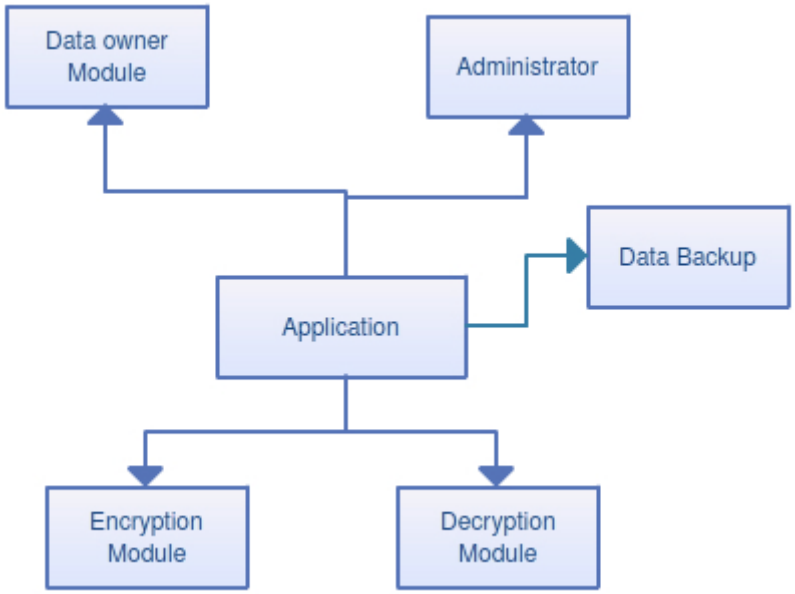
This algorithm focuses on simplicity of the back-up and recovery process. It basically uses the concept of Exclusive– OR (XOR) operation of the computing world. For ex: - Suppose there are two data files: A and B. When we XOR A and B it produced X i.e. X = A ⊕B. If suppose A data file get destroyed and we want our A data file back then we are able to get A data file back, then it is very easy to get back it with the help of B and X data file .i.e. A =X ⊕B .

CHAPTER 5

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE:

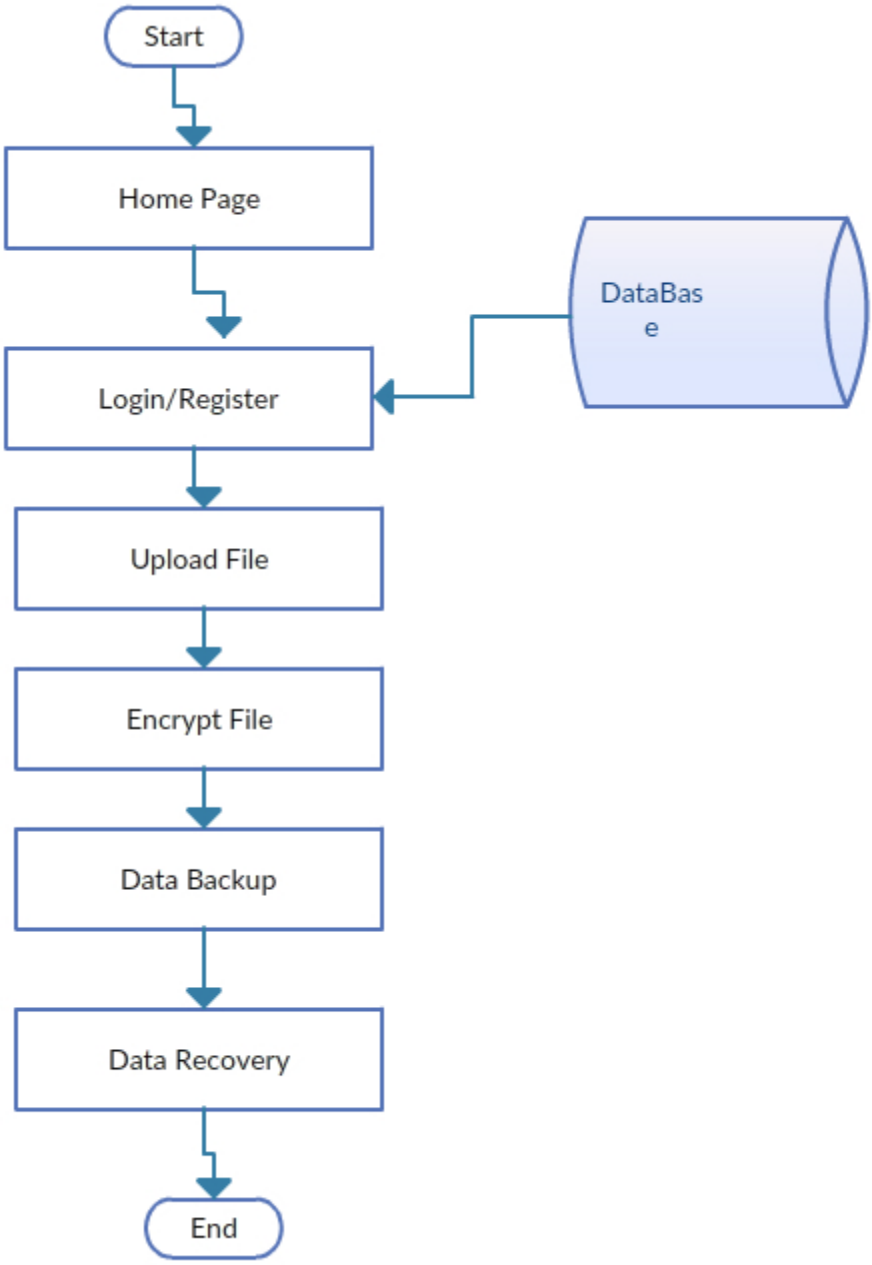




5.2 DATA FLOW DIAGRAM:

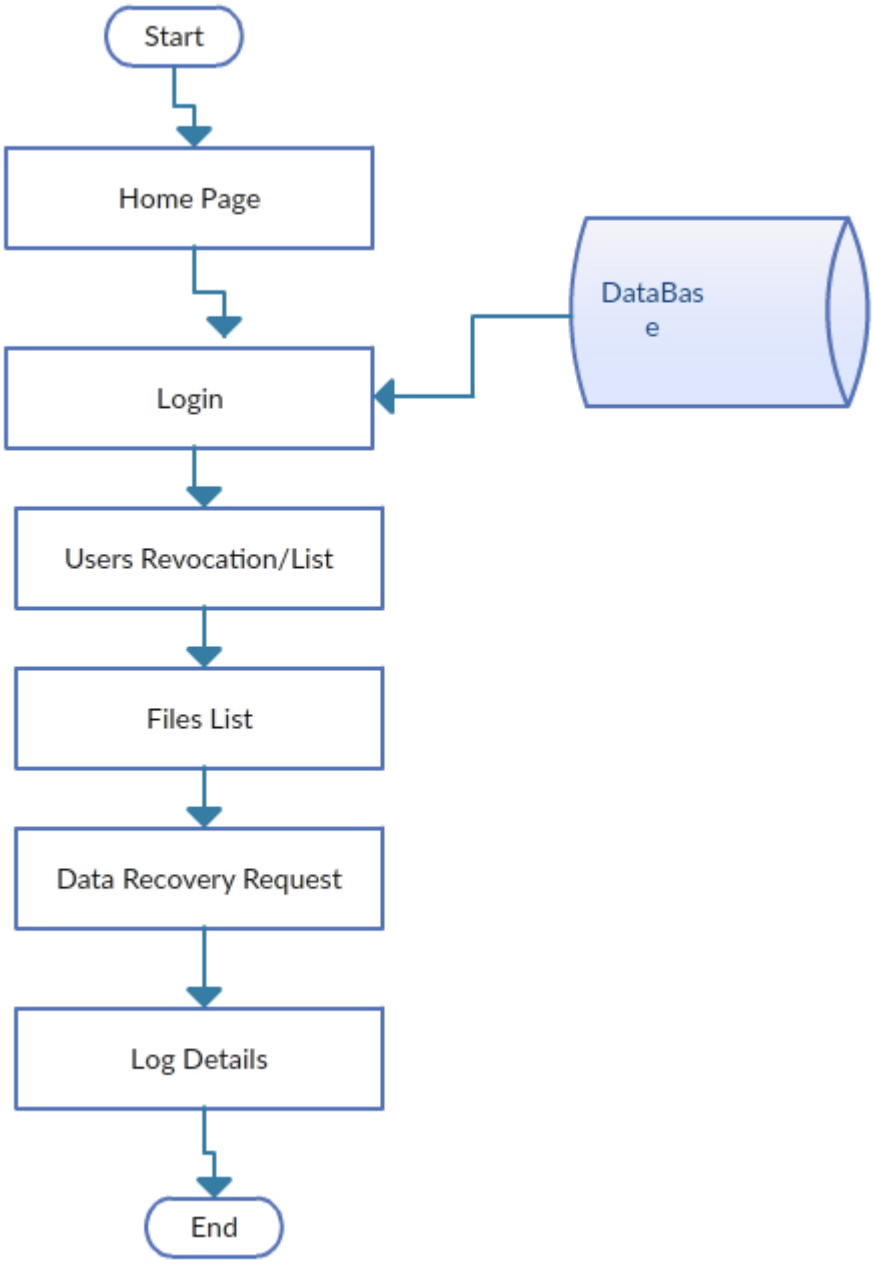
1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

5.3 Data Owner



The user register by giving the few details, after registration, The user details is stored in database, The user logs in and upload the file to be stored in cloud. The uploaded file is encrypted and it is sent to cloud. The stored file can be recovered.

5.4 Administrator



The admin logs and checks for all the user's revocation request from the revocation list and serves for the requested users. The requested file is recovered to the required user. The admin also checks the log details of each user.

5.5 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software

engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling 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.

GOALS:

The Primary goals in the design of the UML are as follows:

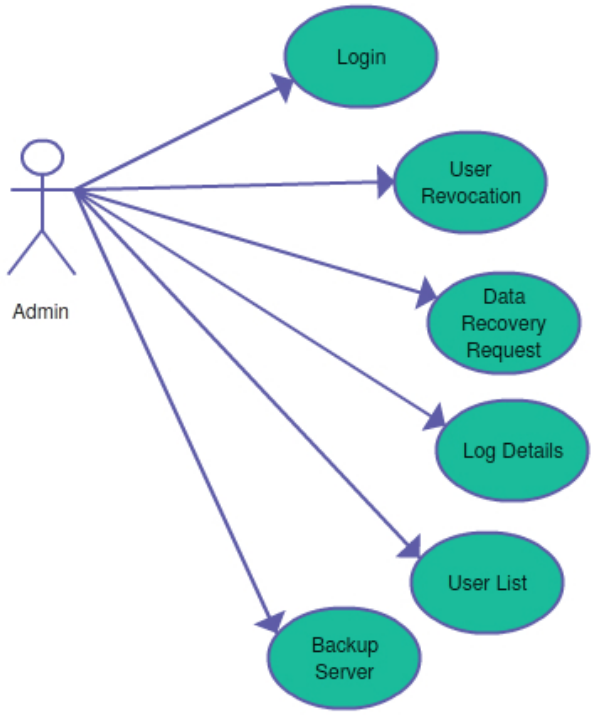
1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

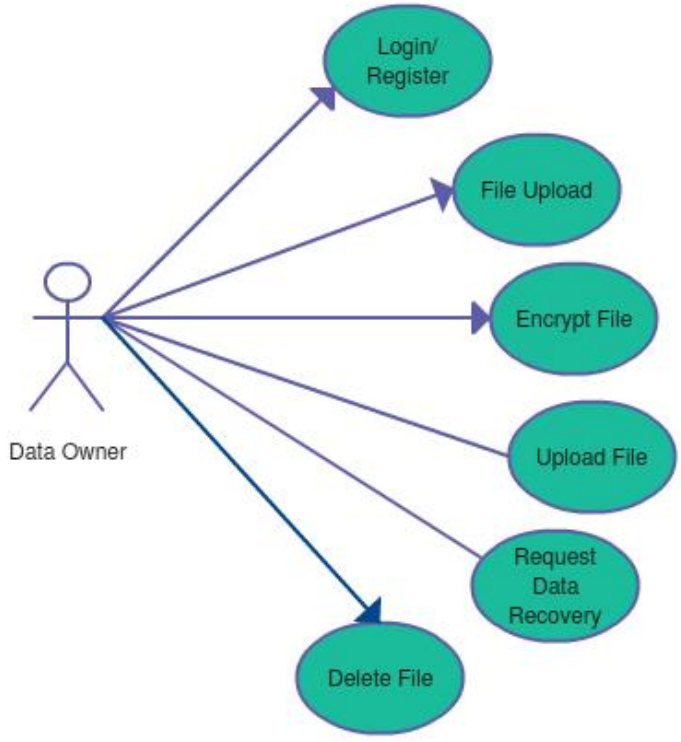
5.6 USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

The user register by giving the few details, after registration, The user details is stored in database, The user logs in and upload the file to be stored in cloud. The uploaded file is encrypted and it is sent to cloud. The stored file can be recovered.

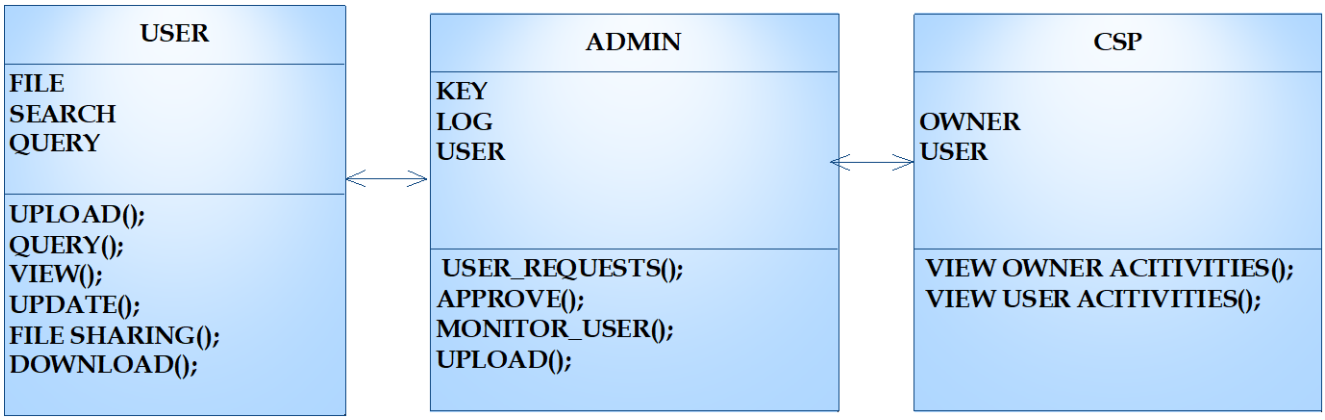
The admin logs and checks for all the user's revocation request from the revocation list and serves for the requested users. The requested file is recovered to the required user. The admin also checks the log details of each user.


Login
User Revocation
Data Recovery Request
Log Details
User List
Backup Server
Admin


Login/ Register
File Upload
Encrypt File
Upload File
Request Data Recovery
Delete File
Data Owner

5.7 CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

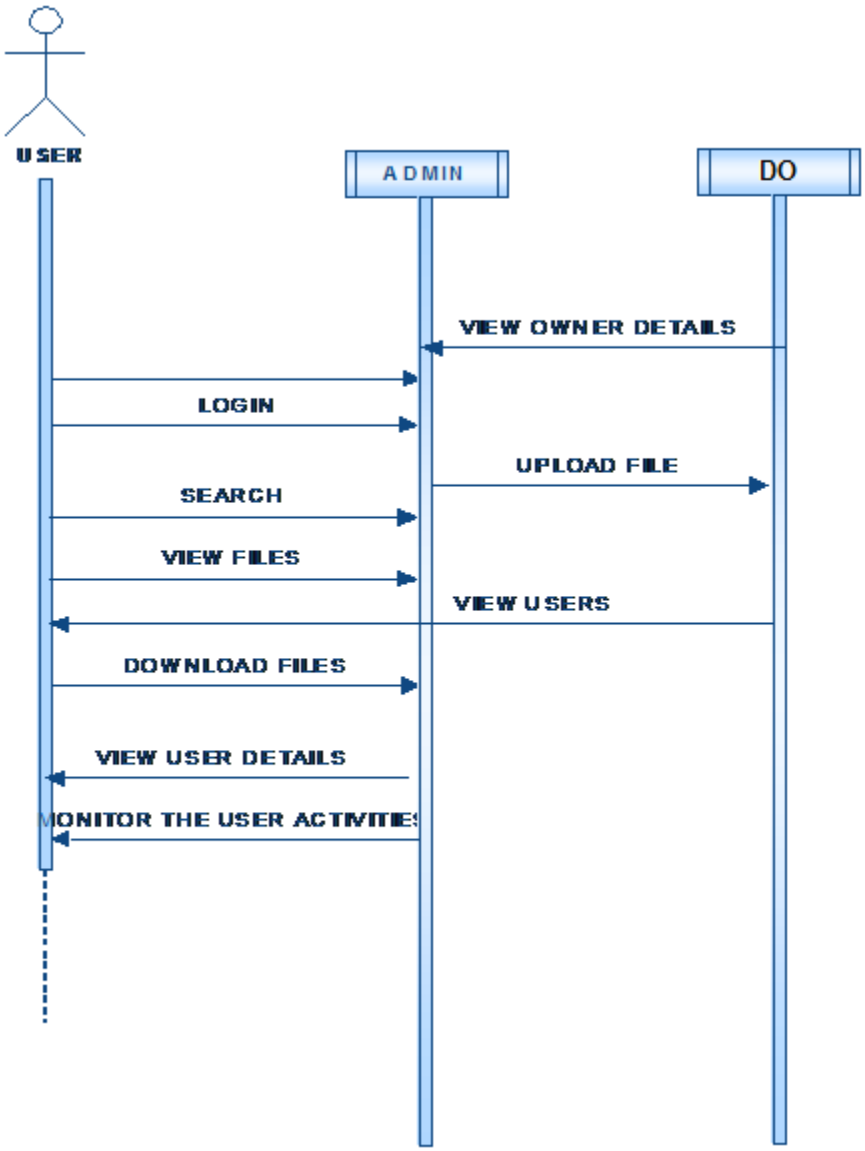


5.8 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

The user log in and search for file which is uploaded. Then the user will download the file and checks the user details. The user also monitors the user activity.

The admin will view the user details and they can also have the rights to upload file and check the user activity.


USER
ADMIN
DO
VIEW OWNER DETAILS
LOGIN
UPLOAD FILE
SEARCH
VIEW FILES
VIEW USERS
DOWNLOAD FILES
VIEW USER DETAILS
MONITOR THE USER ACTIVITIES

CHAPTER 6

IMPLEMENTATION

6.1 MODULES:

1. Data owner Module
    a. File Upload
    b. File Encryption
    c. View File
    d. Delete File
    e. Recover File
2. Administrator
3. Encryption & Decryption Module
4. Data Backup

MODULES DESCRIPTION:

6.1.1 Data owner Module

Data owner is the act of having legal rights and complete control over a single piece or set of data elements. It defines and provides information about the rightful owner of data assets and the acquisition, use and distribution policy implemented by the data owner. Data owner is primarily a data governance process that details an organization's legal ownership of enterprise-wide data. A specific organization or the data owner has the ability to create, edit, modify, share and restrict access to the data.

File Upload

This is module we have provided a option to data owner to upload the data to cloud

File Encyption

This module is designed for the securtity resion where every file uploaded by the data owner will be encrypted and stored in cloud.

View File

This module supports the data owner to view the list of files uploaded & deleted by data owner.

Delete File

This module supports the data owner to delete his files.

Recover File

The main concept of the project is to recover the deleted data from backups, this module support to recover the data deleted from the data owner.

6.1.2 Administrator Module

The administrator is a module where having complete rights of the application. The following are the operation handled by the administrator

1. User Revocation
2. File Recovery
3. Users List
4. User Log Details

User Revocation

The module supports the administrator to block & unblock the data owner account.

File Recovery

This module supports to view all the data recovery requests sent by the data owner

6.1.3 Encryption & Decryption Module

Algorithm

1) **Encryption**

Step 1: Load file with file name

Step 2: Pass text into string 'clearText'

Step 3: Encrypt contents of variable clearText and store in string 'cipher Text'

Step 4: Append cipherText into string builder 'sb'

Step 5: Encrypt key using either RSA or UTF8 encoding

Step 6: For each x<-0 to EncryptArray.Length

6.1Append EncryptArray to ' sb'

End for

Step 7: For each c<-0 to resultArray.Length

7.1Append resultArray to 'sb'

End for

Step 8: Upload Encrypt key, file size, extension path of file

Step 9: Load file with encrypted format

Step 10: Upload file

The data owner externalizes the data to the cloud server. Firstly this algorithm will load the file of that path of the file. Pass the text in to the string "clearText". And encrypt all the contents of the variable "clearText" and store it in the string format "ciphertext". Adding at the end of the file of "cipherText"

into string builder “sb”. It generates encryption key using RSA and UTF8 encoding. It evaluates the length of Encrypted file using array and added at the end of the file “cipherText” to string builder “sb”. And the result of that file is also evaluate the length of that file using array and added at the end of the file “cipherText” to string builder “sb”. Later the file will be uploaded to the cloud server.

Decryption

Step 1: Load the byte code of string

Step 2: Configure file

Step 3: Load the key

Step 4: Decrypt the key using either RSA or UTF8 encoding

Step 5: For each x<-0 to EncryptArray.Length

5.1Append EncryptArray to ‘ sb’

End for

Step 6: For each c<-0 to resultArray.Length

6.1Append resultArray to ‘sb’

End for

Step 7: File retrieved

In this algorithm when the data user selects the file to download it will load the file with file name and contents of file of string. Configure of file is done to get the key to open the file. Before that the key will be load that is encryption key.The encryption key is received from cloud server through email. The key is decrypted using RSA or UTF8 encoding. It evaluates the length of Encrypted file using array and added at the end of the file to string builder “sb”. And the result of that file is also evaluate the length of that file using array and added at

the end of the file “cipher Text” to string builder “sb”. And at last file is retrieved as decryption format.

6.1.4 Data Backup

Backup server of main cloud means the copy of main cloud. When this Backup server is far away from the main server i.e. at remote location and having the complete state of the main cloud, then it is called as Remote Data Backup Server. The main cloud is called as the central repository. Remote backup cloud is termed as remote repository.

Proposed algorithm

This algorithm basically uses the concept of Exclusive–OR (XOR) operation of computation. For ex. Consider two data files: P and Q. When we XOR P and Q, it produced X i.e. X = P Q . If suppose P data file is deleted and we want our P data file back then we can get it with the help of Q and X data file .i.e. P = X Q. In this algorithm, first we set a random number in the cloud and unique client id for every client. Second, whenever the client id is being register in the main cloud; then client id and random number is getting EXORed ( ) with each other to generate seed block for the particular client. The generated seed block corresponds to each client is stored at remote server. Whenever client creates the file in cloud first time, it is stored at the main cloud. When it is stored in main server, the main file of client is being EXORed with the Seed Block of the particular client.

It is also encrypted using public key RSA and that output file is stored at the remote server in the form of file’ (pronounced as File dash). If either unfortunately file in main cloud crashed /damaged or file is been deleted mistakenly, then the user will get the original file. For that, first, using the private key of the user, file‟ is decrypted and then by EXORing file’ with the

seed block of the corresponding client, user can produce the original file and return the resulted file i.e. original file back to the requested client. This encryption operation is used to support security as well as file sharing application such that system user can share his file to another authenticated user providing him private key via email to decrypt file".

Initialization: Main Cloud: Mc; Remote Server: Rs ;

Clients of Main Cloud: Ci; Files: a1 and a'1;

Seed block: Si; Random Number: r ;

Client's ID: Client_ldi

Input: a1 created by Ci; r is generated at Mc ;

Output: Recovered file A1 after deletion at Mc

Given: Authenticated clients could allow uploading,

downloading and do modification on its the files only.

Step 1 : Generate a random number.

Int r=rand( );

Step 2: Create a seed Block Si for each Ci and Store

Si at Rs

Si = r ⊕ Client_Idi (Repeat step 2 for all clients)

Step 3: If Ci / Admin creates/modifies a a1 and Stores at

Mc , then create as

A1'= a1 ⊕ Si

Step 4: Store a' at Rs .

Step 5: If server crashes a1 deleted from Mc

then, we do EXOR to retrieve the original a1 as:

A1 = a'1 ⊕ Si

Step 6: Return a1 to Ci.

Step 7: END.

CHAPTER 7

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTING

7.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

7.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7.3 Functional testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

7.4 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

7.5 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

7.6 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

7.7 Testing results in project

Sl.no	Test Case name	Test case details	Input	Expected output	Actual Output
1	User Login	User ID and Password	If accepted based on the details	Login must be done	Login id done
2	Admin Login	User validation	Validating user name and password	To be allowed to login	Allowed to login
3	Upload file	Uploading of file	Upload	File must be uploaded	File was uploaded

CHAPTER 8

INPUT AND OUTPUT DESIGN

8.1 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.

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 that is easy to follow

8.2 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.

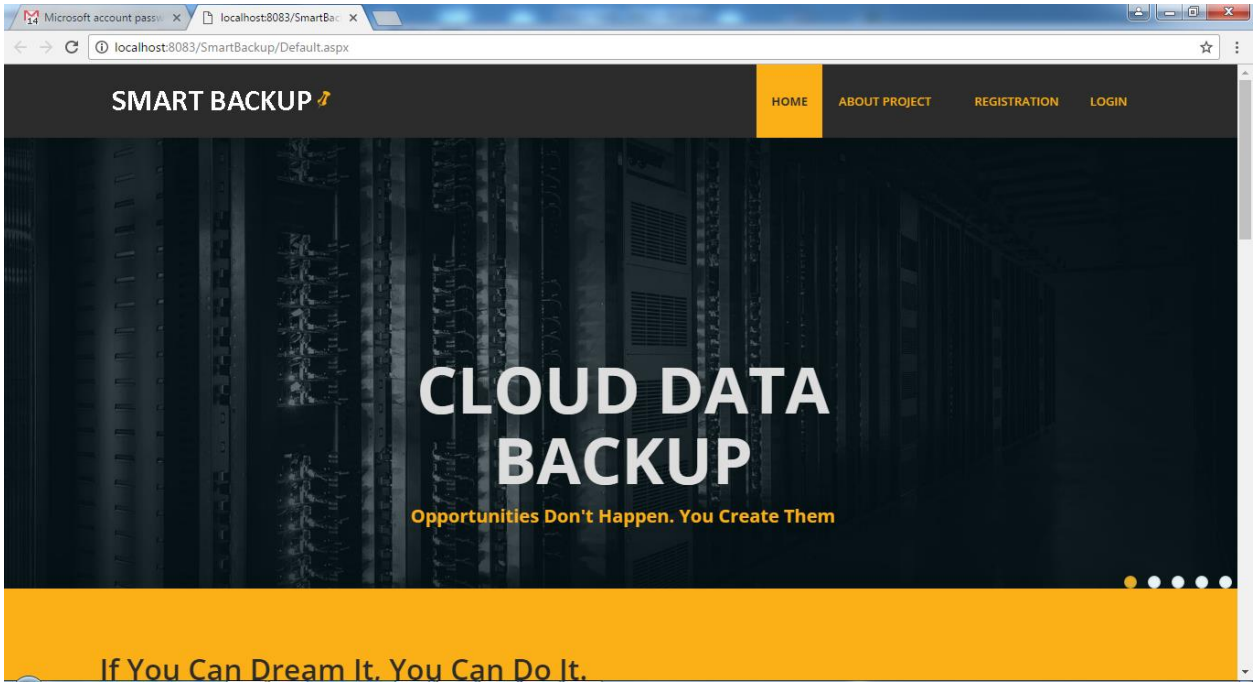
The output form of an information system should accomplish one or more of the following objectives.

- 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.

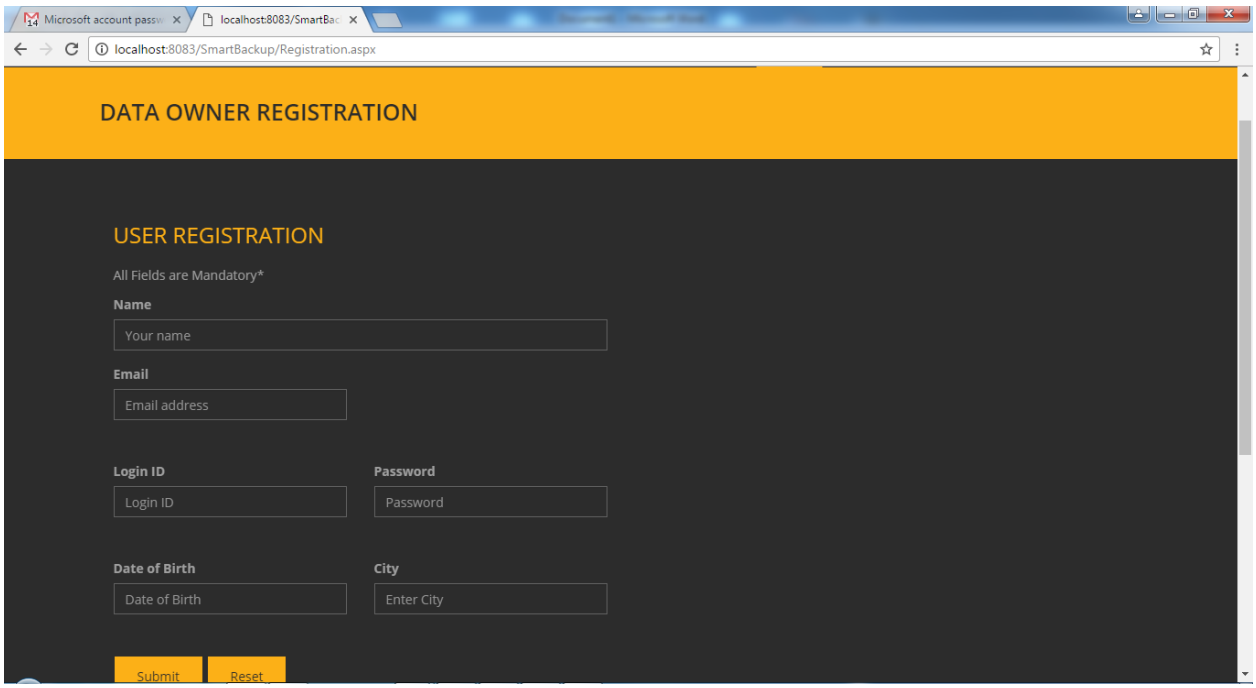
CHAPTER 9

SNAPSHOTS

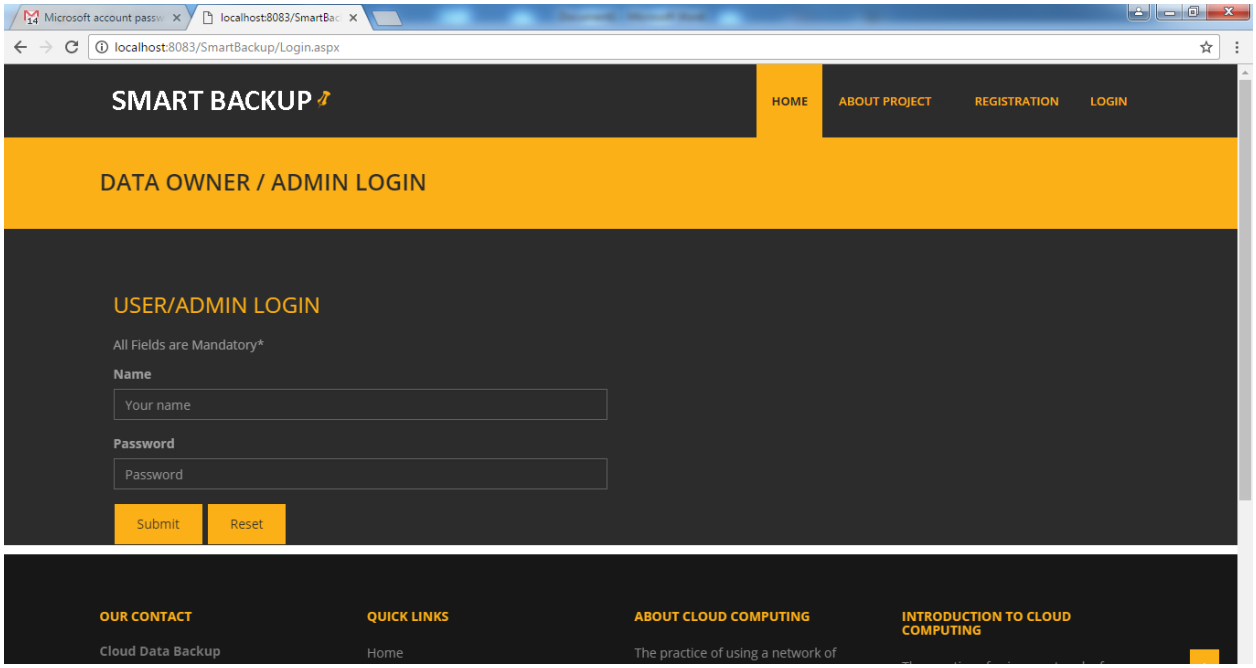
Home Page



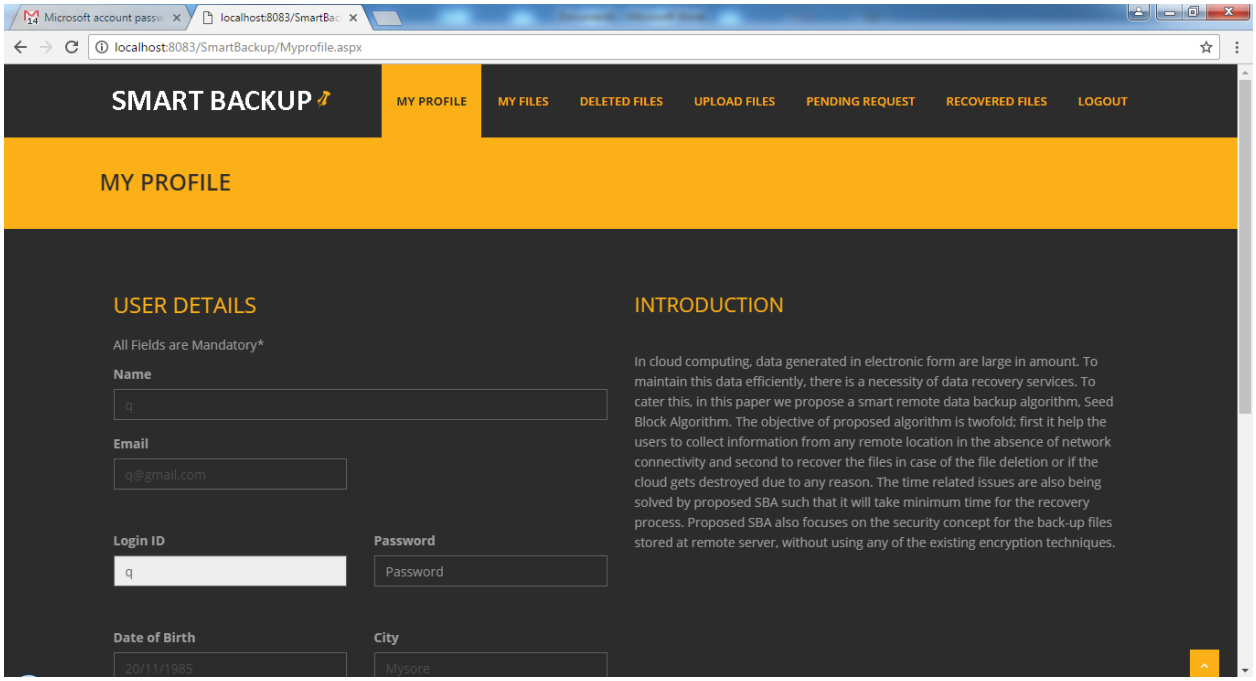
Data Owner Registration



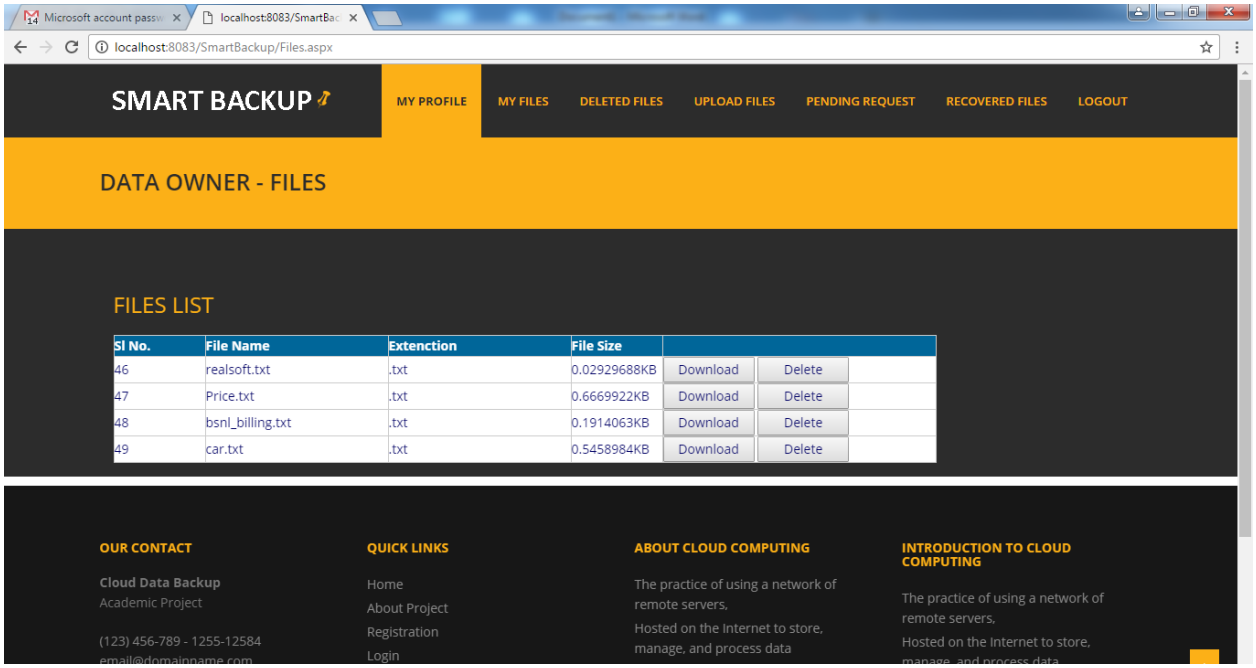
User / Admin **Login**



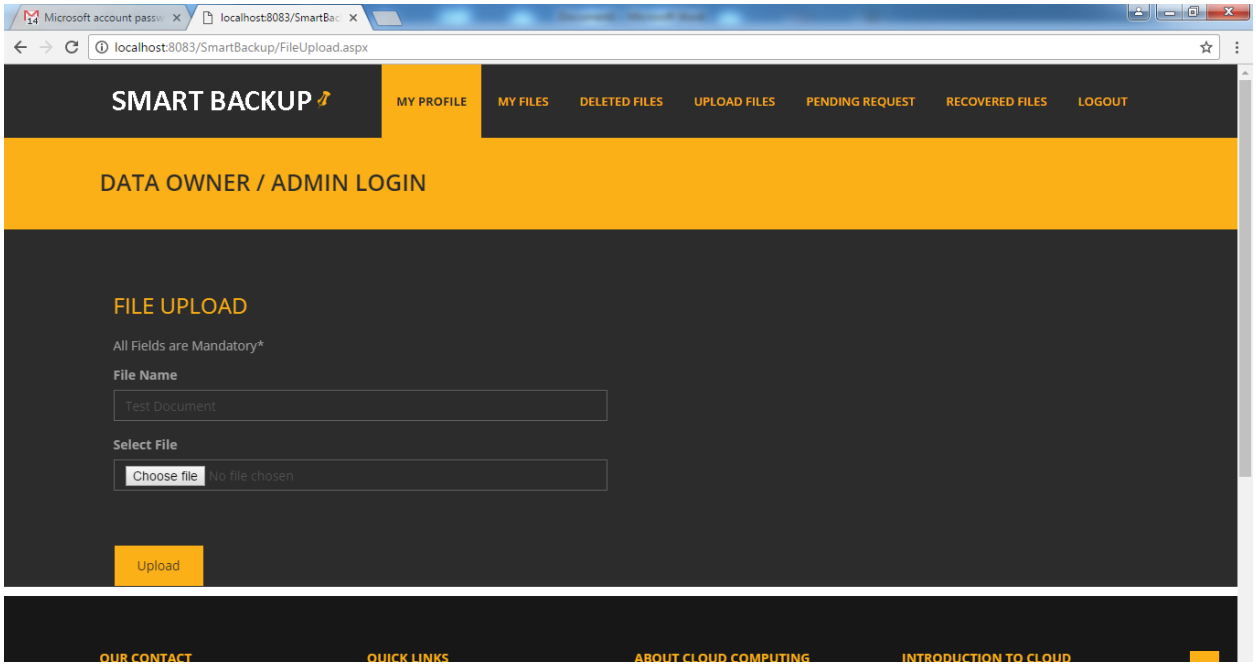
User Dashboard



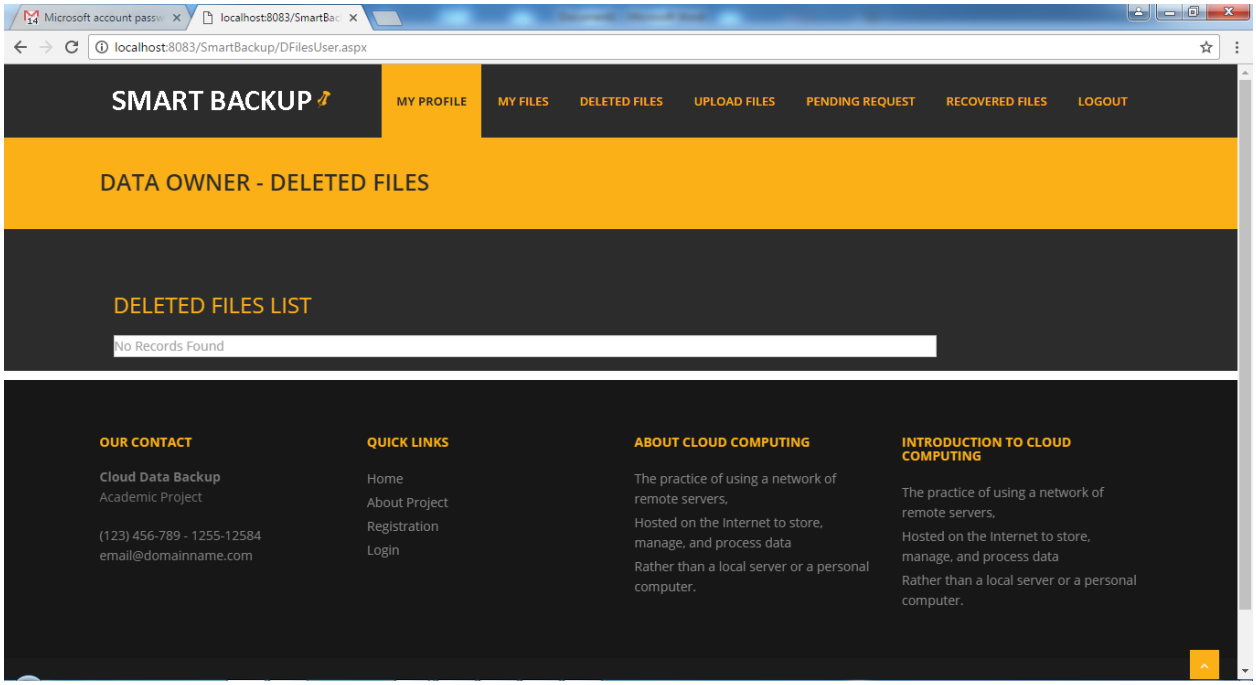
Files List



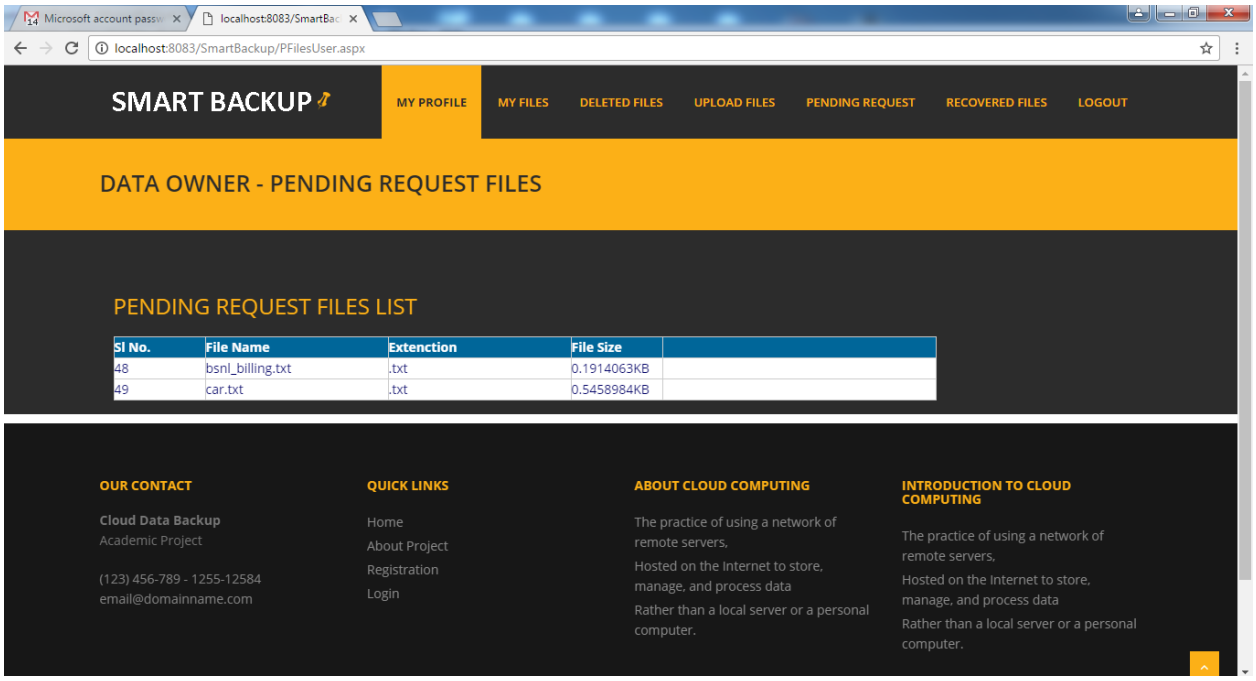
Upload Files



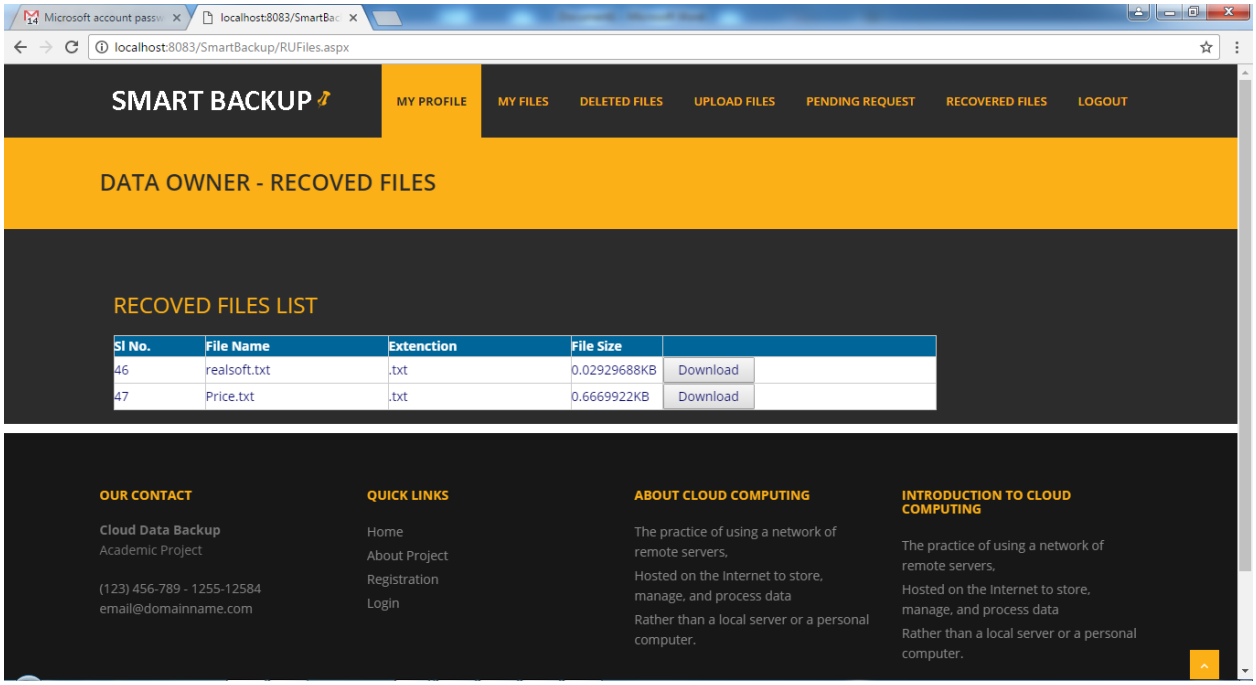
Deleted Files List



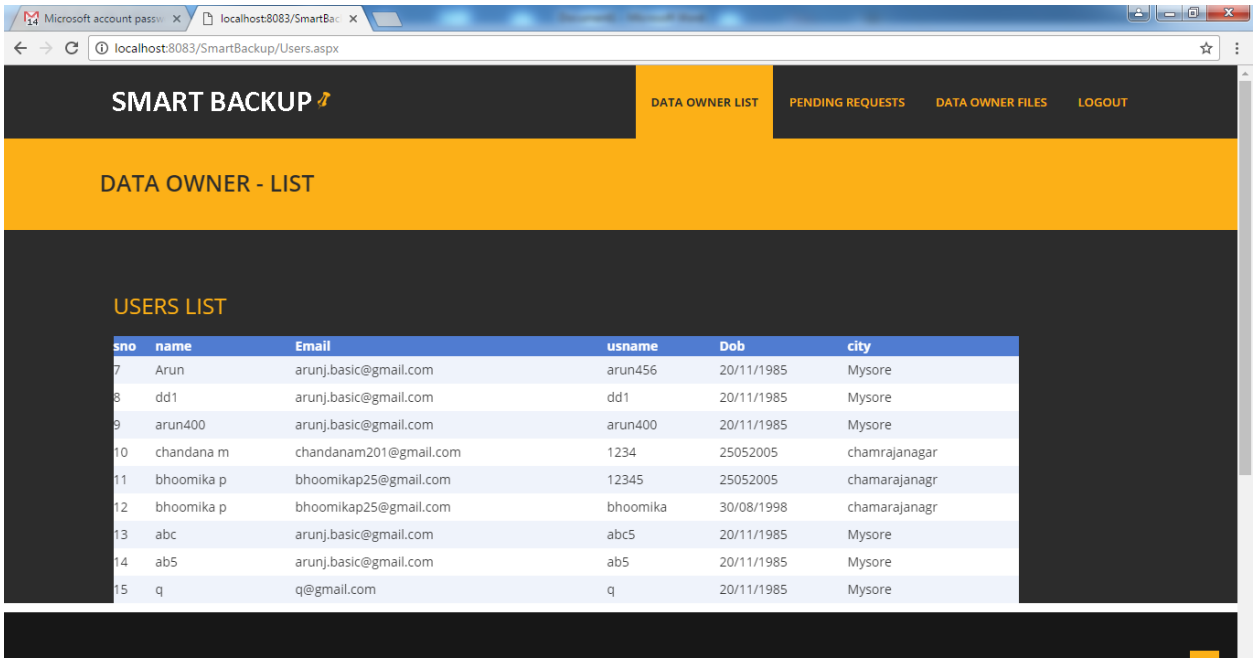
Pending Request



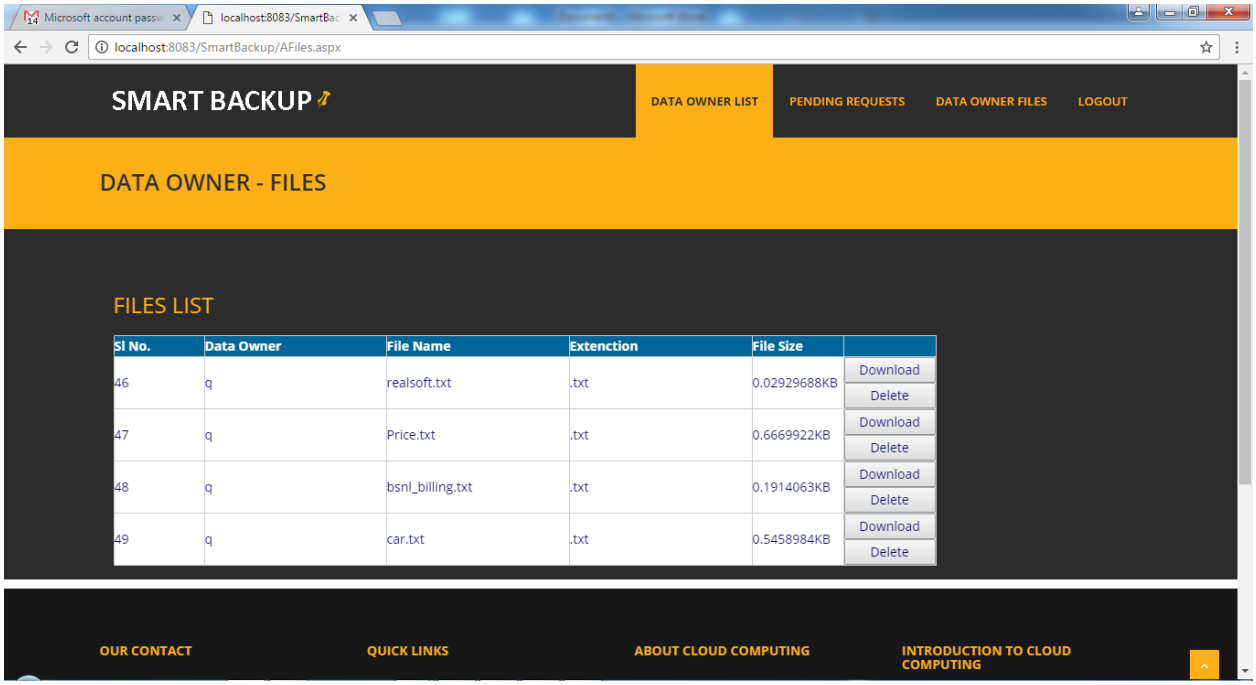
Recovered Files



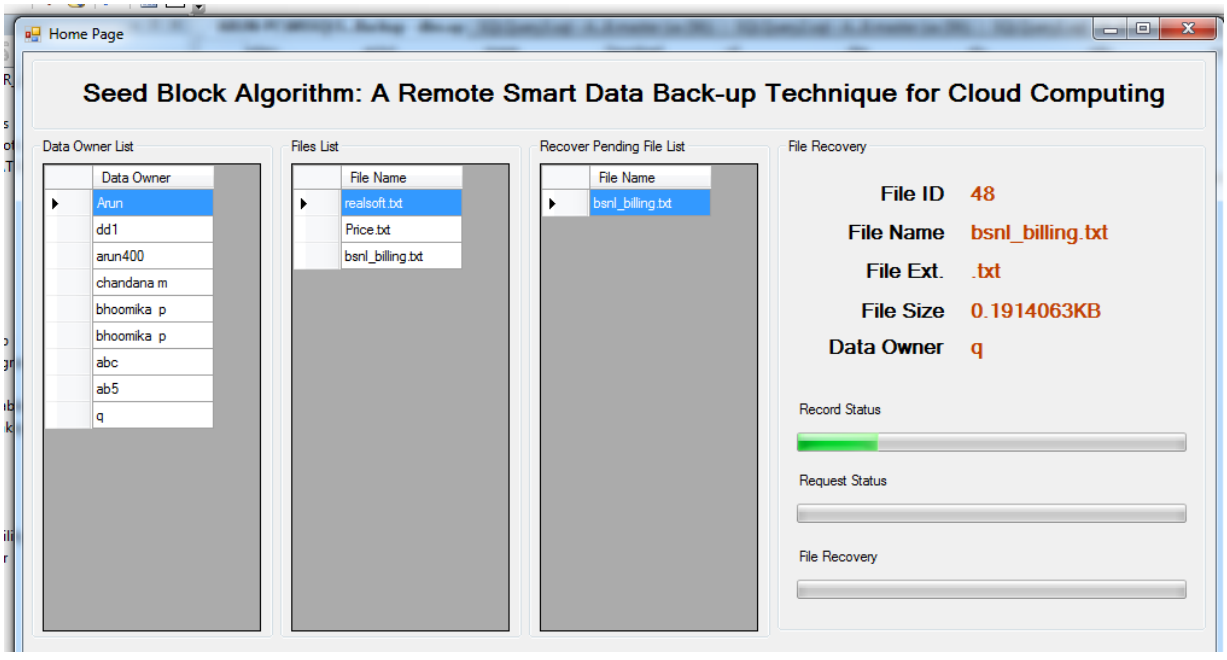
List of Data Owner



Data Owner Files



Restore Server



CONCLUSION

In this project, we presented detailed design of SBA with RSA technique. Proposed SBA focus on the security concept for the back-up files stored at remote server so that security is maintained. By doing the encryption, we are assuring the safety of the data stored on the cloud. Hence, an efficient way to store and retrieve data in a safe manner has been discussed.

In future work, We can design the prototype to support data like : Video, Sound, Other data formats. Current System can replicate data on one remote server. In future, we can extend it to replicate data on Multiple Remote Servers. Current System may fail to satisfy Mobile and Handheld device users. But we can imply better Mobile users and Handheld devices usage support. We can provide great flexibility in storage supports by using Portable Accounts concepts.

Future enhancement

The proposed system can be improved by hosting to any type of cloud .it can also be implemented with the integration of anti-phishing technology. In future the proposed system can also be concentrated on increasing the security level of the application.

REFERENCES

Books:

[1] Yoichiro Ueno, Noriharu Miyaho, Shuichi Suzuki,Muzai Gakuendai, Inzai-shi, Chiba,Kazuo Ichihara, 2010, "Performance Evaluation of a Disaster Recovery System and Practical Network System Applications," Fifth International Conference on Systems and Networks Communications,pp 256-259.

[2] Chi-won Song, Sungmin Park, Dong-wook Kim, Sooyong Kang, 2011, "Parity Cloud Service: A Privacy-Protected Personal Data Recovery Service," International Joint Conference of IEEE TrustCom-11/IEEE ICESS-11/FCST-11.

[3] Y.Ueno, N.Miyaho, and S.Suzuki, , 2009, "Disaster Recovery Mechanism using Widely Distributed Networking and Secure Metadata Handling Technology", Proceedings of the 4th edition of the UPGRADE-CN workshop, pp. 45-48.

[4] Giuseppe Pirr´o, Paolo Trunfio , Domenico Talia, Paolo Missier and Carole Goble, 2010, "ERGOT: A Semantic-based System for Service Discovery in Distributed Infrastructures," 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing.

Websites:

[1] http://en.wikipedia.org

[2] www.mywipro.com

[3] www.w3.org

[4] www.velociq.wipro.com