**5. IMPLEMENTATION**

* 1. **MODULES**
* Application Layer
* Storage Logical Layer
* Object Store Layer
* Persistent Layer
  1. **MODULE DESCRIPTION**
* **Application Layer**: It consists of native software on desktop computers, mobile devices and web-interface, which allow user to upload, download and share their own files.
* **Storage Logical Layer**: it consisted of many queuing services and worker services, ID-Generator services and all logical API for Cloud Storage System. This layer implements business logic part in BFC.
* **Object Store Layer**: It contains many distributed backend services. Two important services of Object Store Layer are FileInfoService and ChunkStoreService. FileInfoService stores information of files. Y-value store mapping data from fileID to FileInfo structure. ChunkStoreService stores data chunks which are created by splitting from the original files that user uploaded.
* **Persistent Layer**: it based on ZDB key-value store. There are many ZDB instances  
  which are deployed as a distributed service and can be scaled when data growing.

**5.3 TECHNOLOGIES USED**

**5.3.1. ABOUT JAVA**

Initially the language was called as “oak” but it was renamed as “java” in 1995.The primary motivation of this language was the need for a platform-independent (i.e. architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices.

* Java is a programmer’s language
* Java is cohesive and consistent
* Except for that constraint imposed by the Internet environment. Java gives the programmer; full control Finally Java is to Internet Programming where c was to System Programming.
* **Importance of Java to the Internet**

Java has had a profound effect on the Internet. This is because; java expands the Universe of objects that can move about freely in Cyberspace. In a network, two categories of objects are transmitted between the server and the personal computer. They are passive information and Dynamic active programs. in the areas of Security and probability. But Java addresses these concerns and by doing so, has opened the door to an exciting new form of program called the Applet.

* **Applications and applets**

An application is a program that runs on computer under the operating system of that computer. It is more or less like one creating using C or C++. Java’s ability to create Applets makes it important. An Applet is the application, designed to be transmitted over the Internet and executed by a Java-compatible web browser. An applet I actually a tiny Java program, dynamically downloaded across the network, just like an image. But the difference is, it is an intelligent program, not just a media file. It can be react to the user input and dynamically change.

* **Java Architecture**

Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each platform by the run-time environment. Java is a dynamic system, able to load code when needed from a machine in the same room or across the planet.

* **Compilation of Code**

When you compile the code, the Java compiler creates machine code (called byte code)for a hypothetical machine called Java Virtual Machine(JVM). The JVM is supposed t executed the byte code. The JVM is created for the overcoming the issue of probability. The code is written and compiled for one machine and interpreted on all machines. This machine is called Java Virtual Machine.

* **Features of Java**

**Simple**

Java was designed to be easy for the Professional programmer to learn and to use effectively. If you are an experienced C++ Programmer. Learning Java will have oriented features of C++. Most of the confusing concepts from C++ are either left out of Java or implemented in a cleaner, more approachable manner. In Java there are a small number of clearly defined ways to accomplish a given task.

### Object oriented

Java was not designed to be source-code compatible with any other language. This allowed the Java team the freedom to design with a blank state. One outcome of this was a clean usable, pragmatic approach to objects. The object model in Java is simple and easy to extend, while simple types, such as integers, are kept as high-performance non-objects.

### Robust

The multi-platform environment of the web places extraordinary demands on a program, because the program must execute reliably in a variety of systems. The ability to create robust programs. Was given a high priority in the design of Java. Java is strictly typed language; it checks your code at compile time and runtime.

Java virtually eliminates the problems of memory management and deal location, which is completely automatic. In a well-written Java program, all run-time errors can and should be managed by your program.

**🡪AWT and Swings**

**AWT**

**Graphical User Interface**

The user interface is that part of a program that interacts with the user of the program. GUI is a type of [user interface](http://en.wikipedia.org/wiki/User_interface) that allows [users](http://en.wikipedia.org/wiki/User_(computing)) to [interact](http://en.wikipedia.org/wiki/Human-computer_interaction) with electronic devices with images rather than text commands. A class library is provided by the Java programming language which is known as Abstract Window Toolkit (AWT) for writing graphical programs. The Abstract Window Toolkit (AWT) contains several graphical widgets which can be added and positioned to the display area with a layout manager.

As the Java programming language, the AWT is not platform-independent. AWT uses system peers object for constructing graphical widgets. A common set of tools is provided by the AWT for graphical user interface design. The implementation of the user interface elements provided by the AWT is done using every platform's native GUI toolkit. One of the AWT's significance is that the look and feel of each platform can be preserved.

**5.3.2. My-SQL**

The Relational Database Management system My-SQL, which is used to organizes data in the form of relations or tables. The RDBMS My-SQL is one of various database servers based on Relational model, which is used to manages a prophet of data that attends three intended things with data structures, integrity of data and manipulation of data. The RDBMS My-SQL provides a cooperative server technology through which can realize the benefits of open network, open relational systems for all the Personalized applications. The RDBMS My- SQL makes well organized use of all systems resources, on all various hardware architecture; The RDBMS My-SQL is a open source Extended relational database management system It is named after co-founder Michael Widenius's.

**Features of My-SQL**

* + - 1. Provides Triggers Programming for Advanced Relational Model.
      2. Provides Dynamic Cursors Programming to control and programming at run time.
      3. The RDBMS provides Updatable views for fasting the selections and updating.
      4. The Information schema provides various role controlling on the Database.
      5. The advanced Independent storage engines controls and restricts the attackers to attack the data.
      6. Query caching Provides Faster Execution of Views for Selection.
      7. The Embedded database library Unicode supports the ACID compliance Cluster.
      8. The Partitioned tables provides with pruning of partitions in optimizer
      9. Custom storage engines Provides role based access and usage.

**5.3.3. BIG DATA TECHNOLOGY**

Big data is a buzzword, or one can say it’s a catch-phrase, which can be used to describe a huge volume of structured, unstructured, text, images, audio, video, log files, emails, simulations, 3D models, military surveillance, e-commerce and so on that is so massive that it's difficult to process using traditional database and software techniques. In most enterprise scenarios the data is too big or it moves too fast or it exceeds current processing capacity. Big data is nothing but a synonym of a huge and complex data that it becomes very tiresome, difficult or slow to collect, store, sort, process, retrieve and analyze it with the help of any existing relational database management tools or traditional data processing techniques. Big Data usually includes data sets with sizes beyond the ability of commonly used software tools to capture, curate, manage, and process the data within a tolerable elapsed time.

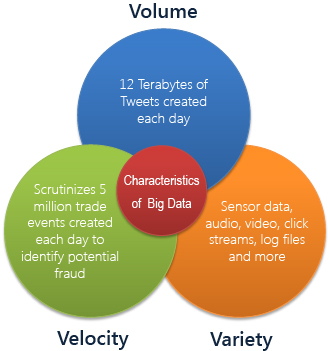
**Characteristics of Big Data:**

Data scientists break big data into four dimensions: volume, variety, velocity and veracity.

**Volume**: BIG DATA depends upon how large it is. It could amount to hundreds of terabytes or even petabytes of information.

**Velocity**: The increasing rate at which data flows into an organization.

**Variety**: A common theme in big data systems is that the source data is diverse and doesn’t fall into neat relational structures.

****

**Figure 5.3.3(a):** Characteristics’ of BIG DATA

**How Fast Data Is Increasing**

Carefully look the picture which explains us “what happens in every 60 seconds on the internet“. Understand how much data being generated in a second, a minute, a day or a year and how exponentially it is generating. As per the analysis by TechNewsDaily might generate more than 8 Zettabytes of data by 2015.

[](http://www.stratapps.net/images/DataIncreasing.jpg)

**Figure 5.3.3.(b):** Big Data In Internet

**5.4. SAMPLE CODE**

**🡪Register.java**

package BFC;

import java.awt.Dimension;

import java.awt.Font;

import java.awt.Toolkit;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.ObjectOutputStream;

import java.io.ObjectInputStream;

import java.net.Socket;

import javax.swing.JButton;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JPasswordField;

import javax.swing.JTextField;

import net.miginfocom.swing.MigLayout;

import java.awt.Color;

import javax.swing.JFrame;

public class Register extends JFrame{

JPanel p1;

Font f1;

JLabel l1,l2,l3,l4,l5;

JTextField tf1,tf2,tf3,tf4,tf5;

JButton b1,b2;

Dimension d;

int w,h;

int center;

public Register(){

super("New User Registration Screen");

d = Toolkit.getDefaultToolkit().getScreenSize();

w = (int)d.getWidth();

h = (int)d.getHeight();

center = w/30;

setBackground(Color.white);

p1 = new JPanel();

p1.setBackground(Color.white);

p1.setLayout(new MigLayout("wrap 2","","[]15[]"));

f1 = new Font("Verdana",Font.PLAIN,14);

p1.add(new JLabel(),"cell 0 30");

p1.add(new JLabel(),"cell 1 30");

l1 = new JLabel("Username");

l1.setFont(f1);

p1.add(l1,"gap left "+center);

tf1 = new JTextField(12);

tf1.setFont(f1);

p1.add(tf1,"gap left 30");

l2 = new JLabel("Password");

l2.setFont(f1);

p1.add(l2,"gap left "+center);

tf2 = new JPasswordField(12);

tf2.setFont(f1);

p1.add(tf2,"gap left 30");

l3 = new JLabel("Contact No");

l3.setFont(f1);

p1.add(l3,"gap left "+center);

tf3 = new JTextField(12);

tf3.setFont(f1);

p1.add(tf3,"gap left 30");

l4 = new JLabel("Email ID");

l4.setFont(f1);

p1.add(l4,"gap left "+center);

tf4 = new JTextField(12);

tf4.setFont(f1);

p1.add(tf4,"gap left 30");

l5 = new JLabel("Address");

l5.setFont(f1);

p1.add(l5,"gap left "+center);

tf5 = new JTextField(30);

tf5.setFont(f1);

p1.add(tf5,"gap left 30");

p1.add(new JLabel(""));

b1 = new JButton("Register");

b1.setFont(f1);

p1.add(b1,"split 2");

b1.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent ae){

process();

}

});

b2 = new JButton("Clear");

b2.setFont(f1);

p1.add(b2);

b2.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent ae){

clearFields();

}

});

add(p1);

}

public void process(){

try{

String uname=tf1.getText();

String pass=tf2.getText();

String contact=tf3.getText();

String email=tf4.getText();

String address=tf5.getText();

if(uname.length() <=0 || uname == null){

JOptionPane.showMessageDialog(this,"Username must be enter");

tf1.requestFocus();

return;

}

if(pass.length() <=0 || pass == null){

JOptionPane.showMessageDialog(this,"Password must be enter");

tf2.requestFocus();

return;

}

if(contact.length() <=0 || contact == null){

JOptionPane.showMessageDialog(this,"Contact No must be enter");

tf3.requestFocus();

return;

}

if(contact.length() <=0 || contact == null){

JOptionPane.showMessageDialog(this,"Contact no must be enter");

tf3.requestFocus();

return; }

if(!validatePhoneNumber(contact.trim())){

JOptionPane.showMessageDialog(this,"Enter valid contact no");

tf3.requestFocus();

return;

}

if(email.length() <=0 || email == null){

JOptionPane.showMessageDialog(this,"Email id must be enter");

tf4.requestFocus();

return;

}

if(!CheckMail.checkMail(email)){

JOptionPane.showMessageDialog(this,"Enter valid mailid");

tf4.requestFocus();

return;

}

if(address.length() <=0 || address == null){

JOptionPane.showMessageDialog(this,"Address must be enter");

tf5.requestFocus();

return;

}

Socket socket=new Socket("localhost",1200);

ObjectOutputStream out=new ObjectOutputStream(socket.getOutputStream());

ObjectInputStream in=new ObjectInputStream(socket.getInputStream());

Object req[]={"register",uname,pass,contact,email,address};

out.writeObject(req);

out.flush();

Object res[]=(Object[])in.readObject();

String msg = res[0].toString();

if(msg.equals("Registration process completed")){

JOptionPane.showMessageDialog(this,msg);

setVisible(false);

}else{

JOptionPane.showMessageDialog(this,msg);

}

}catch(Exception e){

e.printStackTrace();

}

}

public void clearFields(){

tf1.setText("");

tf2.setText("");

tf3.setText("");

tf4.setText("");

tf5.setText("");

}

private static boolean validatePhoneNumber(String phoneNo){

//validate phone numbers of format "1234567890"

if(phoneNo.matches("\\d{10}"))

return true;

//validating phone number with -, . or spaces

else if(phoneNo.matches("\\d{3}[-\\.\\s]\\d{3}[-\\.\\s]\\d{4}"))

return true;

//validating phone number with extension length from 3 to 5

else if(phoneNo.matches("\\d{3}-\\d{3}-\\d{4}\\s(x|(ext))\\d{3,5}"))

return true;

//validating phone number where area code is in braces ()

else if(phoneNo.matches("\\(\\d{3}\\)-\\d{3}-\\d{4}"))

return true;

//return false if nothing matches the input

else

return false;

}}

**Login.java**

package BFC;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.ImageIcon;

import javax.swing.JLabel;

import javax.swing.JTextField;

import javax.swing.JButton;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.BorderLayout;

import java.awt.Color;

import javax.swing.JPasswordField;

import javax.swing.UIManager;

import java.io.ObjectOutputStream;

import java.io.ObjectInputStream;

import java.net.Socket;

import javax.swing.JOptionPane;

public class Login extends JFrame

{

JLabel l1,l2,l3,l4;

JTextField tf1,tf2;

JButton b1,b2,b3;

Font f1,f2;

JPanel p1,p2,p3,p4,p5,p6;

ImageIcon icon;

public Login(){

super("Login Screen");

p1 = new JPanel();

p1.setBackground(Color.black);

f1 = new Font("Monospaced",Font.BOLD,22);

l1=new JLabel("<html><body><center>User Login Screen</center></body></html>");

l1.setForeground(new Color(125,54,2));

l1.setFont(f1);

p1.add(l1);

p1.setBackground(new Color(140,150,180));

p2 = new JPanel();

p2.setBackground(Color.black);

icon = new ImageIcon("img/vista.jpg");

JLabel label = new JLabel(icon);

p2.add(label);

p3 = new JPanel();

p3.setLayout(new BorderLayout());

p4 = new JPanel();

f2 = new Font("Verdana",Font.PLAIN,14);

l3 = new JLabel("Username");

l3.setFont(f2);

p4.add(l3);

tf1 = new JTextField(15);

tf1.setFont(f2);

p4.add(tf1);

p5 = new JPanel();

}

}

}