**CHAPTER-1**

**INTRODUCTION**

**What Is Image Processing?**

Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. It is a type of signal dispensation in which input is image, like video frame or photograph and output may be image or characteristics associated with that image. Usually Image Processing system includes treating images as two-dimensional signals while applying already set signal processing methods to them.

It is among rapidly growing technologies today, with its applications in various aspects of a business. Image Processing forms core research area within engineering and computer science disciplines too.

**Image processing basically includes the following three steps:**

* Importing the image with optical scanner or by digital photography.
* Analyzing and manipulating the image which includes data compression and image enhancement and spotting patterns that are not to human eyes like satellite photographs.
* Output is the last stage in which result can be altered image or report that is based on image analysis.

**Purpose of Image processing:**

The purpose of image processing is divided into 5 groups. They are:

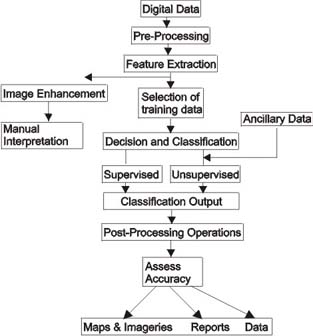
1. Visualization - Observe the objects that are not visible.
2. Image sharpening and restoration - To create a better image
3. Image retrieval - Seek for the image of interest.
4. Measurement of pattern – Measures various objects in an image.
5. Image Recognition – Distinguish the objects in an image.

**Types of Image Processing:**

The two types of methods used for Image Processing are Analog and Digital Image Processing. Analog or visual techniques of image processing can be used for the hard copies like printouts and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques. The image processing is not just confined to area that has to be studied but on knowledge of analyst. Association is another important tool in image processing through visual techniques. So, analysts apply a combination of personal knowledge and collateral data to image processing.

Digital Processing techniques help in manipulation of the digital images by using computers. As raw data from imaging sensors from satellite platform contains deficiencies. To get over such flaws and to get originality of information, it has to undergo various phases of processing. The three general phases that all types of data have to undergo while using digital technique are Pre- processing, enhancement and display, information extraction.

**Working diagram of Image Processing:**

****

**Characteristics of Image Processing:**

Before going to processing an image, it is converted into a digital form. Digitization includes sampling of image and quantization of sampled values. After converting the image into bit information, processing is performed. This processing technique may be, Image enhancement, Image restoration, and Image compression.

**Image enhancement:**

It refers to accentuation, or sharpening, of image features such as boundaries, or contrast to make a graphic display more useful for display & analysis. This process does not increase the inherent information content in data. It includes gray level & contrast manipulation, noise reduction, edge crispening and sharpening, filtering, interpolation and magnification, pseudo coloring, and so on.

**Image restoration:**

It is concerned with filtering the observed image to minimize the effect of degradations. Effectiveness of image restoration depends on the extent and accuracy of the knowledge of degradation process as well as on filter design. Image restoration differs from image enhancement in that the latter is concerned with more extraction or accentuation of image features.

**Image compression:**

It is concerned with minimizing the number of bits required to represent an image. Application of compression are in broadcast TV, remote sensing via satellite, military communication via aircraft, radar, teleconferencing, facsimile transmission, for educational & business documents, medical images that arise in computer tomography, magnetic resonance imaging and digital radiology, motion, pictures, satellite images, weather maps, geological surveys and so on.

* Text compression – CCITT GROUP3 & GROUP4
* Still image compression – JPEG
* Video image compression - MPEG

**Advantages of Image Processing:**

* The processing of images is faster and more cost-effective. One needs less time for processing, as well as less film and other photographing equipment.
* It is more ecological to process images. No processing or fixing chemicals are needed to take and process digital images. However, printing inks are essential when printing digital images.
* When shooting a digital image, one can immediately see if the image is good or not.
* Copying a digital image is easy, and the quality of the image stays good unless it is compressed. For instance, saving an image as jpg format compresses the image. By resaving the image as jpg format, the compressed image will be recompressed, and the quality of the image will get worse with every saving.
* Fixing and retouching of images has become easier. In new Photoshop 7, it is possible to smoother face wrinkles with a new Healing Brush Tool in a couple of seconds.
* The expensive reproduction (compared with rastering the image with a repro camera) is faster and cheaper.
* By changing the image format and resolution, the image can be used in a number of media.

**1.2 About The People:**

As a team we have the clear vision and realize it too. As a statistical evaluation, the team has more than 15,000 hours of expertise in providing real-time solutions in the fields of Android Mobile Apps Development, Networking, Web Designing, Secure Computing, Mobile Computing, Cloud Computing, Image Processing And Implementation, Networking With OMNET++ Simulator, client Server Technologies in Java,(J2EE\J2ME\EJB), ANDROID, DOTNET (ASP.NET, VB.NET, C#.NET), MATLAB, NS2, SIMULINK, EMBEDDED, POWER ELECTRONICS, VB & VC++, Oracle and operating system concepts with LINUX.

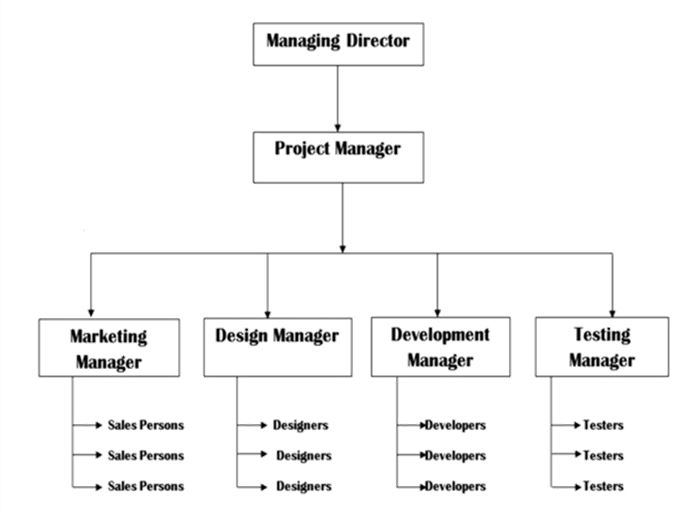
**Our Vision:**

**“WE CAN DEVELOP YOUR OWN IDEAS”** this is our vision; we work according to our vision.

**Customer Focus**

We view our customer relationships as partnerships and are committed to being accountable to ensure that the development, integration and implementation of solutions are performed in a professional and timely manner. We unite this accountability with our dedication to applying the most appropriate methodologies & technologies, and we dispatch our 'best-of-breed' technology professionals to make it all happen. The result is satisfied customers who consistently give us high marks for our expansive offerings of precise engineering.

**Organization Chart:**



**CHAPTER-2**

**SYSTEM STUDY AND ANALYSIS**

**EXISTING SYSTEM:**

* Rasiwasia *et al.* presents that modeling the correlations between modalities is more effective in feature spaces with higher levels of abstraction. The scalability of incorporating new semantic concepts into the semantic space is studied, which allows the updated embedding function to be applied to dynamic image repositories.
* Craswell et al. propose a Markov random walk model to a large click log for finding relevant documents, including those that as-yet unclicked for a query, without analyzing the query content or image content.

**DISADVANTAGES OF EXISTING SYSTEM:**

* These methods are based on the techniques of learning to rank and take the ranking examples as the pairwise (or listwise) input to optimize a certain ranking loss. PAMIR is the first attempt to address the problem of ranking images by text queries.
* The problem of ranking images by text queries. PAMIR formulates the cross-modal retrieval problem in a way similar to that of RankSVM and derives an efficient training procedure by adapting the Passive-Aggressive algorithm.
* The drawback of the existing model is that it cannot be applied to the new emerging queries or images.

**PROPOSED SYSTEM:**

* By optimizing both the truncated random walk loss as well as the distance between the social representation and the internal representation of the vertices, the social representations of the vertices and the parameters of the deep neural networks are learned. Thus the proposed model not only captures more accurate semantics of the training queries and images, but also generalizes to the unseen queries and images better.
* The proposed MRW-NN model can be integrated with the deep structure of these methods, while it remains an open question in our further work whether there is a specific (and better) deep structure for modeling the text queries and the images in the click graph.
* The main goal of the proposed model is to perform cross-modal ranking, which differs from that of DeepWalk that aims to learning the latent representation for classifying the static members of a social network.
* The proposed MRW-NN, CMRNN takes two modality-specific neural networks for mapping the queries and the images into a common subspace, while it optimizes the listwise ranking loss of the cross-modal ranking examples.

**ADVANTAGES OF PROPOSED SYSTEM:**

* The proposed model not only captures more accurate semantics of the training queries and images, but also generalizes to the unseen queries and images better. When used for cross-modal retrieval,
* Noticing that the click graph can be viewed as not only text-query-image ranking examples but also image-query-text ranking examples, Bi-CMSRM proposed in takes bi-directional ranking examples into account, such that two directions of retrieval are optimized simultaneously, yielding a better representation for multimodal data.
* First suggests the use of random walks to learn latent representation on the social community graph. In this work, we constrain ourselves to learn the latent representation of the multimodal click graph.
* Click Aware: The more the explicit clicks between a query and an image, the closer their latent representation should be. The mapped representation should also encode the implicit connections between the vertices in the click graph.
* Generalization Aptitude: It is insufficient to learn latent representation for the present members of the click graph only and the proposed modal should be able to perform cross-modal ranking in the future.

**2.3 FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**2.3.1 ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### **2.3.2 TECHNICAL FEASIBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**2.3.2 SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**2.4 FUNCTIONAL REQUIREMENTS**

**Admin**

It is used by the admin for activating and deactivating user request.

**User**

It is used to add document into database or used to download document from database.

**Modular Manager**

It is used to accept the document that is to be upload

**2.5 NON-FUNCTIONAL REQUIREMENTS:**

Non-Functional Requirements (quality attributes) ensure the delivery of an operable and manageable system which provides the required functionality reliable, uninterrupted or with minimal time of interruption even under unusual situations.

* **Security**

Login requirements - access levels, CRUD levels

Password requirements - length, special characters, expiry, recycling policies Inactivity timeouts – durations, actions

* **Audit**

Audited elements – what business elements will be audited?

Audited fields – which data fields will be audited?

Audit file characteristics - before image, after image, user and time stamp, etc

* **Performance**

Response times - application loading, screen open and refresh times, etc Processing times – functions, calculations, imports, exports.

Query and Reporting times – initial loads and subsequent loads Capacity.

* **Availability**

Hours of operation – when is it available? Consider weekends, holidays, maintenance times, etc

Locations of operation – where should it be available from, what are the connection requirements?

* **Reliability**

Mean Time Between Failures – What is the acceptable threshold for down-time? e .g. One a year, 4,000 hours

Mean Time To Recovery – if broken, how much time is available to get the system back up again?

* **Integrity**

Fault trapping (I/O) – how to handle electronic interface failures, etc

Bad data trapping - data imports, flag-and-continue or stop the import policies, etc

Data integrity – referential integrity in database tables and interfaces.Image compression and decompression standards

* **Recovery**

Recovery process – how do recoveries work, what is the process?

Recovery time scales – how quickly should a recovery take to perform?

Backup frequencies – how often is the transaction data, set-up data, and system (code) backed-up?

* **Compatibility**

Compatibility on different operating systems – What does it have to be able to run on?

Compatibility on different platforms – What are the hardware platforms it needs to work on?

* **Maintainability**

Conformance to architecture standards – What are the standards it needs to conform to or have exclusions from?

Conformance to design standards – What design standards must be adhered to or exclusions created?

Conformance to coding standards – What coding standards must be adhered to or exclusions created?

* **Usability**

Look and feel standards - screen element density, layout and flow, colours, UI metaphors, keyboard shortcuts

Internationalization / localization requirements – languages, spellings, keyboards, paper sizes, etc

**CHAPTER-3**

**DEVELOPMENT ENVIRONMENT**

**3.1 SOFTWARE REQUIREMENTS:**

* Programming Language : JAVA/J2EE
* Java Version : JDK 1.6 & above
* Data Base : MY SQL
* Operating System : Windows OS
  1. **HARDWARE REQUIREMENTS:**
* Processor : Intel-i3
* Ram : 1GB(min)
* Hard Disk : 256 GB

## ABOUT FRONT END

## 3.3 Java Technology

Java technology is both a programming language and a platform.

### The Java Programming Language

### The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

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

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called Java byte codes —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.



You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it’s a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make “write once, run anywhere” possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.



### The Java Platform

A platform is the hardware or software environment in which a program runs. We’ve already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and Mac OS. Most platforms can be described as a combination of the operating system and hardware. 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.

The Java platform has two components:

* The Java Virtual Machine (Java VM)
* The Java Application Programming Interface (Java API)

You’ve already been introduced to the Java VM. 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 of related classes and interfaces; these libraries are known as packages. The next section, what Can Java Technology Do? Highlights what functionality some of the packages in the Java API provide.

The following figure depicts a program that’s running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.



Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time byte code compilers can bring performance close to that of native code without threatening portability.

## What Can Java Technology Do?

The most common types of programs written in the Java programming language are applets and applications. If you’ve surfed the Web, you’re probably already familiar with applets. An applet is a program that adheres to certain conventions that allow it to run within a Java-enabled browser.

However, the Java programming language is not just for writing cute, entertaining applets for the Web. The general-purpose, high-level Java programming language is also a powerful software platform. Using the generous API, you can write many types of programs.

An application is a standalone program that runs directly on the Java platform. A special kind of application known as a server serves and supports clients on a network. Examples of servers are Web servers, proxy servers, mail servers, and print servers. Another specialized program is a servlet. A servlet can almost be thought of as an applet that runs on the server side. Java Servlets are a popular choice for building interactive web applications, replacing the use of CGI scripts. Servlets are similar to applets in that they are runtime extensions of applications. Instead of working in browsers, though, servlets run within Java Web servers, configuring or tailoring the server.

How does the API support all these kinds of programs? It does so with packages of software components that provides a wide range of functionality. Every full implementation of the Java platform gives you the following features:

* **The essentials**: Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.
* **Applets**: The set of conventions used by applets.
* **Networking**: URLs, TCP (Transmission Control Protocol), UDP (User Data gram Protocol) sockets, and IP (Internet Protocol) addresses.
* **Internationalization**: Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.
* **Security**: Both low level and high level, including electronic signatures, public and private key management, access control, and certificates.
* **Software components**: Known as Java Beans TM, can plug into existing component architectures.
* **Object serialization**: Allows lightweight persistence and communication via Remote Method Invocation (RMI).
* **Java Database Connectivity (JDBCTM)**: Provides uniform access to a wide range of relational databases.

The Java platform also has APIs for 2D and 3D graphics, accessibility, servers, collaboration, telephony, speech, animation, and more. The following figure depicts what is included in the Java 2 SDK.



## How Will Java Technology Change My Life?

We can’t promise you fame, fortune, or even a job if you learn the Java programming language. Still, it is likely to make your programs better and requires less effort than other languages. We believe that Java technology will help you do the following:

* **Get started quickly**: Although the Java programming language is a powerful object-oriented language, it’s easy to learn, especially for programmers already familiar with C or C++.
* **Write less code**: Comparisons of program metrics (class counts, method counts, and so on) suggest that a program written in the Java programming language can be four times smaller than the same program in C++.
* **Write better code**: The Java programming language encourages good coding practices, and its garbage collection helps you avoid memory leaks. Its object orientation, its JavaBeans component architecture, and its wide-ranging, easily extendible API let you reuse other people’s tested code and introduce fewer bugs.
* **Develop programs more quickly**: Your development time may be as much as twice as fast versus writing the same program in C++. Why? You write fewer lines of code and it is a simpler programming language than C++.
* **Avoid platform dependencies with 100% Pure Java**: You can keep your program portable by avoiding the use of libraries written in other languages. The 100% Pure JavaTM Product Certification Program has a repository of historical process manuals, white papers, brochures, and similar materials online.
* **Write once, run anywhere**: Because 100% Pure Java programs are compiled into machine-independent byte codes, they run consistently on any Java platform.
* **Distribute software more easily**: You can upgrade applets easily from a central server. Applets take advantage of the feature of allowing new classes to be loaded “on the fly,” without recompiling the entire program.

### ODBC

Microsoft Open Database Connectivity (ODBC) is a standard programming interface for application developers and database systems providers. Before ODBC became a de facto standard for Windows programs to interface with database systems, programmers had to use proprietary languages for each database they wanted to connect to. Now, ODBC has made the choice of the database system almost irrelevant from a coding perspective, which is as it should be. Application developers have much more important things to worry about than the syntax that is needed to port their program from one database to another when business needs suddenly change.

Through the ODBC Administrator in Control Panel, you can specify the particular database that is associated with a data source that an ODBC application program is written to use. Think of an ODBC data source as a door with a name on it. Each door will lead you to a particular database. For example, the data source named Sales Figures might be a SQL Server database, whereas the Accounts Payable data source could refer to an Access database. The physical database referred to by a data source can reside anywhere on the LAN.

The ODBC system files are not installed on your system by Windows 95. Rather, they are installed when your setup a separate database application, such as SQL Server Client or Visual Basic 4.0. When the ODBC icon is installed in Control Panel, it uses a file called ODBCINST.DLL. It is also possible to administer your ODBC data sources through a stand-alone program called ODBCADM.EXE. There is a 16-bit and a 32-bit version of this program and each maintains a separate list of ODBC data sources.

From a programming perspective, the beauty of ODBC is that the application can be written to use the same set of function calls to interface with any data source, regardless of the database vendor. The source code of the application doesn’t change whether it talks to Oracle or SQL Server. We only mention these two as an example. There are ODBC drivers available for several dozen popular database systems. Even Excel spreadsheets and plain text files can be turned into data sources. The operating system uses the Registry information written by ODBC Administrator to determine which low-level ODBC drivers are needed to talk to the data source (such as the interface to Oracle or SQL Server). The loading of the ODBC drivers is transparent to the ODBC application program. In a client/server environment, the ODBC API even handles many of the network issues for the application programmer.

The advantages of this scheme are so numerous that you are probably thinking there must be some catch. The only disadvantage of ODBC is that it isn’t as efficient as talking directly to the native database interface. ODBC has had many detractors make the charge that it is too slow. Microsoft has always claimed that the critical factor in performance is the quality of the driver software that is used. In our humble opinion, this is true. The availability of good ODBC drivers has improved a great deal recently. And anyway, the criticism about performance is somewhat analogous to those who said that compilers would never match the speed of pure assembly language. Maybe not, but the compiler (or ODBC) gives you the opportunity to write cleaner programs, which means you finish sooner. Meanwhile, computers get faster every year.

**JDBC**

In an effort to set an independent database standard API for Java; Sun Microsystems developed Java Database Connectivity, or JDBC. JDBC offers a generic SQL database access mechanism that provides a consistent interface to a variety of RDBMSs. This consistent interface is achieved through the use of “plug-in” database connectivity modules, or drivers. If a database vendor wishes to have JDBC support, he or she must provide the driver for each platform that the database and Java run on.

To gain a wider acceptance of JDBC, Sun based JDBC’s framework on ODBC. As you discovered earlier in this chapter, ODBC has widespread support on a variety of platforms. Basing JDBC on ODBC will allow vendors to bring JDBC drivers to market much faster than developing a completely new connectivity solution.

JDBC was announced in March of 1996. It was released for a 90 day public review that ended June 8, 1996. Because of user input, the final JDBC v1.0 specification was released soon after.

The remainder of this section will cover enough information about JDBC for you to know what it is about and how to use it effectively. This is by no means a complete overview of JDBC. That would fill an entire book.

### JDBC Goals

Few software packages are designed without goals in mind. JDBC is one that, because of its many goals, drove the development of the API. These goals, in conjunction with early reviewer feedback, have finalized the JDBC class library into a solid framework for building database applications in Java.

The goals that were set for JDBC are important. They will give you some insight as to why certain classes and functionalities behave the way they do. The eight design goals for JDBC are as follows:

1. **SQL Level API**

The designers felt that their main goal was to define a SQL interface for Java. Although not the lowest database interface level possible, it is at a low enough level for higher-level tools and APIs to be created. Conversely, it is at a high enough level for application programmers to use it confidently. Attaining this goal allows for future tool vendors to “generate” JDBC code and to hide many of JDBC’s complexities from the end user.

1. **SQL Conformance**

SQL syntax varies as you move from database vendor to database vendor. In an effort to support a wide variety of vendors, JDBC will allow any query statement to be passed through it to the underlying database driver. This allows the connectivity module to handle non-standard functionality in a manner that is suitable for its users.

1. **JDBC must be implemental on top of common database interfaces** The JDBC SQL API must “sit” on top of other common SQL level APIs. This goal allows JDBC to use existing ODBC level drivers by the use of a software interface. This interface would translate JDBC calls to ODBC and vice versa.
2. **Provide a Java interface that is consistent with the rest of the Java system**

Because of Java’s acceptance in the user community thus far, the designers feel that they should not stray from the current design of the core Java system.

1. **Keep it simple**

This goal probably appears in all software design goal listings. JDBC is no exception. Sun felt that the design of JDBC should be very simple, allowing for only one method of completing a task per mechanism. Allowing duplicate functionality only serves to confuse the users of the API.

1. **Use strong, static typing wherever possible**

Strong typing allows for more error checking to be done at compile time; also, less error appear at runtime.

1. **Keep the common cases simple**

Because more often than not, the usual SQL calls used by the programmer are simple SELECT’s, INSERT’s, DELETE’s and UPDATE’s, these queries should be simple to perform with JDBC. However, more complex SQL statements should also be possible.

Finally we decided to proceed the implementation using Java Networking.

And for dynamically updating the cache table we go for MS Access database.

Java ha two things: a programming language and a platform.

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

Simple Architecture-neutral

Object-oriented Portable

Distributed High-performance

Interpreted multithreaded

Robust Dynamic

Secure

Java is also unusual in that each Java program is both compiled and interpreted. With a compile you translate a Java program into an intermediate language called Java byte codes the platform-independent code instruction is passed and run on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The figure illustrates how this works.

**Java Program**

**Compilers**

**Interpreter**

**My Program**

You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it’s a Java development tool or a Web browser that can run Java applets, is an implementation of the Java VM. The Java VM 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 my platform that has a Java compiler. The byte codes can then be run any implementation of the Java VM. For example, the same Java program can run Windows NT, Solaris, and Macintosh

## Networking

### TCP/IP stack

The TCP/IP stack is shorter than the OSI one:



TCP is a connection-oriented protocol; UDP (User Datagram Protocol) is a connectionless protocol.

### IP datagram’s

The IP layer provides a connectionless and unreliable delivery system. It considers each datagram independently of the others. Any association between datagram must be supplied by the higher layers. The IP layer supplies a checksum that includes its own header. The header includes the source and destination addresses. The IP layer handles routing through an Internet. It is also responsible for breaking up large datagram into smaller ones for transmission and reassembling them at the other end.

### UDP

UDP is also connectionless and unreliable. What it adds to IP is a checksum for the contents of the datagram and port numbers. These are used to give a client/server model - see later.

### TCP

TCP supplies logic to give a reliable connection-oriented protocol above IP. It provides a virtual circuit that two processes can use to communicate.

### Internet addresses

In order to use a service, you must be able to find it. The Internet uses an address scheme for machines so that they can be located. The address is a 32 bit integer which gives the IP address. This encodes a network ID and more addressing. The network ID falls into various classes according to the size of the network address.

### Network address

Class A uses 8 bits for the network address with 24 bits left over for other addressing. Class B uses 16 bit network addressing. Class C uses 24 bit network addressing and class D uses all 32.

### Subnet address

Internally, the UNIX network is divided into sub networks. Building 11 is currently on one sub network and uses 10-bit addressing, allowing 1024 different hosts.

### Host address

8 bits are finally used for host addresses within our subnet. This places a limit of 256 machines that can be on the subnet.

### Total address



The 32-bit address is usually written as 4 integers separated by dots.

### Port addresses

A service exists on a host, and is identified by its port. This is a 16 bit number. To send a message to a server, you send it to the port for that service of the host that it is running on. This is not location transparency! Certain of these ports are "well known".

### Sockets

A socket is a data structure maintained by the system to handle network connections. A socket is created using the call socket. It returns an integer that is like a file descriptor. In fact, under Windows, this handle can be used with Read File and Write File functions.

#include <sys/types.h>

#include <sys/socket.h>

int socket(int family, int type, int protocol);

Here "family" will be AF\_INET for IP communications, protocol will be zero, and type will depend on whether TCP or UDP is used. Two processes wishing to communicate over a network create a socket each. These are similar to two ends of a pipe - but the actual pipe does not yet exist.

**J Free Chart**

J Free Chart is a free 100% Java chart library that makes it easy for developers to display professional quality charts in their applications. Fréchet’s extensive feature set includes:

* A consistent and well-documented API, supporting a wide range of chart types;
* A flexible design that is easy to extend, and targets both server-side and client-side applications;
* Support for many output types, including Swing components, image files (including PNG and JPEG), and vector graphics file formats (including PDF, EPS and SVG);
* J Free Chart is "open source" or, more specifically, [free software](http://www.gnu.org/philosophy/free-sw.html). It is distributed under the terms of the [GNU Lesser General Public License](http://www.gnu.org/licenses/lgpl.html) (LGPL), which permits use in proprietary applications.

## 1. Map Visualizations

Charts showing values that relate to geographical areas. Some examples include: (a) population density in each state of the United States, (b) income per capita for each country in Europe, (c) life expectancy in each country of the world. The tasks in this project include:

Sourcing freely redistributable vector outlines for the countries of the world, states/provinces in particular countries (USA in particular, but also other areas);

Creating an appropriate dataset interface (plus default implementation), a rendered, and integrating this with the existing XY Plot class in J Free Chart;

Testing, documenting, testing some more, documenting some more.

## 2. Time Series Chart Interactivity

Implement a new (to J Free Chart) feature for interactive time series charts --- to display a separate control that shows a small version of ALL the time series data, with a sliding "view" rectangle that allows you to select the subset of the time series data to display in the main chart.

## 3. Dashboards

There is currently a lot of interest in dashboard displays. Create a flexible dashboard mechanism that supports a subset of J Free Chart chart types (dials, pies, thermometers, bars, and lines/time series) that can be delivered easily via both Java Web Start and an applet.

## 4. Property Editors

The property editor mechanism in J Free Chart only handles a small subset of the properties that can be set for charts. Extend (or re implement) this mechanism to provide greater end-user control over the appearance of the charts.

**J2ME (Java 2 Micro edition): -**

Sun Microsystems defines J2ME as "a highly optimized Java run-time environment targeting a wide range of consumer products, including pagers, cellular phones, screen-phones, digital set-top boxes and car navigation systems." Announced in June 1999 at the Java One Developer Conference, J2ME brings the cross-platform functionality of the Java language to smaller devices, allowing mobile wireless devices to share applications. With J2ME, Sun has adapted the Java platform for consumer products that incorporate or are based on small computing devices.

**1. General J2ME architecture**



J2ME uses configurations and profiles to customize the Java Runtime Environment (JRE). As a complete JRE, J2ME is comprised of a configuration, which determines the JVM used, and a profile, which defines the application by adding domain-specific classes. The configuration defines the basic run-time environment as a set of core classes and a specific JVM that run on specific types of devices. We'll discuss configurations in detail. The profile defines the application; specifically, it adds domain-specific classes to the J2ME configuration to define certain uses for devices. We'll cover profiles in depth. The following graphic depicts the relationship between the different virtual machines, configurations, and profiles. It also draws a parallel with the J2SE API and its Java virtual machine. While the J2SE virtual machine is generally referred to as a JVM, the J2ME virtual machines, KVM and CVM, are subsets of JVM. Both KVM and CVM can be thought of as a kind of Java virtual machine -- it's just that they are shrunken versions of the J2SE JVM and are specific to J2ME.

**2. Developing J2ME applications**

Introduction In this section, we will go over some considerations you need to keep in mind when developing applications for smaller devices. We'll take a look at the way the compiler is invoked when using J2SE to compile J2ME applications. Finally, we'll explore packaging and deployment and the role preverification plays in this process.

**3. Design considerations for small devices**

Developing applications for small devices requires you to keep certain strategies in mind during the design phase. It is best to strategically design an application for a small device before you begin coding. Correcting the code because you failed to consider all of the "got chas” before developing the application can be a painful process. Here are some design strategies to consider:

\* Keep it simple. Remove unnecessary features, possibly making those features a separate, secondary application.

\* Smaller is better. This consideration should be a "no brainer" for all developers. Smaller applications use less memory on the device and require shorter installation times. Consider packaging your Java applications as compressed Java Archive (jar) files.

\* Minimize run-time memory use. To minimize the amount of memory used at run time, use scalar types in place of object types. Also, do not depend on the garbage collector. You should manage the memory efficiently yourself by setting object references to null when you are finished with them. Another way to reduce run-time memory is to use lazy instantiation, only allocating objects on an as-needed basis. Other ways of reducing overall and peak memory use on small devices are to release resources quickly, reuse objects, and avoid exceptions.

**4. Configurations overview**

The configuration defines the basic run-time environment as a set of core classes and a specific JVM that run on specific types of devices. Currently, two configurations exist for J2ME, though others may be defined in the future:

\* **Connected Limited Device Configuration (CLDC)**

It is used specifically with the KVM for 16-bit or 32-bit devices with limited amounts of memory. This is the configuration (and the virtual machine) used for developing small J2ME applications. Its size limitations make CLDC more interesting and challenging (from a development point of view) than CDC. CLDC is also the configuration that we will use for developing our drawing tool application. An example of a small wireless device running small applications is a Palm hand-held computer.

\* **Connected Device Configuration (CDC)**

It is used with the C virtual machine (CVM) and is used for 32-bit architectures requiring more than 2 MB of memory. An example of such a device is a Net TV box.**5. J2ME profiles**

**What is a J2ME profile?**

As we mentioned earlier in this tutorial, a profile defines the type of device supported. The Mobile Information Device Profile (MIDP), for example, defines classes for cellular phones. It adds domain-specific classes to the J2ME configuration to define uses for similar devices. Two profiles have been defined for J2ME and are built upon CLDC: K Java and MIDP. Both KJava and MIDP are associated with CLDC and smaller devices. Profiles are built on top of configurations.

**Profile 1: K Java**

K Java is Sun's proprietary profile and contains the K Java API. The K Java profile is built on top of the CLDC configuration. The K Java virtual machine, KVM, accepts the same byte codes and class file format as the classic J2SE virtual machine. K Java contains a Sun-specific API that runs on the Palm OS. The K Java API has a great deal in common with the J2SE Abstract Windowing Toolkit (AWT). However, because it is not a standard J2ME package, its main package is com. sun. kjava. We'll learn more about the K Java API later in this tutorial when we develop some sample applications.

**Profile 2: MIDP**

MIDP is geared toward mobile devices such as cellular phones and pagers. The MIDP, like KJava, is built upon CLDC and provides a standard run-time environment that allows new applications and services to be deployed dynamically on end user devices. MIDP is a common, industry-standard profile for mobile devices that is not dependent on a specific vendor. packages.

\* java.lang \* java.io \* java.util

\* javax.microedition.io \* javax.microedition.lcdui

\* javax.microedition.midlet \* javax.microedition.rms

**CHAPTER-4**

**SYSTEM DESIGN**

**4.1 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.

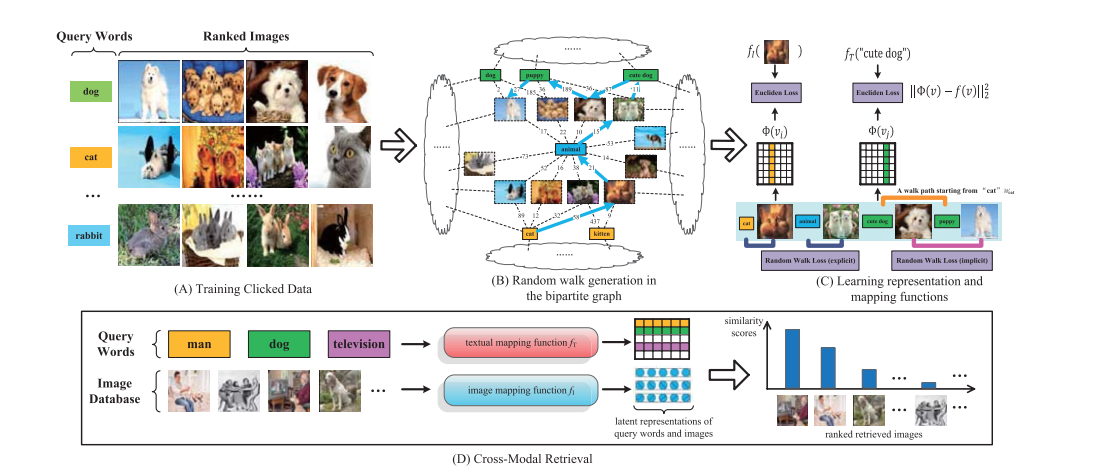
The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

**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.

**SYSTEM ARCHITECTURE:**



**BLOCK DIAGRAM:**

Upload images

admin

user detailsasdfge1

categories

DB

Clicks

Likes,dislikes

ADMIN

Graph

LOGIN

user

Search images

USER

KEYWORD

LIKES

DISLIKES

COMMENT

**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.

Update user images

Suggest tags

feedback

LOGIN

Upload images

User details

User keyword

Search images

USER

Image detail

Key word

ADMIN

Dislikes

Download images

clicks

Graph

likes

**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.

Admin

USER

**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.

USER

ADMIN

Login

User list

Image list

us

Registraction

Iogin

Upload images()

Edit user images()

User views()

User search history()

User feedback()

Graph()

Search images()

Image details()

likes ()

Dislikes()

Save images ()

Suggest tags()

**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.

Admin

Users

Database

Login

Upload images

Image details

User details

Search images

Search history

Edit images

Likes & dislikes

Image details

User views

Likes

Dislikes

Comments

Suggest tags

**ACTIVITY DIAGRAM:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

Graph analysis

Likes & dislikes

User views

Split Categories

UPLOAD IMAGES

Accept

NO

lOGIN

Commet

Dislikes

Likes

ADMIN

Image details

Login

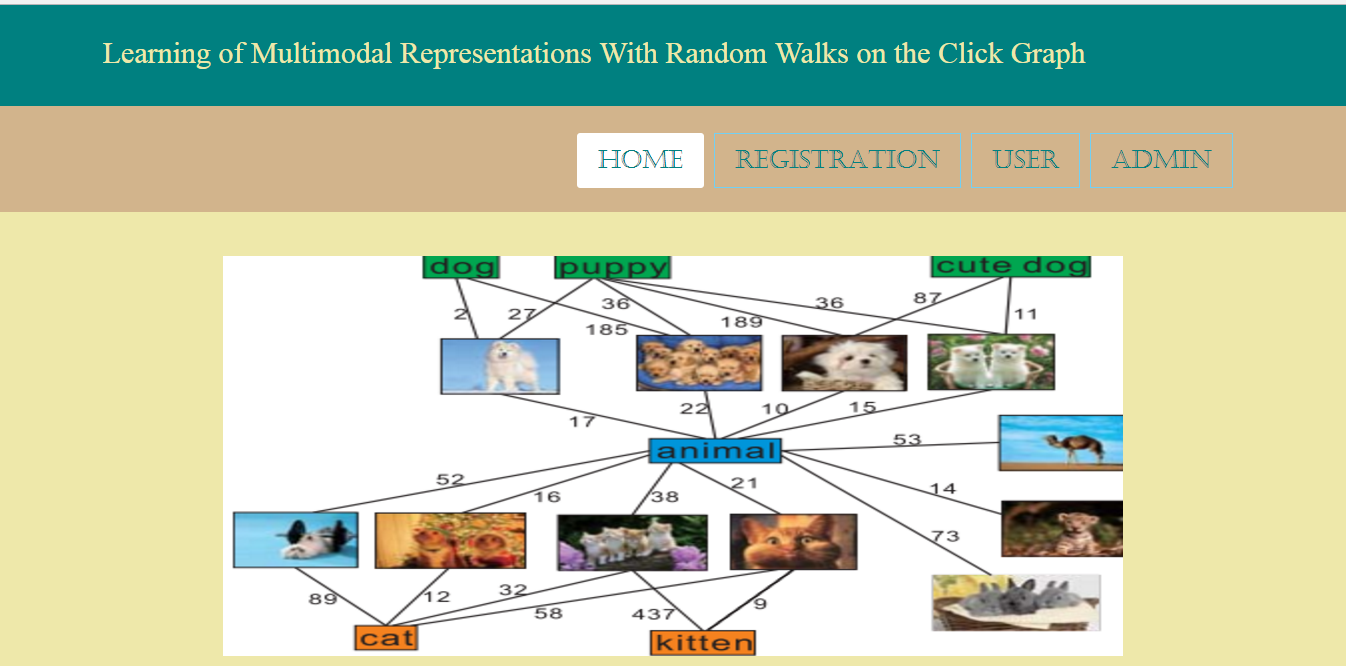
Start

USER

Key based

SEARCH IMAGES

**4.2 USER INTERFACE DIAGRAM:**





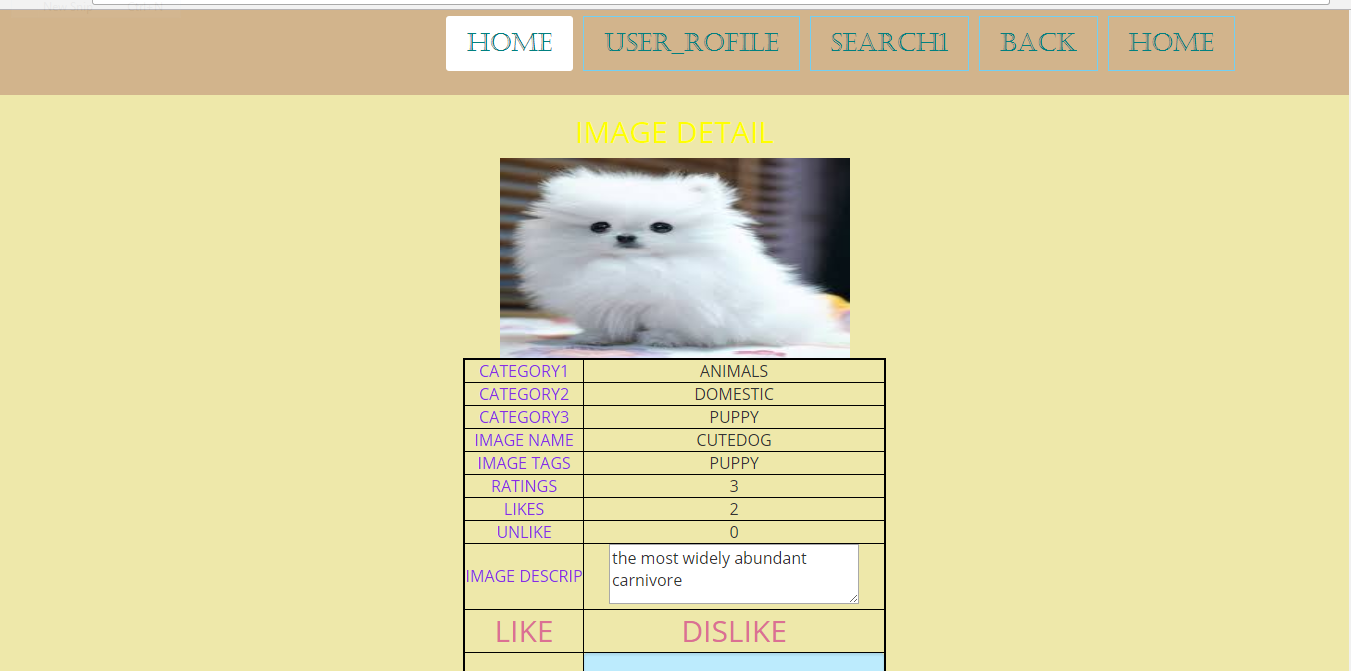






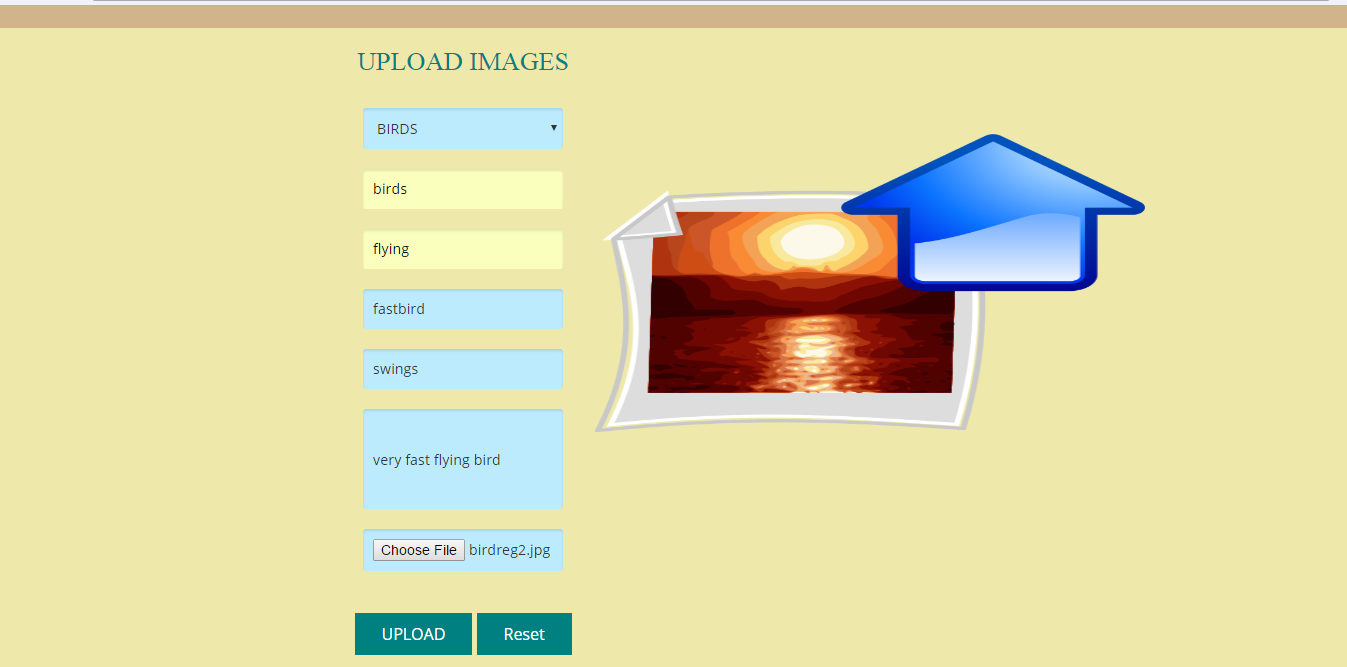


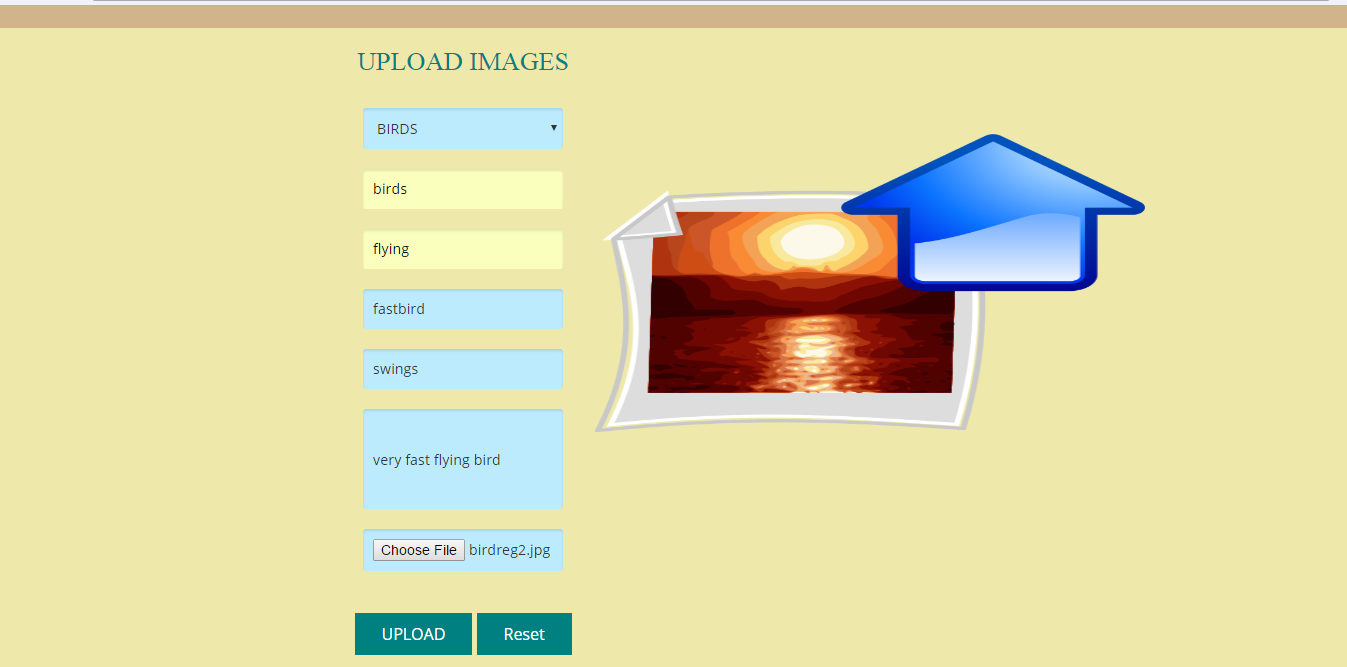




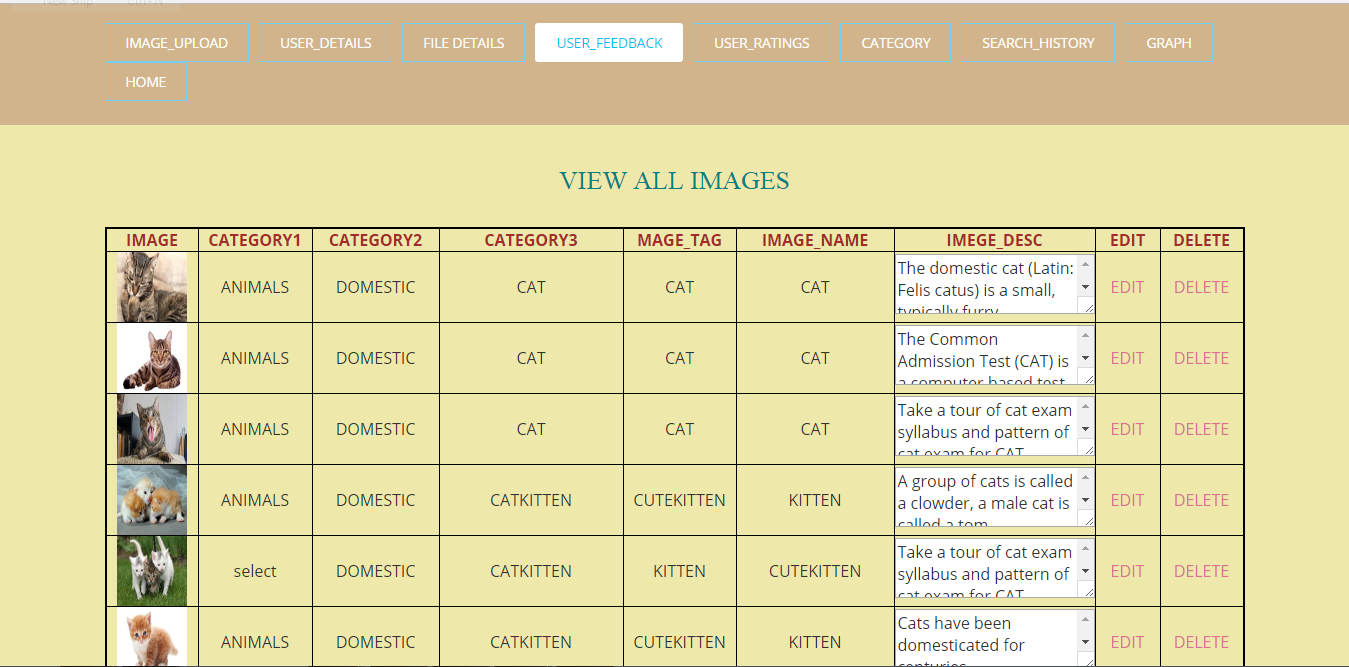






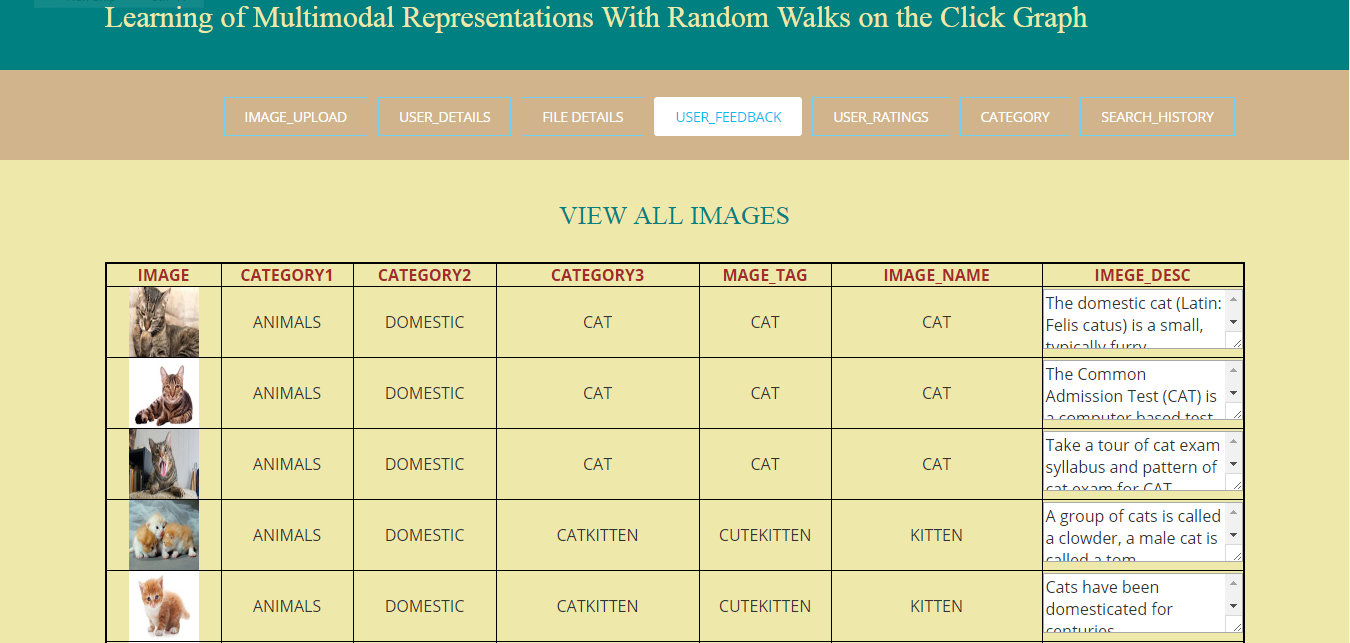






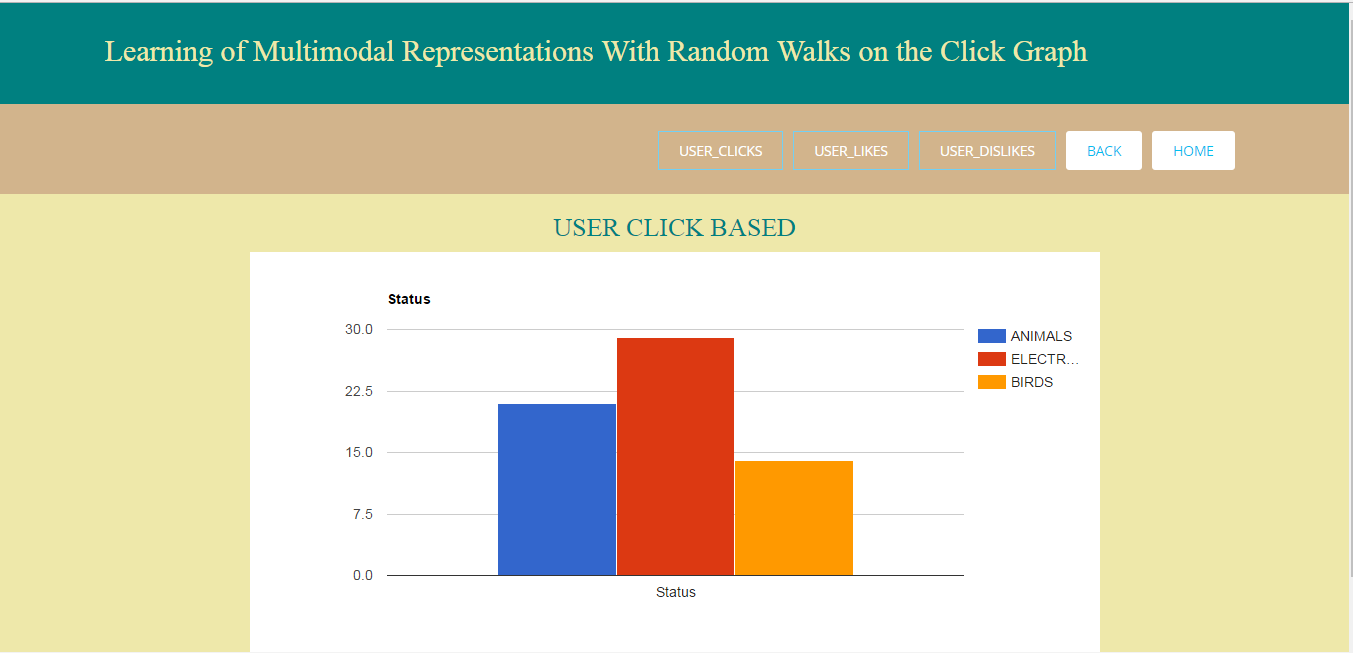


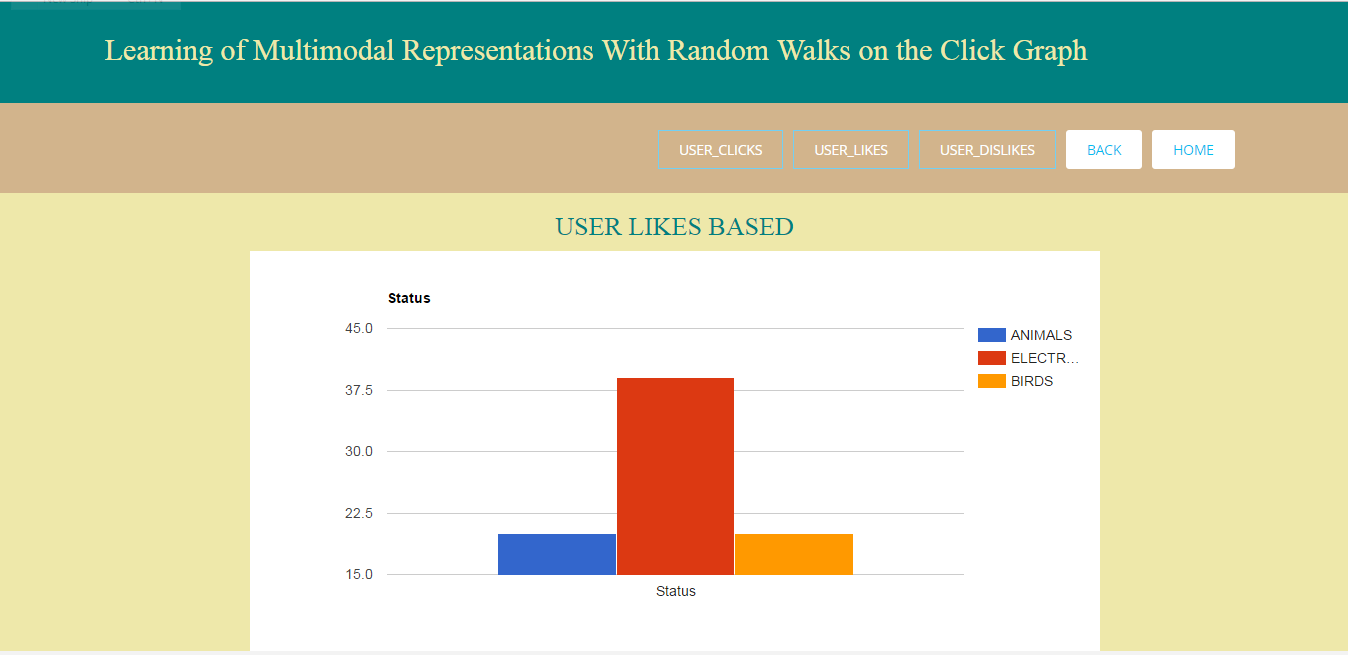


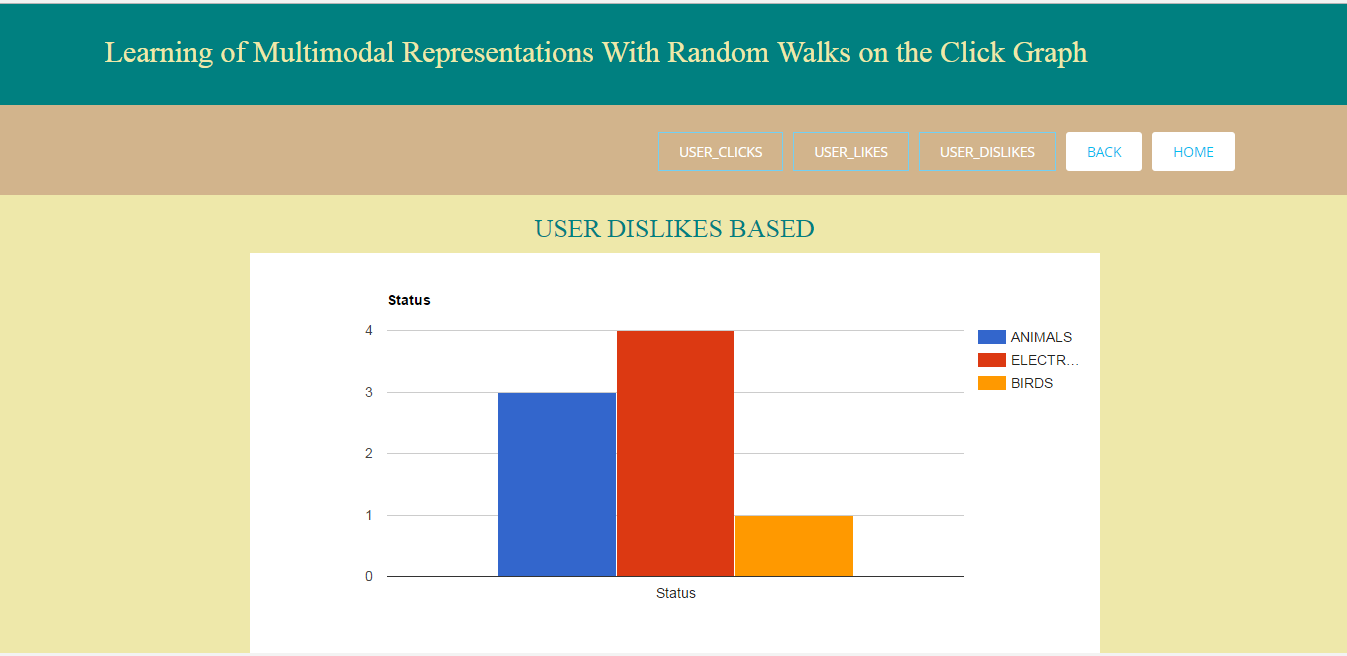












**CHAPERT-5**

**SYSTEM TESTING**

**Implementation and Testing:**

Implementation is one of the most important tasks in project is the phase in which one has to be cautions because all the efforts undertaken during the project will be very interactive. Implementation is the most crucial stage in achieving successful system and giving the users confidence that the new system is workable and effective. Each program is tested individually at the time of development using the sample data and has verified that these programs link together in the way specified in the program specification. The computer system and its environment are tested to the satisfaction of the user.

## Implementation

The implementation phase is less creative than system design. It is primarily concerned with user training, and file conversion. The system may be requiring extensive user training. The initial parameters of the system should be modifying as a result of a programming. A simple operating procedure is provided so that the user can understand the different functions clearly and quickly. The different reports can be obtained either on the inkjet or dot matrix printer, which is available at the disposal of the user. The proposed system is very easy to implement. In general implementation is used to mean the process of converting a new or revised system design into an operational one.

## Testing

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. Actually, testing is the state of implementation which aimed at ensuring that the system works accurately and efficiently before the actual operation commence. The following is the description of the testing strategies, which were carried out during the testing period.

### System Testing

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to user the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus, the code was exhaustively checked for all possible correct data and the outcomes were also checked.

### Module Testing

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus, all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example, the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. The comparison shows that the results proposed system works efficiently than the existing system. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

### Integration Testing

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system.

### Acceptance Testing

When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

**CHAPER-6**

**CONCLUSION**

In this work, we have presented a new approach to learning latent representation of the multimodal data from a click graph. By the minimization of the random walk error and the regularization penalty from the output of the modal-specific neural networks, the learned model has the ability not only to represent the explicit connections and the implicit connections of the vertices in the click graph with low-dimensional continuous vectors, but also to map the unseen queries and images to the latent subspace to support cross-modal retrieval. We have demonstrated the effectiveness of the learned representation by the proposed method MRW-NN and shown its superior to the comparative methods on cross-modal retrieval on a large-scale click log dataset.

**CHAPTER-7**

**REFERENCES**

[1] P. Mell and T. Grance, “The nist definition of cloud computing,” http://dx.doi.org/10.602/NIST.SP.800-145.

[2] K. Ren, C. Wang, and Q. Wang, “Security challenges for the public cloud,” IEEE Internet Computing, vol. 16, no. 1, pp. 69–73, 2012.

[3] S. Kamara and K. Lauter, “Cryptographic cloud storage,” in Springer RLCPS, January 2010.

[4] D. Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data,” in IEEE Symposiumon Security and Privacy, vol. 8, 2000, pp. 44–55.

[5] E.-J.Goh, “Secure indexes,” IACR ePrint Cryptography Archive, http://eprint.iacr.org/2003/216, Tech. Rep., 2003.

[6] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, “Public-key encryption with keyword search,” in EUROCRYPR, 2004, pp. 506–522.

[7] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, “Searchable symmetric encryption: improved deinitions and efficient constructions,” in ACM CCS, vol. 19, 2006, pp. 79–88.

[8] M. Bellare, A. Boldyreva, and A. O’Neill, “Deterministic and efficiently searchable encryption,” in Springer CRYPTO, 2007.

**SAMPLE CODE**

<%@page import="Dbcon.DbConnection"%>

<%@page import="java.sql.Connection"%>

<%@page import="java.sql.Statement"%>

<%@page import="java.sql.ResultSet"%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE HTML>

<html>

<head>

<title>Learning of Multimodal Representations</title>

<link href="css/bootstrap.css" rel='stylesheet' type='text/css' />

<link href="css/style.css" rel='stylesheet' type='text/css' />

<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<link href="css/main1.css" rel='stylesheet' type='text/css' />

<link href="css/main2.css" rel='stylesheet' type='text/css' />

<script type="application/x-javascript"></script>

<!--<link href="css/style.css" rel="stylesheet" type="text/css" media="all" />-->

<script src="js/jquery.min.js"></script>

<script type="text/javascript">

jQuery(document).ready(function($) {

$(".scroll").click(function(event){

event.preventDefault();

$('html,body').animate({scrollTop:$(this.hash).offset().top},1200);

});

});

</script>

<!-- grid-slider -->

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

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

<script type="text/javascript" src="js/jquery.easing.1.3.js"></script>

<!-- //grid-slider -->

</head>

<body>

<div class="header-bottom" style="background-color:teal">

<div class="container">

<div class="header-bottom\_left">

<h2><font style="font-family: Copperplate Gothic Bold; color:palegoldenrod ">Learning of Multimodal Representations With Random Walks on the Click Graph</font></h2><br>

</div><br>

<div class="social">

</div><br>

<div class="clear"></div>

</div>

</div>

<div class="menu" id="menu" style="background-color: tan">

<div class="container">

<div class="h\_menu4"><!-- start h\_menu4 -->

<a class="toggleMenu" href="#">Menu</a>

<ul class="nav">

<li><a href="imagupload.jsp">IMAGE\_UPLOAD</a></li>

<li><a href="auserdetail.jsp">USER\_DETAILS</a></li>

<li><a href="aviewalllimage.jsp">File Details</a></li>

<li class="active"><a href="userfeedback.jsp">USER\_FEEDBACK</a></li>

<li><a href="userratings.jsp">USER\_RATINGS</a></li>

<li><a href="category.jsp">CATEGORY</a></li>

<li><a href="asearckey.jsp">SEARCH\_HISTORY</a></li>

</ul>