

Privacy Protection in Interactive Content Based Image Retrieval

Submitted in partial fulfillment of the requirements for the award of

Bachelor of Degree in B.Sc. Computer Science

By

Gokul Anandh.R (Reg. No. 38290021)

Immanuel . J (Reg. No. 38290025)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

**SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU**

March 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Gokul Anandh**
RegNo:38290021&Immanuel.J RegNo:38290025 who carried out the project
entitled “**Privacy Protection in Interactive Content Based Image Retrieval**”
under our supervision from December 2020 to March 2021.

Internal Guide

Dr. Judji Nagaraj an M.E,Ph.D

Head of the Department

Dr. Lakshmanan L M.E,Ph.D

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

I am personally indebted to a number of persons that a complete acknowledgement would be encyclopedic. First of all, I love to record my deepest gratitude to the Almighty Lord and my family.

My sincere thanks and performed sense of gratitude goes to the respected chairman for all his effort in educating me in a premier institution.

I take this opportunity to thank the Director of this prestigious institution, for his kind cooperation in completing this project.

I like to express my gratitude to our principal, and the Head of the Department, of Computer Science and Engineering, Mrs., , for their guidance and advise all through the project.

I convey my sincere and in depth gratitude to my internal guide for her valuable guidance throughout the duration of this project.

I would also like to thank our friends for the support they extended during the course of this project.

ABSTRACT

Storage requirements for visual data have been increasing in recent years, following the emergence of many highly interactive multimedia services and applications for mobile devices in both personal and corporate scenarios. This has been a key driving factor for the adoption of cloud-based data outsourcing solutions. However, outsourcing data storage to the Cloud also leads to new security challenges that must be carefully addressed, especially regarding privacy. In this paper we propose a secure framework for outsourced privacy-preserving storage and retrieval in large shared image repositories. Our proposal is based on IES-CBIR, a novel Image Encryption Scheme that exhibits Content-Based Image Retrieval properties. The framework enables both encrypted storage and searching using Content-Based Image Retrieval queries while preserving privacy against honest-but-curious cloud administrators. We have built a prototype of the proposed framework, formally analyzed and proven its security properties, and experimentally evaluated its performance and retrieval precision. Our results show that IES-CBIR is provably secure, allows more efficient operations than existing proposals, both in terms of time and space complexity, and paves the way for new practical application scenarios.

CHAPTER	TITLE	PAGE NO.
	LIST OF FIGURES	ii
	LIST OF ABBREVIATIONS	iii
1	INTRODUCTION	
	1.1 About the Project	
2	SYSTEM ANALYSIS	
	2.1 Existing system	
	2.2 Proposed system	
3	REQUIREMENTS SPECIFICATION	
	3.1 Introduction	
	3.2 Hardware and Software specification	
	3.2.1 Hardware Requirements	
	3.2.2 Software Requirements	
	3.3 Technologies Used	
	3.3.1 Introduction to Java	
	3.3.2 Working of Java	
4	SYSTEM DESIGN	
	4.1 Architecture Diagram	
	4.2 Sequence Diagram	

4.3 Use Case Diagram

4.4 Activity Diagram

4.5 Data Flow Diagram

5 SYSTEM DESIGN – DETAILED

5.1 Modules

5.2 Module explanation

6 CODING AND TESTING

6.1 Coding

6.2 Coding standards

6.3 Test procedure

6.4 Test data and output

SNAP SHOTS

REFERENCES

LIST OF FIGURES

4	System Design
5.2	Patterns of the peer-peer edges
5.2	Patterns of the service-provider edges
5.2	Discovering missing links in internet

LIST OF ABBREVIATIONS

JDK Java Development Toolkit.

JMF Java Media Framework.

TCP Transmission Control Protocol.

IP Internet Protocol.

HTTP Hyper Text Transfer Protocol

CHAPTER 1

INTRODUCTION

Aim:

To propose a secure framework for outsourced privacy-preserving storage and retrieval in large shared image repositories.

Abstract:

Cloud computing is a new form of internet-based computing that provides shared computer processing resources and data to computers and other devices on demand. It is the delivery of hosted services over the internet. Cloud computing services can be public, private or hybrid. The growing industry of cloud has provide a service paradigm of storage/computation outsourcing helps to reduce users' burden of IT infrastructure maintenance, and reduce the cost for both the enterprises and individual users. The three main benefits of cloud computing are self-service provisioning, elasticity, pay per use. The three broad categories of cloud computing are Infrastructure as a Service, Platform as a Service and Software as a Service.

CHAPTER 2

2 SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

Storage requirements for visual data have been increasing in recent years, following the emergence of many highly interactive multimedia services and applications for mobile devices in both personal and corporate scenarios. Existing proposals in this domain remain largely unpractical, namely those requiring fully homomorphism encryption, which is still computationally too expensive. Since mobile clients usually have limited computational and storage resources, they tend to rely on cloud services for storing and processing bulky data such as images. In this scenario, mobile clients (users) want to delegate their private image repositories storage to a cloud provider, while coping with the limitations of their device's storage capability, computational power, and battery life.

Problem Statement:

- ✓ In general, Encryption techniques in image processing lead to change in the size of an encrypted image. So, retrieval cannot be achieved properly.
- ✓ User's privacy is affected due to the carelessness of cloud service provider.

Images are leaked due to the lowest security level in the cloud.

2.2 PROPOSED SYSTEM

Our proposal is based on IES-CBIR, a novel Image Encryption Scheme that exhibits Content-Based Image Retrieval properties.

The framework enables both encrypted storage and searching using Content-Based Image Retrieval queries. Images are outsourced to repositories that reside in the cloud. Each repository is used by multiples Users, where they can both add their own images and/or search using a query image. Each repository is created by a single user. Upon the creation of a repository, a new repository key is generated by that user and then shared with other trusted users, allowing them to search in the repository and add/update images. In this work, we use the Bag-Of-Visual-Words (BOVW) representation to build a vocabulary tree and an inverted list index for each repository. We choose this approach for indexing as it shows good search performance and scalability properties. In the BOVW model, feature-vectors are hierarchically clustered into a vocabulary tree (also known as codebook), where each node denotes a representative feature-vector in the collection and leaf nodes are selected as the most representative nodes (called visual words).

CHAPTER 3

REQUIREMENT SPECIFICATIONS

3.1 INTRODUCTION

The requirements specification is a technical specification of requirements for the software products. It is the first step in the requirements analysis process it lists the requirements of a particular software system including functional, performance and security requirements. The requirements also provide usage scenarios from a user, an operational and an administrative perspective. The purpose of software requirements specification is to provide a detailed overview of the software project, its parameters and goals. This describes the project target audience and its user interface, hardware and software requirements. It defines how the client, team and audience see the project and its functionality.

3.2 HARDWARE AND SOFTWARE SPECIFICATION

3.2.1 HARDWARE REQUIREMENTS

- Hard Disk : 500GB and Above
- RAM : 4 GB and Above
- Processor : I3 and Above

3.2.2 SOFTWARE REQUIREMENTS

- Windows 7 and above
- JDK 1.7
- Tomcat 6.0
- MySQL5.0

3.3 TECHNOLOGIES USED

a) JAVA

It is a Platform Independent. Java is an object-oriented programming language developed initially by James Gosling and colleagues at Sun Microsystems. The language, initially called Oak (named after the oak trees outside Gosling's office), was intended to replace C++, although the feature set better resembles that of Objective C.

3.3.1 INTRODUCTION TO JAVA

Java has been around since 1991, developed by a small team of Sun Microsystems developers in a project originally called the Green project. The intent of the project was to develop a platform-independent software technology that would be used in the consumer electronics industry. The language that the team created was originally called Oak.

The first implementation of Oak was in a PDA-type device called Star Seven (*7) that consisted of the Oak language, an operating system called GreenOS, a user interface, and hardware. The name *7 was derived from the telephone sequence that was used in the team's office and that was dialed in order to answer any ringing telephone from any other phone in the office.

Around the time the First Person project was floundering in consumer electronics, a new craze was gaining momentum in America; the craze was called "Web surfing." The World Wide Web, a name applied to the Internet's millions of linked HTML documents was suddenly becoming popular for use by the masses. The reason for this was the introduction of a graphical Web browser called Mosaic,

developed by ncSA. The browser simplified Web browsing by combining text and graphics into a single interface to eliminate the need for users to learn many confusing UNIX and DOS commands. Navigating around the Web was much easier using Mosaic.

It has only been since 1994 that Oak technology has been applied to the Web. In 1994, two Sun developers created the first version of Hot Java, and then called Web Runner, which is a graphical browser for the Web that exists today. The browser was coded entirely in the Oak language, by this time called Java. Soon after, the Java compiler was rewritten in the Java language from its original C code, thus proving that Java could be used effectively as an application language. Sun introduced Java in May 1995 at the Sun World 95 convention.

Web surfing has become an enormously popular practice among millions of computer users. Until Java, however, the content of information on the Internet has been a bland series of HTML documents. Web users are hungry for applications that are interactive, that users can execute no matter what hardware or software platform they are using, and that travel across heterogeneous networks and do not spread viruses to their computers. Java can create such applications.

a) WORKING OF JAVA

For those who are new to object-oriented programming, the concept of a class will be new to you. Simplistically, a class is the definition for a segment of code that can contain both data (called attributes) and functions (called methods).

When the interpreter executes a class, it looks for a particular method by the name of main, which will sound familiar to C programmers. The main method is passed as a parameter an array of strings (similar to the argv[] of C), and is declared as a static method.

To output text from the program, we execute the println method of System.out, which is java's output stream. UNIX users will appreciate the theory behind such a stream, as it is actually standard output. For those who are instead used to the Wintel platform, it will write the string passed to it to the user's program.

Java consists of two things :

- Programming language
- Platform

b) Cloud Computing

- Cloud computing is a style of computing in which dynamically scalable and often virtualized resources are provided as a service over the Internet.
- Users need not have knowledge of, expertise in, or control over the technology infrastructure in the "cloud" that supports them.
- The concept generally incorporates combinations of the following: • infrastructure as a service (IaaS) • platform as a service (PaaS) • software as a service (SaaS)

- Cloud computing customers do not generally own the physical infrastructure serving as host to the software platform in question. Instead, they avoid capital expenditure by renting usage from a third-party provider.
- They consume resources as a service and pay only for resources that they use.
- Many cloud-computing offerings employ the utility computing model, which is analogous to how traditional utility services (such as electricity) are consumed.

c) THE JAVA PROGRAMMING LANGUAGE

Java is a high-level programming language that is all of the following:

- Simple
- Object-oriented
- Distributed
- Interpreted
- Robust
- Secure
- Architecture-neutral
- Portable
- High-performance
- Multithreaded
- Dynamic

The code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

Java is unusual in that each Java program is both compiled and interpreted. With a compiler, you translate a Java program into an intermediate language called **Java byte codes** – the platform independent codes interpreted by the Java interpreter. With an interpreter, each Java byte code instruction is parsed and run on the computer. Compilation happens just once;

interpretation occurs each time the program is executed. This figure illustrates how it works :

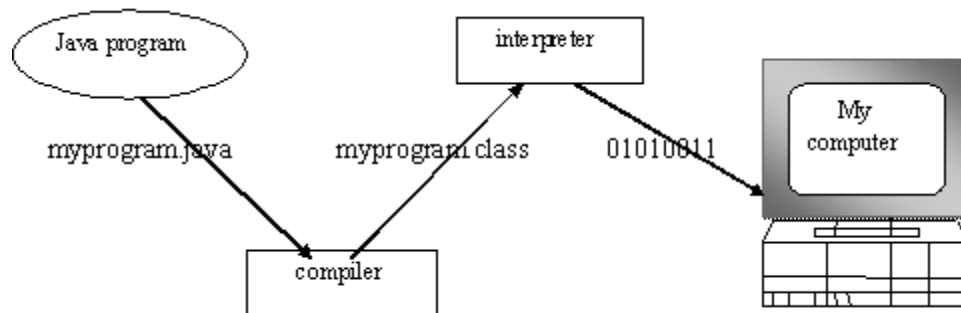
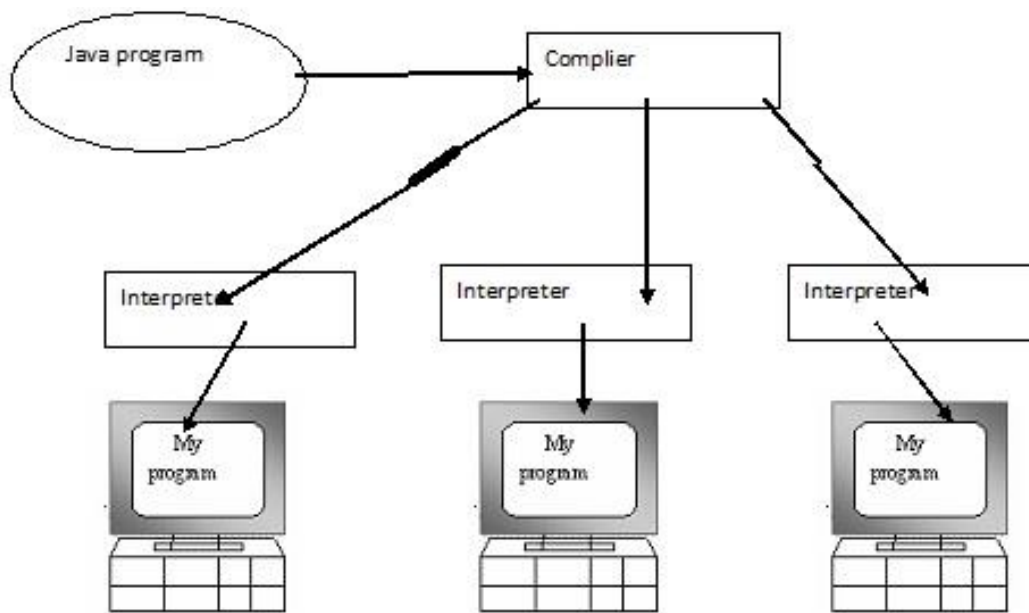


Fig.3.1

You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (JVM). Every Java interpreter, whether it's a Java development tool or a Web browser that can run Java applets, is an implementation of JVM. That JVM can also be implemented in hardware. Java byte codes help make “write once, run anywhere” possible.

You can compile your Java program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the JVM. For example, that same Java program can be run on Windows NT, Solaris and Macintosh



Windows NT
System 8

d) THE JAVA PLATFORM

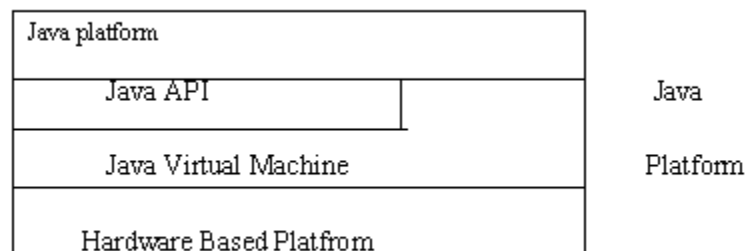
A platform is the hardware or software environment in which a program runs. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other, hardware-based platforms. Most other platforms are described as a combination of hardware and operating system.

The Java platform has two components :

- The Java Virtual Machine (JVM)
- The Java Application Programming Interface (Java API)

You've already been introduced to the JVM. It's the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries (**packages**) of related components. The following figure depicts a Java program, such as an application or applet, that's running on the Java platform. As the figure shows, the Java API and Virtual Machine insulates the Java program from hardware dependencies.



As a platform-independent environment, Java can be a bit slower than native code. However, smart compilers, weel-tuned interpreters, and just-in-time byte compilers can bring Java's performance close to that of native code without threatening protability.

3.3.2 WORKING OF JAVA

For those who are new to object-oriented programming, the concept of a class will be new to you. Simplistically, a class is the definition for a segment of code that can contain both data and functions.

When the interpreter executes a class, it looks for a particular method by the name of main, which will sound familiar to C programmers. The main method is passed as a parameter an array of strings (similar to the argv[] of C), and is declared as a static method.

To output text from the program, we execute the println method of System.out, which is java's output stream. UNIX users will appreciate the theory behind such a stream, as it is actually standard output. For those who are instead used to the Wintel platform, it will write the string passed to it to the user's program.

APACHE TOMCAT SERVER

Apache Tomcat (formerly under the Apache Jakarta Project; Tomcat is now a top level project) is a web container developed at the Apache Software Foundation. Tomcat implements the servlet and the JavaServer Pages ([JSP](#)) specifications from Sun Microsystems, providing an environment for Java code to run in cooperation with a web server. It adds tools for configuration and management but can also be configured by editing configuration files that are normally [XML](#)-formatted. Because Tomcat includes its own HTTP server internally, it is also considered a standalone web server.

Environment

Tomcat is a web server that supports servlets and JSPs. Tomcat comes with the Jasper compiler that compiles JSPs into servlets.

The Tomcat servlet engine is often used in combination with an Apache web server or other web servers. Tomcat can also function as an independent web server. Earlier in its development, the perception existed that standalone Tomcat was only suitable for development environments and other environments with minimal requirements for speed and transaction handling. However, that perception no longer exists; Tomcat is increasingly used as a standalone web server in high-traffic, high-availability environments.

Since its developers wrote Tomcat in Java, it runs on any operating system that has a JVM.

Product features

Tomcat 3.x (initial release)

- Implements the Servlet 2.2 and JSP 1.1 specifications
- Servlet reloading
- Basic HTTP functionality Tomcat 4.x
- Implements the Servlet 2.3 and JSP 1.2 specifications
- Servlet container redesigned as Catalina

- JSP engine redesigned as Jasper
- Coyote connector
- Java Management Extensions (JMX), JSP & Struts-based administration Tomcat 5.x
- Implements the Servlet 2.4 and JSP 2.0 specifications
- Reduced garbage collection, improved performance and scalability
- Native Windows and Unix wrappers for platform integration
- Faster JSP parsing

History

Tomcat started off as a servlet specification implementation by James Duncan Davidson, a software architect at Sun. He later helped make the project open source and played a key role in its donation by Sun to the Apache Software Foundation.

Davidson had initially hoped that the project would become open-sourced and, since most open-source projects had O'Reilly books associated with them featuring an animal on the cover, he wanted to name the project after an animal. He came up with Tomcat since he reasoned the animal represented something that could take care of and fend for itself. His wish to see an animal cover eventually came true when O'Reilly published their Tomcat book with a tomcat on the cover.

Introduction

Purpose

To propose a secure framework for outsourced privacy-preserving storage and retrieval in large shared image repositories.

Project Scope

Storage requirements for visual data have been increasing in recent years, following the emergence of many highly interactive multimedia services and applications for mobile devices in both personal and corporate scenarios.

Existing proposals in this domain remain largely unpractical, namely those requiring fully homomorphism encryption, which is still computationally too expensive. Since mobile clients usually have limited computational and storage resources, they tend to rely on cloud services

for storing and processing bulky data such as images. In this scenario, mobile clients (users) want to delegate their private image repositories storage to a cloud provider, while coping with the limitations of their device's storage capability, computational power, and battery life.

Overall Description

Product Perspective

Cloud computing is a new form of internet-based computing that provides shared computer processing resources and data to computers and other devices on demand. It is the delivery of hosted services over the internet. Cloud computing services can be public, private or hybrid. The growing industry of cloud has provide a service paradigm of storage/computation outsourcing helps to reduce users' burden of IT infrastructure maintenance, and reduce the cost for both the enterprises and individual users.

The three main benefits of cloud computing are self-service provisioning, elasticity, pay per use. The three broad categories of cloud computing are Infrastructure as a Service, Platform as a Service and Software as a Service.

Product Features

Our proposal is based on IES-CBIR, a novel Image Encryption Scheme that exhibits Content-Based Image Retrieval properties. The framework enables both encrypted storage and searching using Content-Based Image Retrieval queries. Images are outsourced to repositories that reside in the cloud. Each repository is used by multiples Users, where they can both add their own images and/or search using a query image. Each repository is created by a single user. Upon the creation of a repository, a new repository key is generated by that user and then shared with other trusted users, allowing them to search in the repository and add/update images. In this work, we use the Bag-Of-Visual-Words (BOVW) representation to build a vocabulary tree and an inverted list index for each repository. We choose this approach for indexing as it shows good search performance and scalability properties. In the BOVW model, feature-vectors are hierarchically clustered into a vocabulary tree (also known as codebook), where each node denotes a representative feature-vector in the collection and leaf nodes are selected as the most representative nodes (called visual words).

. User Classes and Characteristics

Create Repository & Upload Images:

Repository is storage space of collection of data. Each repository is created by single user. He is the owner of that repository. Then, he generates a key for that repository by using RSA algorithm and shared with the users who are all have an account to access it

Now, Repository can be accessed by multiple users with the permission of an owner. Then, owner upload huge amount of image datasets as zip file into the cloud.

Codebook & Index generation:

The admin of cloud has responsibility to create documents based on images which is useful for searching of images by users. So, he extracts zip file and applying CBIR Encryption technique. It encrypts images based on color values and texture features and also shuffling the pixels in column-wise as well as row-wise. Then, he creates codebook, index and image key for those encrypted images. These files are used to improve the searching efficiency of cloud and also manage the time properly while retrieving answer.

Add Image/Query to cloud:

Now, Users can access the cloud to add their own images into the repository. So, if that cloud has 'n' number of users, then repository has chance to increase rapidly.

Now, the repository has collection of 'n' number of images in different domains. All the images are stored in encrypted format for security. Then, user has to ask query to cloud. Its take query is in the format of encrypted image using CBIR encryption technique.

Content Based Searching & Retrieval:

After receiving encrypted image query, the cloud extracts the features of an original image. Now applying content based searching on the codebook and image

index by using that extracted features. Obviously, now searching results will be an encrypted image. This resulted answer will send to that corresponding user. Now, user can apply CBIR decryption technique to decrypt the retrieved images. So, the answer will be very fine and delicious due to huge dataset.

a) Design and Implementation Constraints

Constraints in Analysis

- Constraints as Informal Text
- Constraints as Operational Restrictions
- Constraints Integrated in Existing Model Concepts
- Constraints as a Separate Concept
- Constraints Implied by the Model Structure

Constraints in Design

- Determination of the Involved Classes
- Determination of the Involved Objects
- Determination of the Involved Actions
- Determination of the Require Clauses
- Global actions and Constraint Realization

Constraints in Implementation

A hierarchical structuring of relations may result in more classes and a more complicated structure to implement. Therefore it is advisable to transform the hierarchical relation structure to a simpler structure such as a classical flat one. It is rather straightforward to transform the developed hierarchical model into a bipartite, flat model, consisting of classes

on the one hand and flat relations on the other. Flat relations are preferred at the design level for reasons of simplicity and implementation ease. There is no identity or functionality associated with a flat relation. A flat relation corresponds with the relation concept of entity-relationship modeling and many object oriented methods.

System Features

Our proposal is based on IES-CBIR, a novel Image Encryption Scheme that exhibits Content-Based Image Retrieval properties. The framework enables both encrypted storage and searching using Content-Based Image Retrieval queries. Images are outsourced to repositories that reside in the cloud. Each repository is used by multiples Users, where they can both add their own images and/or search using a query image. Each repository is created by a single user. Upon the creation of a repository, a new repository key is generated by that user and then shared with other trusted users, allowing them to search in the repository and add/update images. In this work, we use the Bag-Of-Visual-Words (BOVW) representation to build a vocabulary tree and an inverted list index for each repository. We choose this approach for indexing as it shows good search performance and scalability properties. In the BOVW model, feature-vectors are hierarchically clustered into a vocabulary tree (also known as codebook), where each node denotes a representative feature-vector in the collection and leaf nodes are selected as the most representative nodes (called visual words).

External Interface Requirements

a)User Interfaces

1. All the contents in the project are implemented using Graphical User Interface (GUI) in Java through JavaFX concepts.
2. Every conceptual part of the projects is reflected using the JavaFX.
3. System gets the input and delivers through the GUI based.

b)Hardware Interfaces

Ethernet

Ethernet on the AS/400 supports TCP/IP, Advanced Peer-to-Peer Networking (APPN) and advanced program-to-program communications (APPC).

ISDN

You can connect your AS/400 to an Integrated Services Digital Network (ISDN) for faster, more accurate data transmission. An ISDN is a public or private digital communications network that can support data, fax, image, and other services over the same physical interface. Also, you can use other protocols on ISDN, such as IDLC and X.25.

c) Software Interfaces

This software is interacted with the TCP/IP protocol, Socket and listening on unused ports. Server Socket and listening on unused ports and JDK 1.6

d) Communications Interfaces

1. TCP/IP protocol.
2. LAN settings

1. Other Nonfunctional Requirements

Performance Requirements

The performance of the wireless sensor network, to execute this project on LAN or wifi communication channel. So we need to one or more than machine to execute the demo. Machine needs the enough hard disk space to install the software and run our project.

1.2 Safety Requirements

- The software may be safety-critical. If so, there are issues associated with its integrity level
- The software may not be safety-critical although it forms part of a safety-critical system. For example, software may simply log transactions.
- If a system must be of a high integrity level and if the software is shown to be of that integrity level, then the hardware must be at least of the same integrity level.
- There is little point in producing 'perfect' code in some language if hardware and system software (in widest sense) are not reliable.

- If a computer system is to run software of a high integrity level then that system should not at the same time accommodate software of a lower integrity level.
- Systems with different requirements for safety levels must be separated.
- Otherwise, the highest level of integrity required must be applied to all systems in the same environment.

1.3 Security Requirements

Do not block the some available ports through the windows firewall

Software Quality Attributes

Functionality: are the required functions available, including Interoperability and security.

Reliability: maturity, fault tolerance and recoverability

Usability: how easy it is to understand, learn, and operate the software System

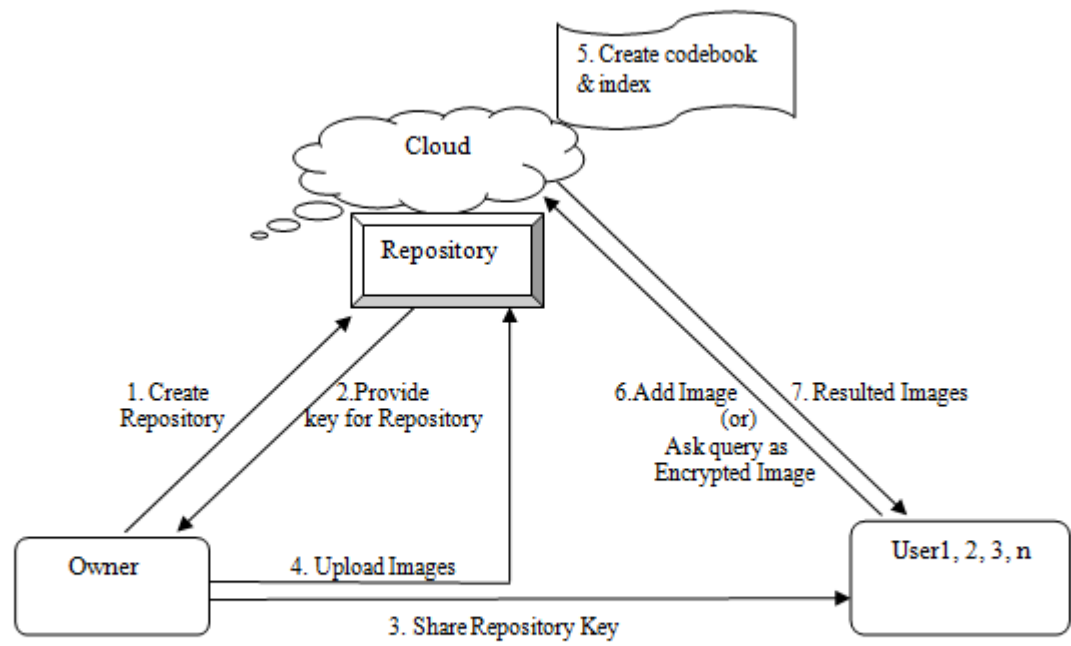
Efficiency: performance and resource behavior.

Maintainability: Maintaining the software.

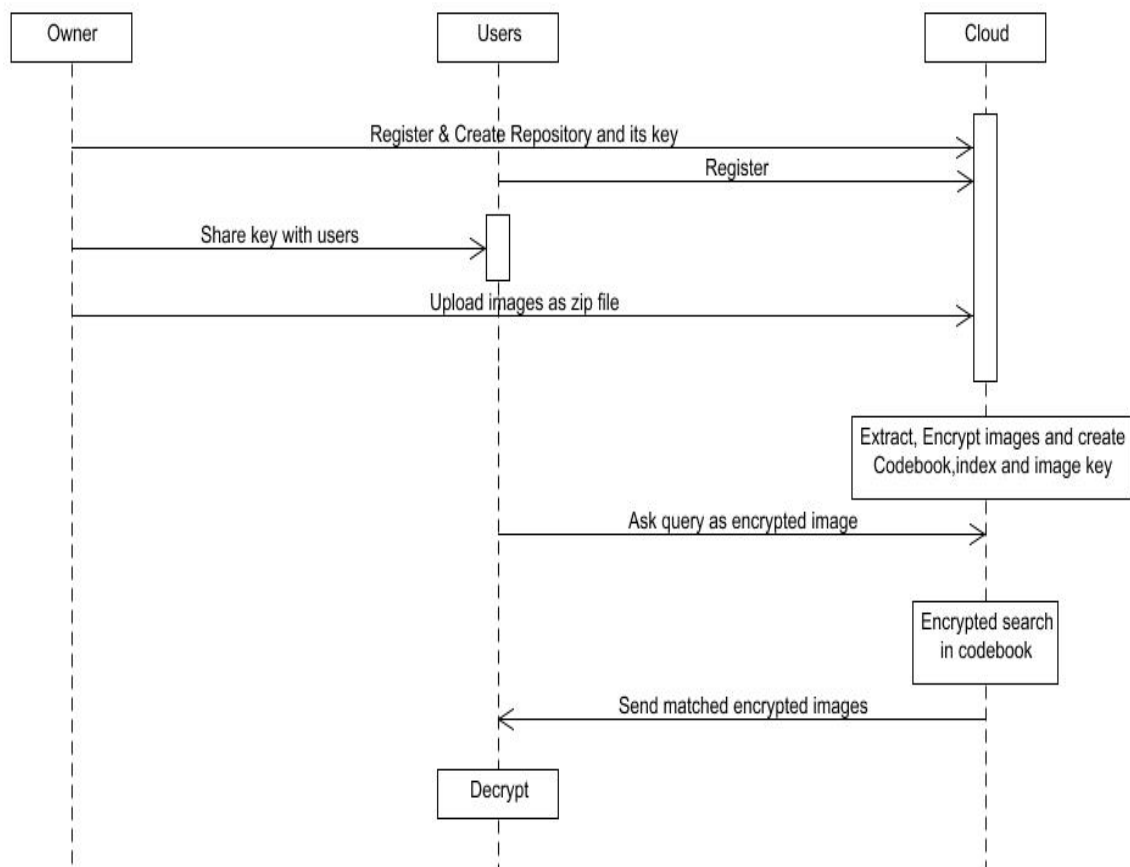
Portability: can the software easily be transferred to another environment, Including installability

CHAPTER 4

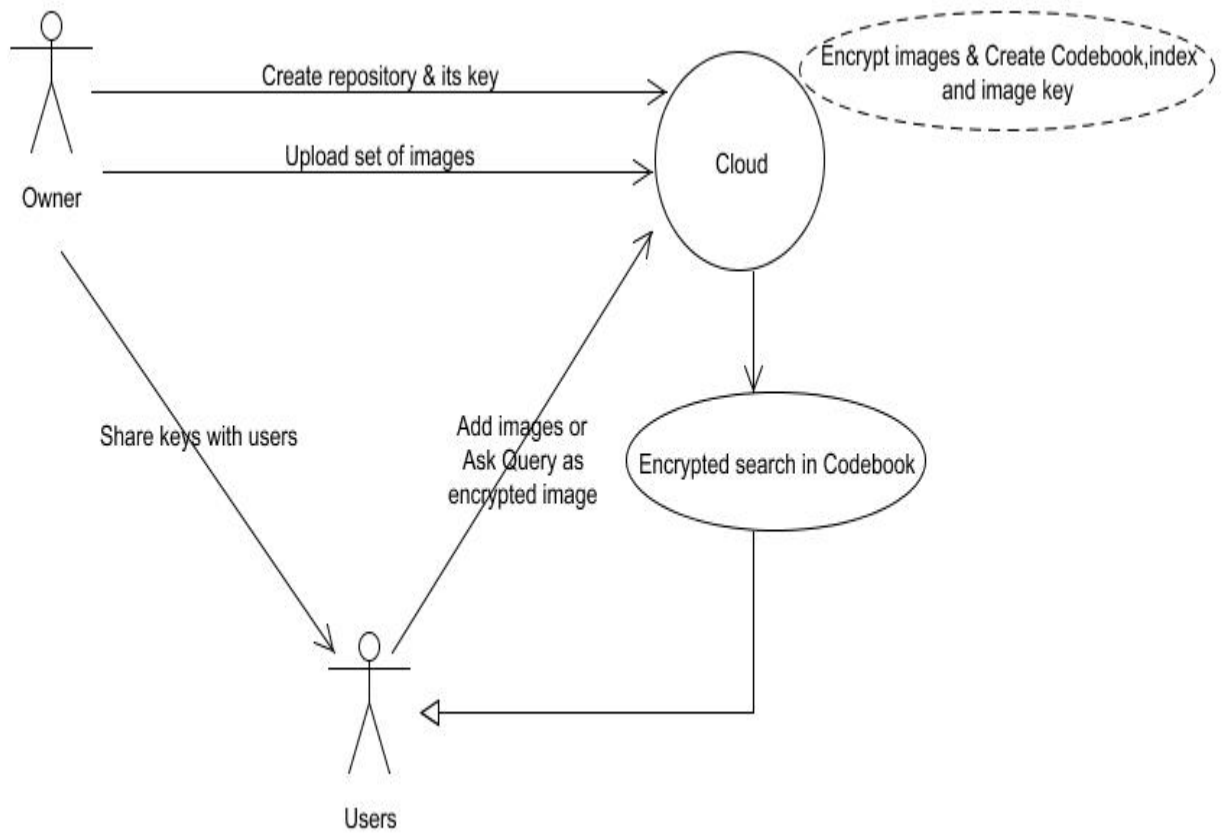
Architecture Diagram:



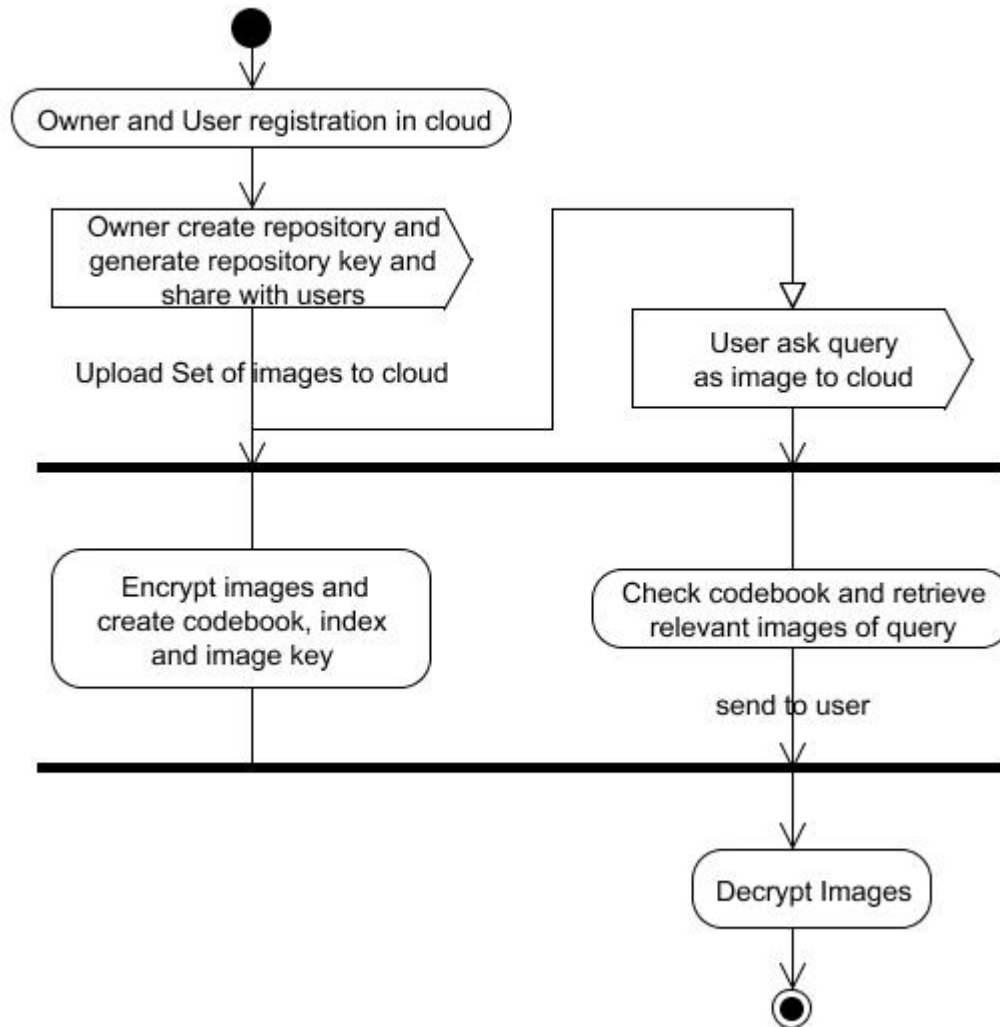
4.1 Sequence Diagram:



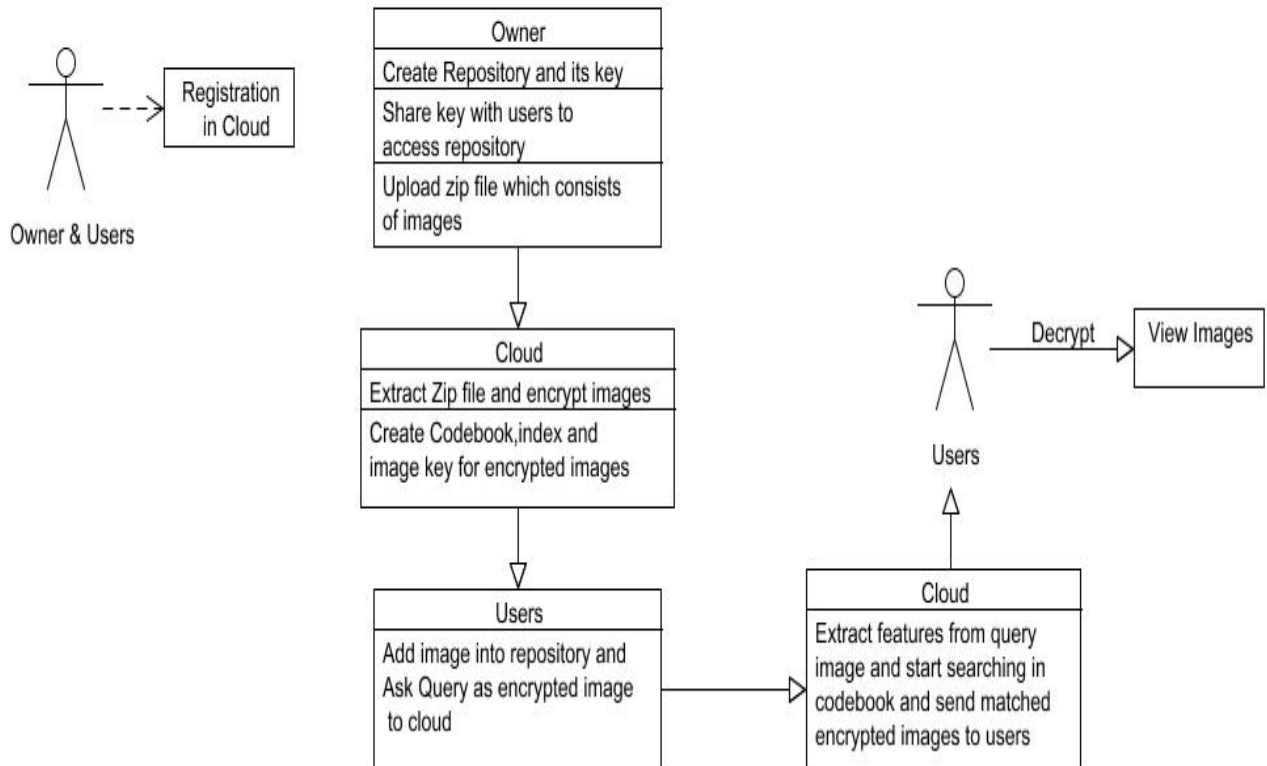
4.2 Use Case Diagram:



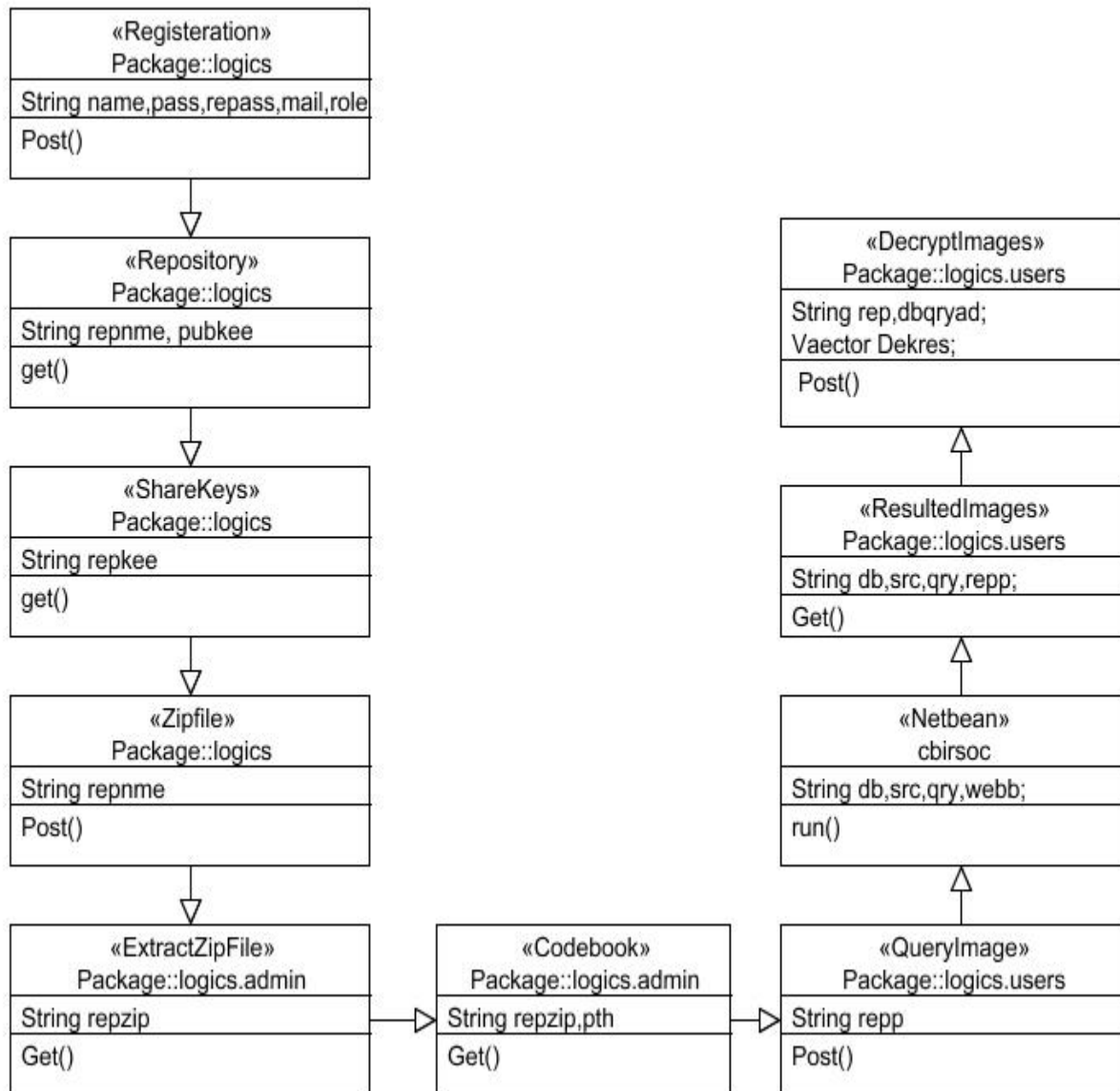
4.3 Activity Diagram:



Collaboration Diagram:



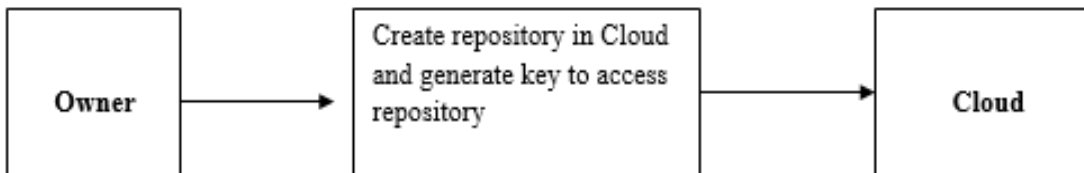
Class Diagram:



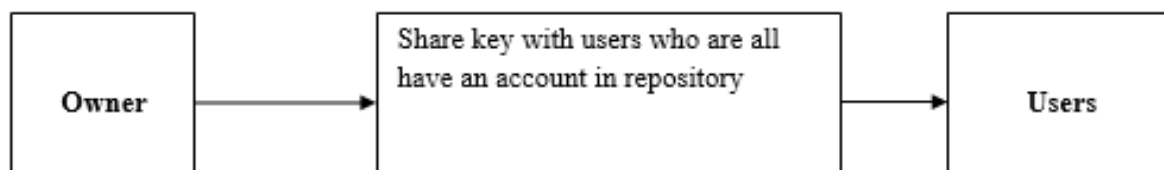
Data Flow Diagram:

Level

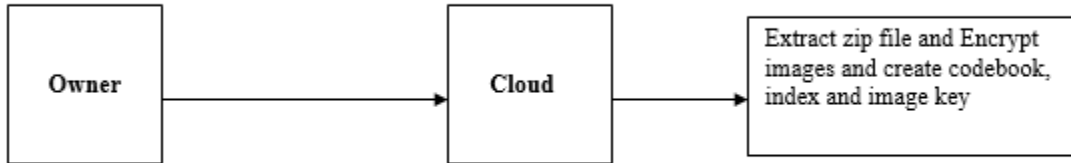
1



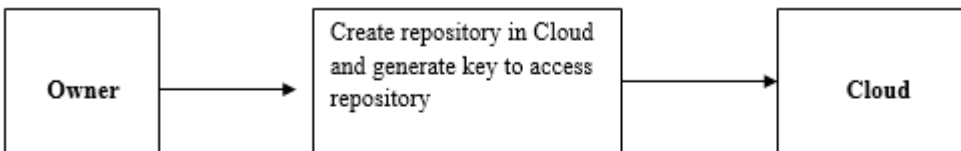
Level 2 Upload set of images



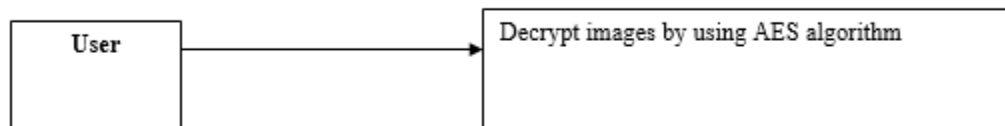
Level 3



Level 4



Level 5



CHAPTER 5

SYSTEM DESIGN

5.1 MODULES

Modules:

- **Create Repository & Upload Images**
- **Codebook & Index generation**
- **Add Image/Query to cloud**
- **Content Based Searching & Retrieval**

Create Repository & Upload Images:

Repository is storage space of collection of data. Each repository is created by single user. He is the owner of that repository. Then, he generates a key for that repository by using RSA algorithm and shared with the users who are all have an account to access it. Now, Repository can be accessed by multiple users with the permission of an owner. Then, owner upload huge amount of image datasets as zip file into the cloud.

Codebook & Index generation:

The admin of cloud has responsibility to create documents based on images which is useful for searching of images by users. So, he extracts zip file and applying CBIR Encryption technique. It encrypts images based on color values and

texture features and also shuffling the pixels in column-wise as well as row-wise. Then, he creates codebook, index and image key for those encrypted images. These files are used to improve the searching efficiency of cloud and also manage the time properly while retrieving answer.

Add Image/Query to cloud:

Now, Users can access the cloud to add their own images into the repository. So, if that cloud has 'n' number of users, then repository has chance to increase rapidly. Now, the repository has collection of 'n' number of images in different domains. All the images are stored in encrypted format for security. Then, user has to ask query to cloud. Its take query is in the format of encrypted image using CBIR encryption technique.

Content Based Searching & Retrieval:

After receiving encrypted image query, the cloud extracts the features of an original image. Now applying content based searching on the codebook and image index by using that extracted features. Obviously, now searching results will be an encrypted image. This resulted answer will send to that corresponding user. Now, user can apply CBIR decryption technique to decrypt the retrieved images. So, the answer will be very fine and delicious due to huge dataset.

CHAPTER 6

6.1 CODING STANDARDS

Coding standards are guidelines to programming that focuses on the physical structure and appearance of the program. They make the code easier to read, understand and maintain. This phase of the system actually implements the blueprint developed during the design phase. The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

- Program should be simple, clear and easy to understand.
- Naming conventions
- Value conventions
- Script and comment procedure
- Message box format
- Exception and error handling

6.1.1 NAMING CONVENTIONS

Naming conventions of classes, data member, member functions, procedures etc., should be **self-descriptive**. One should even get the meaning and scope of the variable by its name. The conventions are adopted for **easy understanding** of the intended message by the user. So it is customary to follow the conventions. These conventions are as follows:

Class names

Class names are problem domain equivalence and begin with capital letter and have mixed cases.

Member Function and Data Member name

Member function and data member name begins with a lowercase letter with each subsequent letters of the new words in uppercase and the rest of letters in lowercase.

6.1.2 VALUE CONVENTIONS

Value conventions ensure values for variable at any point of time. This involves the following:

- Proper default values for the variables.
- Proper validation of values in the field.
- Proper documentation of flag values.

6.1.3 SCRIPT WRITING AND COMMENTING STANDARD

Script writing is an art in which indentation is utmost important. Conditional and looping statements are to be properly aligned to facilitate easy understanding. Comments are included to minimize the number of surprises that could occur when going through the code.

6.1.4 MESSAGE BOX FORMAT

When something has to be prompted to the user, he must be able to understand it properly. To achieve this, a specific format has been adopted in displaying messages to the user. They are as follows:

- X – User has performed illegal operation.
- ! – Information to the user.

6.2 TEST PROCEDURE

SYSTEM TESTING

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself.

For example the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough.

Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

- Static analysis is used to investigate the structural properties of the Source code.
- Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

6.3 TEST DATA AND OUTPUT

6.3.1 UNIT TESTING

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing

6.3.2 FUNCTIONAL TEST

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and special values, such as logically related inputs, files of identical elements, and empty files

Three types of tests in Functional test:

- Performance Test
- Stress Test
- Structure Test

6.3.3 PERFORMANCE TEST

It determines the amount of execution time spent in various parts of the unit, program throughput, and response time and device utilization by the program unit.

6.3.4 STRESS TEST

Stress Test is those test designed to intentionally break the unit. A Great deal can be learned about the strength and limitations of a program by examining the manner in which a programmer in which a program unit breaks.

6.3.5 STRUCTURED TEST

Structure Tests are concerned with exercising the internal logic of a program and traversing particular execution paths. The way in which White-Box test strategy was employed to ensure that the test cases could Guarantee that all independent paths within a module have been have been exercised at least once.

- Exercise all logical decisions on their true or false sides.
- Execute all loops at their boundaries and within their operational bounds.

- Exercise internal data structures to assure their validity.
- Checking attributes for their correctness.
- Handling end of file condition, I/O errors, buffer problems and textual errors in output information
-

6.3.6 INTEGRATION TESTING

Integration testing is a systematic technique for construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that, interface dispenses can be easily found and corrected.

The major error that was faced during the project is linking error. When all the modules are combined the link is not set properly with all support files. Then we checked out for interconnection and the links. Errors are localized to the new module and its intercommunications.

The product development can be staged, and modules integrated in as they complete unit testing. Testing is completed when the last module is integrated and tested.

6.3.7 TESTING TECHNIQUES / TESTING STRATEGIES

a) TESTING

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet – undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate the review of specification design and coding.

Testing is the process of executing the program with the intent of finding the error. A good test case design is one that has a probability of finding an yet undiscovered error. A successful test is one that uncovers an yet undiscovered error. Any engineering product can be tested in one of the two ways:

b) WHITE BOX TESTING

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing:

- Flow graph notation
- Cyclometric complexity
- Deriving test cases
- Graph matrices Control

c) BLACK BOX TESTING

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”,

that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

- Graph based testing methods
- Equivalence partitioning
- Boundary value analysis
- Comparison testing

d) SOFTWARE TESTING STRATEGIES:

A software testing strategy provides a road map for the software developer. Testing is a set activity that can be planned in advance and conducted systematically. For this reason a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:

- Testing begins at the module level and works “outward” toward the integration of the entire computer based system.
- Different testing techniques are appropriate at different points in time.
- The developer of the software and an independent test group conducts testing.
- Testing and Debugging are different activities but debugging must be accommodated in any testing strategy.

e) INTEGRATION TESTING:

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with. Individual modules, which are highly prone to interface errors, should not be assumed to work instantly when we put them together. The problem of course, is “putting them together”- interfacing. There may be the chances of data lost across on another’s sub functions, when combined may not produce the desired major function; individually acceptable impression may be magnified to unacceptable levels; global data structures can present problems.

f) PROGRAM TESTING:

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that in violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax error. These errors are shown through error messages generated by the computer. A logic error on the other hand deals with the incorrect data fields, out-off-range items and invalid combinations. Since the compiler s will not deduct logical error, the programmer must examine the output. Condition testing exercises the logical conditions contained in a module. The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses A relational operator or on arithmetic expression.

g) SECURITY TESTING

Security testing attempts to verify the protection mechanisms built in to a system well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attack must also be tested for invulnerability from rear attack. During security, the tester places the role of individual who desires to penetrate system.

h) VALIDATION TESTING

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted, one of two conditions exists.

- The function or performance characteristics confirm to specifications and are accepted.
- A validation from specification is uncovered and a deficiency created.

Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily. Though there were deficiencies in the system they were not catastrophic.

i) USER ACCEPTANCE TESTING

User acceptance of the system is key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system and user at the time of developing and making changes whenever required. This is done in regarding to the following points.

- Input screen design.
- Output screen design.

REFERENCES

- [1] M. Meeker, “Internet Trends 2015,” in Code Conf., 2015.
- [2] Global Web Index, “Instagram tops the list of social network growth,” <http://tinyurl.com/hnwwlzm>, 2013.
- [3] C. D. Manning, P. Raghavan, and H. Schütze, An Introduction to Information Retrieval. Cambridge University Press, 2009, vol. 1.
- [4] R. Chow, P. Golle, M. Jakobsson, E. Shi, J. Staddon, R. Masuoka, and J. Molina, “Controlling data in the cloud: outsourcing computation without outsourcing control,” in CCSW’09, 2009.
- [5] D. Rushe, “Google: don’t expect privacy when sending to Gmail,” <http://tinyurl.com/kjga34x>, 2013.
- [6] G. Greenwald and E. MacAskill, “NSA Prism program taps in to user data of Apple, Google and others,” <http://tinyurl.com/oea3g8t>, 2013.
- [7] A. Chen, “GCreep: Google Engineer Stalked Teens, Spied on Chats,” <http://gawker.com/5637234>, 2010.
- [8] J. Halderman and S. Schoen, “Lest we remember: cold-boot attacks on encryption keys,” in Commun. ACM, vol. 52, no. 5, 2009.

[9] National Vulnerability Database, “CVE Statistics,”
<http://web.nvd.nist.gov/view/vuln/statistics>, 2014.

[10] D. Lewis, “iCloud Data Breach: Hacking And Celebrity Photos,”
<https://tinyurl.com/nohznmr>, 2014.