METHODIST

College of Engineering &
Technology

(Affiliated to OU)

Department of CSE

DISTRIBUTED SYSTEMS LAB

LAB MANUAL

INDEX

1) **Implementation FTP Client:**

FTP Client:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.net.*;
import java.io.*;
 class One extends JFrame implements ActionListener
{
        /* ctrl space */
        public JButton b,b1;
        public JLabel l;
        public  JLabel l1,lmsg1,lmsg2;
        One()
        {
                b=new JButton("Upload");
                l=new JLabel("Uplaod a file : ");
                lmsg1=new JLabel("");

                b1=new JButton("Download");
                l1=new JLabel("Downlaod a file");
                lmsg2=new JLabel("");

                setLayout(new GridLayout(2,3,10,10));
                add(l);add(b);add(lmsg1);add(l1);add(b1);add(lmsg2);
                b.addActionListener(this);
                b1.addActionListener(this);
                setVisible(true);
                setSize(600,500);
            }
        public  void actionPerformed(ActionEvent e)
          {
                // TODO Auto-generated method stub
            try {
                /* String s=e.getActionCommand();
                if(s.equals("Upload"))*/

                if (b.getModel().isArmed())
                {

                        Socket s=new Socket("localhost",1010);
                        System.out.println("Client connected to server");
                        JFileChooser j=new JFileChooser();
                        int val;
                         val=j.showOpenDialog(One.this);
                           String filename=j.getSelectedFile().getName();
                           String path=j.getSelectedFile().getPath();

                            PrintStream out=new
PrintStream(s.getOutputStream());
                        out.println("Upload");
                        out.println(filename);
                        FileInputStream fis=new FileInputStream(path);
                        int n=fis.read();
                        while (n!=-1)
                            {
                        out.print((char)n);n=fis.read();
                        }
                       fis.close(); out.close();lmsg1.setText(filename+"is
uploaded");
                      //s.close();
                     repaint();
                 }

                if (b1.getModel().isArmed())
                    {
                        Socket s=new Socket("localhost",1010);
                        System.out.println("Client connected to server");
                        String
remoteadd=s.getRemoteSocketAddress().toString();
                        System.out.println(remoteadd);
                        JFileChooser j1=new JFileChooser(remoteadd);
                        int val;
                        val=j1.showOpenDialog(One.this);
                          String filename=j1.getSelectedFile().getName();
                          String filepath=j1.getSelectedFile().getPath();

                        System.out.println("File name:"+filename);
                        PrintStream out=new
PrintStream(s.getOutputStream());
                        out.println("Download");
                        out.println(filepath);

                        FileOutputStream fout=new
FileOutputStream(filename);
                        DataInputStream fromserver=new
DataInputStream(s.getInputStream());
                        int ch;
                        while ((ch=fromserver.read())!=-1)
 {
                                fout.write((char) ch);
                        }
                        fout.close();//s.close();
                        lmsg2.setText(filename+"is downlaoded");
                        repaint();
                      }
                }
catch (Exception ee)
{
                // TODO: handle exception
                System.out.println(ee);
                }
        }
```

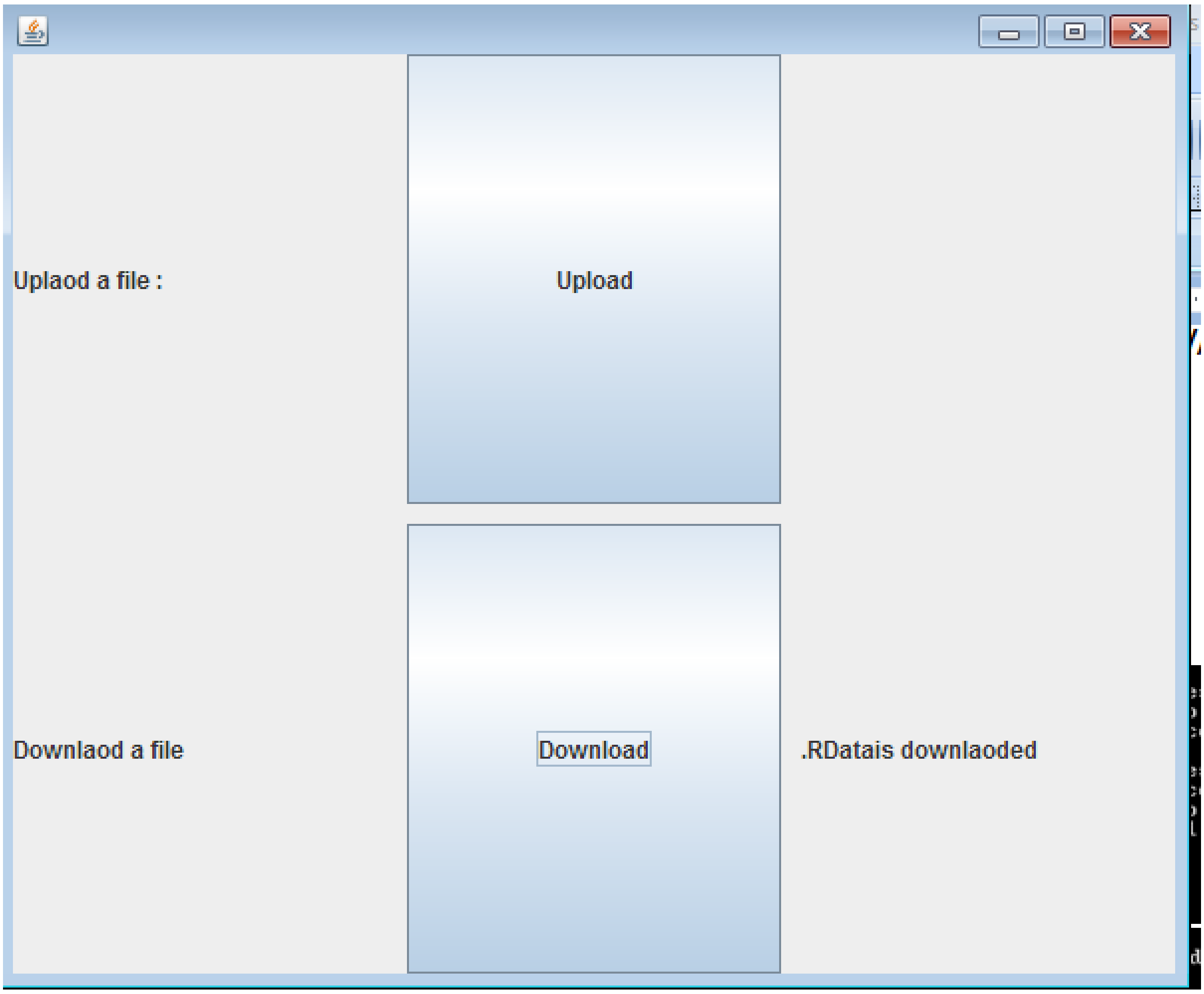
```
}
public class FTPClient
{
     public static void main(String[] args)
 {
          new One();
     }
}
```

FTP Server:

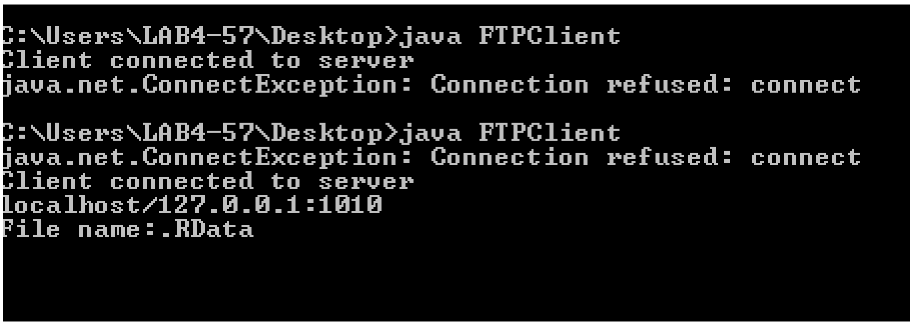
```
import java.io.DataInputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.PrintStream;
import java.net.ServerSocket;
import java.net.Socket;


public class FTPServer {
public static void main(String[] args)
 {
     try {
          while (true)
{
          ServerSocket ss=new ServerSocket(1010);
          Socket sl=ss.accept();
          System.out.println("Server scoket is created....");
          System.out.println(" test1");
          DataInputStream fromserver=new
DataInputStream(sl.getInputStream());
          System.out.println(" test2");
          String option=fromserver.readLine();
          if (option.equalsIgnoreCase("upload"))
{
          System.out.println("upload test");
          String filefromclient=fromserver.readLine();
          File clientfile=new File(filefromclient);

          FileOutputStream fout=new FileOutputStream(clientfile);
          int ch;
          while ((ch=fromserver.read())!=-1)
{
            fout.write((char)ch);
          }
          fout.close();
          }
          if (option.equalsIgnoreCase("download"))
{
          System.out.println("download test");
          String filefromclient=fromserver.readLine();
          File clientfile=new File(filefromclient);

          FileInputStream fis=new FileInputStream(clientfile);

          PrintStream out=new PrintStream(sl.getOutputStream());
          int n=fis.read();
          while (n!=-1)
 {
            out.print((char)n);
            n=fis.read();
          }
          fis.close();
          out.close();

      } //while
     }
    }
catch (Exception e)
```

```
{
                        System.out.println(e);
                        // TODO: handle exception
        }
       }
}
```

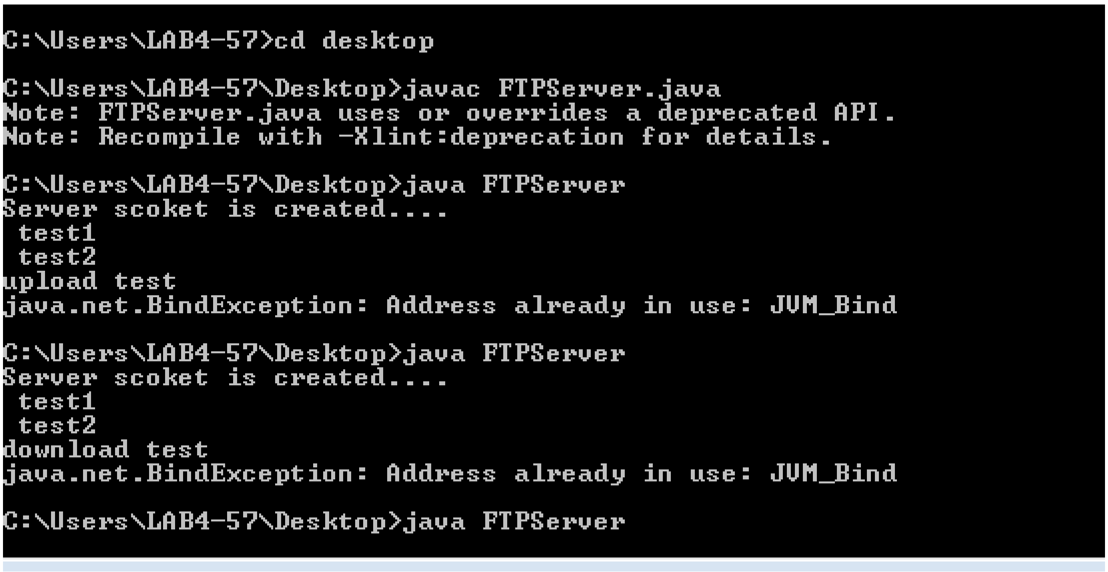
Output:



.RDatais downlaoded



```
C:\Users\LAB4-57\Desktop>java FTPClient
Client connected to server
java.net.ConnectException: Connection refused: connect

C:\Users\LAB4-57\Desktop>java FTPClient
java.net.ConnectException: Connection refused: connect
Client connected to server
localhost/127.0.0.1:1010
File name:.RData
```



```
C:\Users\LAB4-57>cd desktop

C:\Users\LAB4-57\Desktop>javac FTPServer.java
Note: FTPServer.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\Users\LAB4-57\Desktop>java FTPServer
Server scoket is created....
 test1
 test2
upload test
java.net.BindException: Address already in use: JVM_Bind

C:\Users\LAB4-57\Desktop>java FTPServer
Server scoket is created....
 test1
 test2
download test
java.net.BindException: Address already in use: JVM_Bind

C:\Users\LAB4-57\Desktop>java FTPServer
```

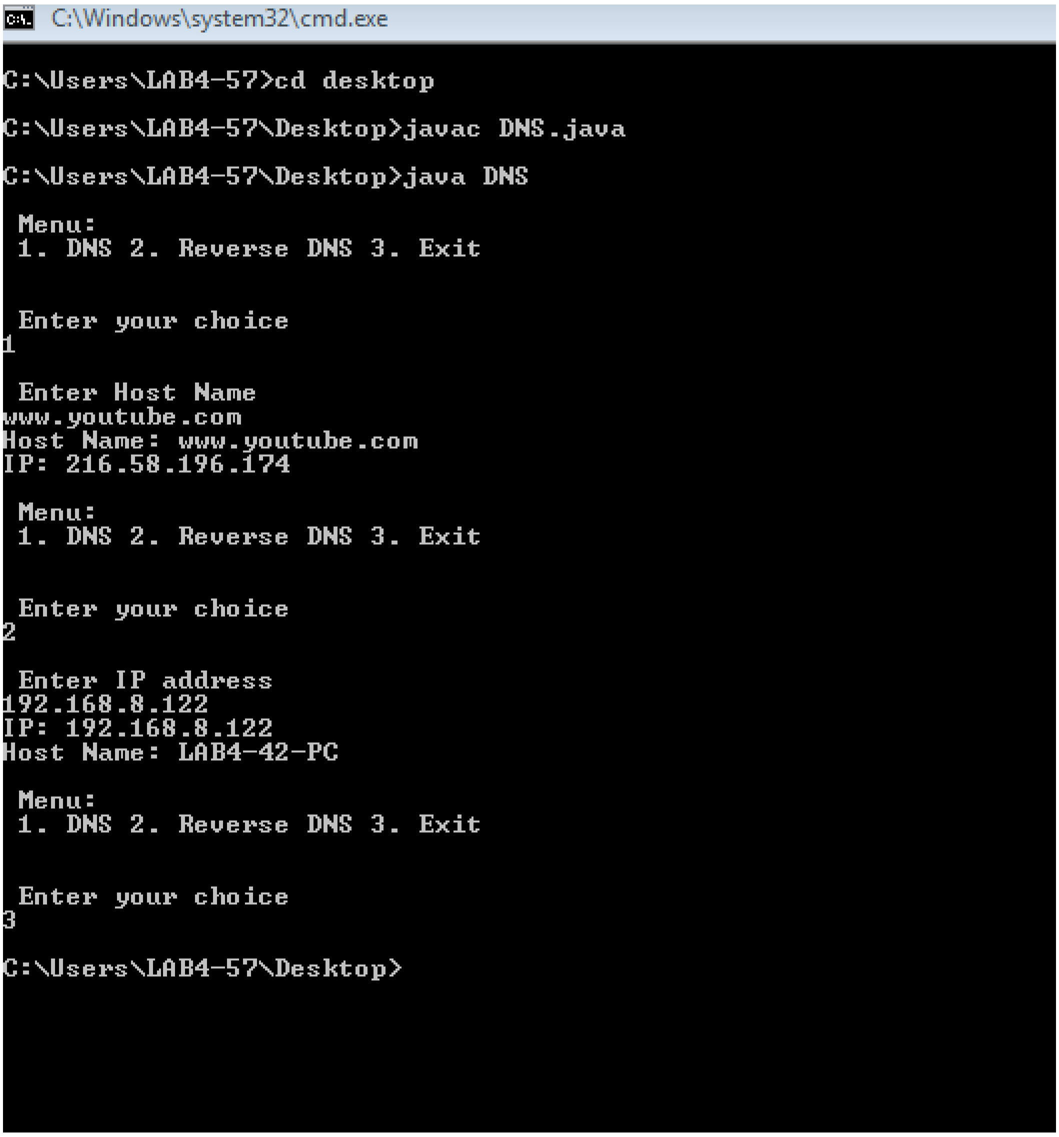
2) Implementation of Name Server

```
import java.net.*;

import java.io.*;

import java.util.*;


public class DNS

{

 public static void main(String[] args)

 {

  int n;

  BufferedReader in = new BufferedReader(new
InputStreamReader(System.in));

  do

  {

   System.out.println("\n Menu: \n 1. DNS 2. Reverse DNS 3. Exit \n");

   System.out.println("\n Enter your choice");
```

```
n = Integer.parseInt(System.console().readLine());
if(n==1)
{
 try
 {
  System.out.println("\n Enter Host Name ");
  String hname=in.readLine();
  InetAddress address;
  address = InetAddress.getByName(hname);
  System.out.println("Host Name: " + address.getHostName());
  System.out.println("IP: " + address.getHostAddress());
 }
 catch(IOException ioe)
 {
  ioe.printStackTrace();
 }
}
if(n==2)
{
 try
 {
    System.out.println("\n Enter IP address");
    String ipstr = in.readLine();
```

```
       InetAddress ia = InetAddress.getByName(ipstr);
       System.out.println("IP: "+ipstr);
       System.out.println("Host Name: " +ia.getHostName());
     }
    catch(IOException ioe)
    {
     ioe.printStackTrace();
    }
   }
  }while(!(n==3));
 }
}
```

Output:



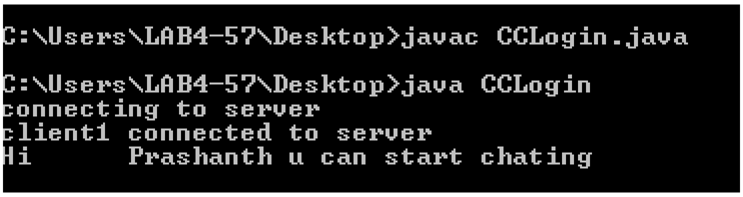
```
C:\Users\LAB4-57>cd desktop

C:\Users\LAB4-57\Desktop>javac DNS.java

C:\Users\LAB4-57\Desktop>java DNS

 Menu:
 1. DNS 2. Reverse DNS 3. Exit


 Enter your choice
1

 Enter Host Name
www.youtube.com
Host Name: www.youtube.com
IP: 216.58.196.174

 Menu:
 1. DNS 2. Reverse DNS 3. Exit


 Enter your choice
2

 Enter IP address
192.168.8.122
IP: 192.168.8.122
Host Name: LAB4-42-PC

 Menu:
 1. DNS 2. Reverse DNS 3. Exit


 Enter your choice
3

C:\Users\LAB4-57\Desktop>
```

3) Implementation of Chat Server

CCLogin.java

```
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.IOException;
import javax.swing.JButton;
 import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;
import java.awt.GridLayout;

public class CCLogin implements ActionListener
{
 JFrame frame1; JTextField tf,tf1; JButton button;
         JLabel heading; JLabel label,label1;

        public static void main(String[] paramArrayOfString)
         {
        new  CCLogin();
}

        public  CCLogin()
 {
         this.frame1 = new JFrame("Login Page");
         this.tf = new JTextField(10);
          this.button = new JButton("Login");

        this.heading = new JLabel("Chat Server");
        this.heading.setFont(new Font("Impact", 1, 40));
         this.label = new JLabel("Enter you Login Name");
           this.label.setFont(new Font("Serif", 0, 24));

        JPanel localJPanel = new JPanel();
          this.button.addActionListener(this);
          localJPanel.add(this.heading); localJPanel.add(this.label);
        localJPanel.add(this.tf);
        localJPanel.add(this.button);
         this.heading.setBounds(30, 20, 280, 50);
           this.label.setBounds(20, 100, 250, 60);
           this.tf.setBounds(50, 150, 150, 30);
         this.button.setBounds(70, 190, 90, 30);
        this.frame1.add(localJPanel);
        localJPanel.setLayout(null);
```

```
          this.frame1.setSize(300,300);
          this.frame1.setVisible(true);
          this.frame1.setDefaultCloseOperation(3);
 }

 public void actionPerformed(ActionEvent paramActionEvent)
{
   String str = "";
  try
 {
     str = this.tf.getText();
     this.frame1.dispose();
     Client1 c1= new Client1(str);
     c1.main(null);
   }
   catch(Exception localIOException)
   {
   }
 }
}
```

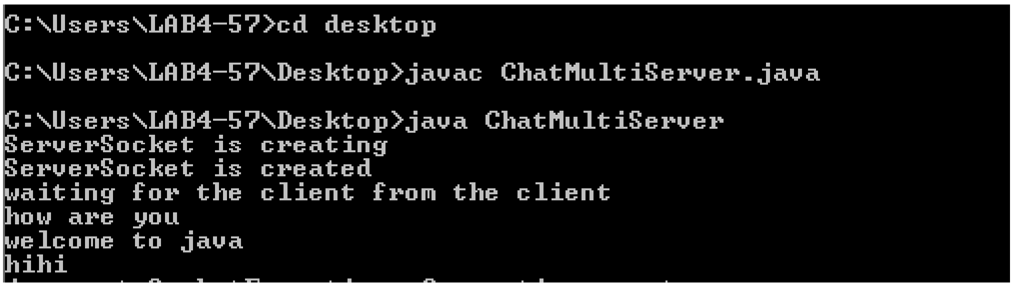


```
C:\Users\LAB4-57\Desktop>javac CCLogin.java

C:\Users\LAB4-57\Desktop>java CCLogin
connecting to server
client1 connected to server
Hi      Prashanth u can start chating
```

ChatMultiServer:

```
import java.net.*;
import java.io.*;
class A implements Runnable
{
  Thread t;
  Socket s;
  A(Socket x)
  {
   s=x;
   t=new Thread(this);
   t.start();
  }
  public void run()
  {
   try
   {
            /* Reading data from client */
           InputStream is=s.getInputStream();
           byte data[]=new byte[50];
                 is.read(data);
           String mfc=new String(data);
           mfc=mfc.trim();
           System.out.println(mfc);

           /* Sending message to the server */
           //System.out.println("Hi"+name+"u can start chating");
           BufferedReader br=new BufferedReader(new
InputStreamReader(System.in));
           String n=br.readLine();
           OutputStream os=s.getOutputStream();
           os.write(n.getBytes());
       }
   catch(Exception e)
   {
    e.printStackTrace();
   }
  }
}

class ChatMultiServer
{
static int c=0;
```

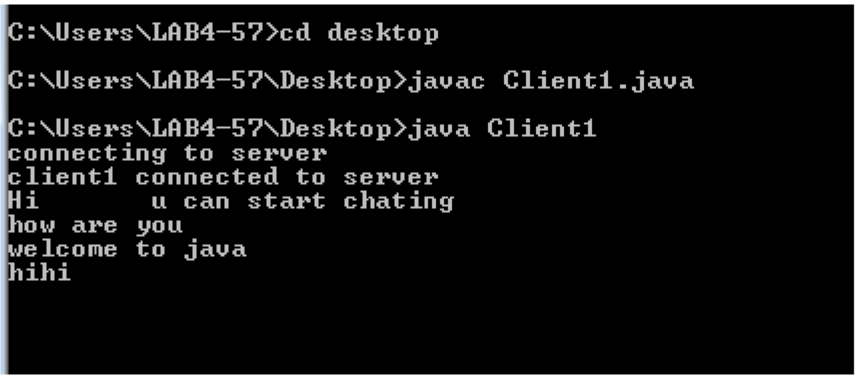
```
 public static void main(String args[]) throws Exception
 {
   System.out.println("ServerSocket is creating");
   ServerSocket ss=new ServerSocket(1010);
   System.out.println("ServerSocket is created");
   System.out.println("waiting for the client from the client");

         while(true)
   {
             Socket s=ss.accept();
             new A(s);
   }
 }
}
```



```
C:\Users\LAB4-57>cd desktop

C:\Users\LAB4-57\Desktop>javac ChatMultiServer.java

C:\Users\LAB4-57\Desktop>java ChatMultiServer
ServerSocket is creating
ServerSocket is created
waiting for the client from the client
how are you
welcome to java
hihi
```

```
Client1.java
import java.net.*;
import java.io.*;
class Client1
{
 static String name="";
 public Client1(String n)
 {
             name=n;
 }
 public static void main(String args[]) throws Exception
 {
  System.out.println("connecting to server");
  System.out.println("client1 connected to server");
```

```
    BufferedReader br=new BufferedReader(new
InputStreamReader(System.in));

            /* Sending message to the server */
         System.out.println("Hi\t"+name+" u can start chating");
         while(true)
         {
                  Socket s=new Socket("localhost",1010);
                  String n=br.readLine();
                  OutputStream os=s.getOutputStream();
                  os.write(n.getBytes());

                   /* Reading data from client */
                  InputStream is=s.getInputStream();
                  byte data[]=new byte[50];
                           is.read(data);
                  String mfc=new String(data);
                           mfc=mfc.trim();
                  System.out.println(mfc);

          }
       }
}
```



```
C:\Users\LAB4-57>cd desktop

C:\Users\LAB4-57\Desktop>javac Client1.java

C:\Users\LAB4-57\Desktop>java Client1
connecting to server
client1 connected to server
Hi       u can start chating
how are you
welcome to java
hihi
```

4)Understanding of Working of NFS(includes exercises Configuration of NFS)

Study of NFS.

1. Create a Folder nfs/abc.txt

2. Know the ipaddress

Applications->System Settings->Network—edit (ipaddress, subnetmask)

(or) In terminal type ifconfig

3. Enable the desired services

1. System Services->Server Settings->Services

- **Network (Enable)**
- **Nfs (Enable)**
- **Iptables (Disable) (we do not firewalls)**

2. System Settings ->Security Level (Firewall options-disable, Selinux-disable)

Creation of NFS Server

1. **System Settings->Server Settings->NFS**
   + Add (All are making security levels low)
2. **Open Terminal**
   Type: service nfs restart
   Creation of NFS Client
   Open terminal
   Type: df
   Type: mount –t nfs 135.135.5.120:/usr/nfs /root/abc
   cd abc
   ls : abc.txt
   Unmount: umount –t nfs 135.135.5.120:/usr/nfs
   Note: service network restart (if n/w is disabled use this)

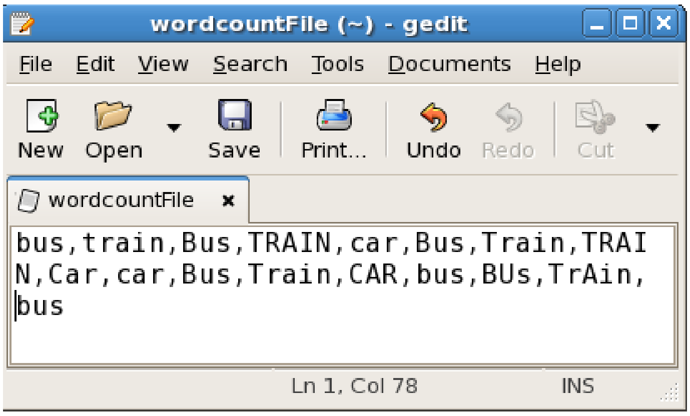
5) Implementation of Bulletin Board.

6) Implement a word count application which counts the number of occurrences of each word a large collection of documents Using Map Reduce model.

```
package PackageDemo;
import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import
org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;
public class WordCount {
public static void main(String [] args) throws Exception
{
Configuration c=new Configuration();
String[] files=new
GenericOptionsParser(c,args).getRemainingArgs();
Path input=new Path(files[0]);
Path output=new Path(files[1]);
Job j=new Job(c,"wordcount");
j.setJarByClass(WordCount.class);
j.setMapperClass(MapForWordCount.class);
j.setReducerClass(ReduceForWordCount.class);
j.setOutputKeyClass(Text.class);
j.setOutputValueClass(IntWritable.class);
FileInputFormat.addInputPath(j, input);
FileOutputFormat.setOutputPath(j, output);
System.exit(j.waitForCompletion(true)?0:1);
}
public static class MapForWordCount extends
Mapper<LongWritable, Text, Text, IntWritable>{
public void map(LongWritable key, Text value, Context con)
throws IOException, InterruptedException
{
String line = value.toString();
String[] words=line.split(",");
for(String word: words )
{
      Text outputKey = new Text(word.toUpperCase().trim());
  IntWritable outputValue = new IntWritable(1);
  con.write(outputKey, outputValue);
}
}
}
public static class ReduceForWordCount extends Reducer<Text,
IntWritable, Text, IntWritable>
{
public void reduce(Text word, Iterable<IntWritable> values,
Context con) throws IOException, InterruptedException
{
int sum = 0;
   for(IntWritable value : values)
   {
   sum += value.get();
   }
   con.write(word, new IntWritable(sum));
}
}
}
```

Output:

1. Take a text file and move it into HDFS format:



wordcountFile (~) - gedit
File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut
wordcountFile
Ln 1, Col 78
INS

```
bus,train,Bus,TRAIN,car,Bus,Train,TRAIN,Car,car,Bus,Train,CAR,bus,BUs,TrAin,bus
```

To move this into Hadoop directly, open the terminal and enter the following commands:

```
[training@localhost ~]$ hadoop fs -put wordcountFile
wordCountFile
```

2. Run the jar file:

(Hadoop jar jarfilename.jar packageName.ClassName PathToInputTextFile PathToOutputDirectry)

```
[training@localhost ~]$ hadoop jar MRProgramsDemo.jar
PackageDemo.WordCount wordCountFile MRDir1
```

3. Open the result:

```
[training@localhost ~]$ hadoop fs -ls MRDir1
Found 3 items
-rw-r--r--   1 training supergroup          0 2016-02-23 03:36
/user/training/MRDir1/_SUCCESS
drwxr-xr-x   - training supergroup          0 2016-02-23 03:36
/user/training/MRDir1/_logs
-rw-r--r--   1 training supergroup         20 2016-02-23 03:36
/user/training/MRDir1/part-r-00000
[training@localhost ~]$ hadoop fs -cat MRDir1/part-r-00000
BUS     7
CAR     4
TRAIN   6
```

7) Develop an application (small game like scrabble, Tic-tac-Toe Using Android SDK)

Creating the Board

First step is to create the Board for the Tic-Tac-Toe game. The *Board* class will store the elements of the grid in an array and will contain a boolean indicating if the game is ended or no.

The *play* method will let you to set the mark of the *currentPlayer* on the grid at a given (x,y) position. A *changePlayer* method will be used to change the current player for the next play. Besides, a *computer* method is defined to let the user to randomly place a mark on the grid. Finally, we define a *checkEnd* method to check if the game is ended. The game is ended if there is a winner or a draw : all the cases of the grids are filled and no one wins the game.

This gives us the following code for the *Board* class :

```
package
com.ssaurel.tictactoe;
import java.util.Random;
public class Board {
    private static final Random RANDOM = new
Random();
    private char[] elts;
    private char currentPlayer;
    private boolean ended;
    public Board() {
        elts = new char[9];
        newGame();
    }
    public boolean isEnded() {
        return ended;
    }
    public char play(int x, int y) {
        if (!ended  &&  elts[3 * y + x] == ' ') {
            elts[3 * y + x] = currentPlayer;
            changePlayer();
        }
        return checkEnd();
    }
    public void changePlayer() {
        currentPlayer = (currentPlayer == 'X' ? 'O'
: 'X');
    }
    public char getElt(int x, int y) {
        return elts[3 * y + x];
    }
    public void newGame() {
        for (int i = 0; i  < elts.length; i++) {
            elts[i] = ' ';
        }
        currentPlayer = 'X';
        ended = false;
    }
    public char checkEnd() {
        for (int i = 0; i < 3; i++) {
            if (getElt(i, 0) != ' ' &&
                getElt(i, 0) == getElt(i, 1)  &&
                getElt(i, 1) == getElt(i, 2)) {
                ended = true;
                return getElt(i, 0);
            }
            if (getElt(0, i) != ' ' &&
                getElt(0, i) == getElt(1, i)  &&
                getElt(1, i) == getElt(2, i)) {
                ended = true;
                return getElt(0, i);
            }
        }
        if (getElt(0, 0) != ' '  &&
            getElt(0, 0) == getElt(1, 1)  &&
            getElt(1, 1) == getElt(2, 2)) {
            ended = true;
            return getElt(0, 0);
        }
        if (getElt(2, 0) != ' '  &&
            getElt(2, 0) == getElt(1, 1)  &&
            getElt(1, 1) == getElt(0, 2)) {
            ended = true;
            return getElt(2, 0);
        }
        for (int i = 0; i < 9; i++) {
            if (elts[i] == ' ')
                return ' ';
        }
        return 'T';
    }
    public char computer() {
        if (!ended) {
            int position = -1;
            do {
                position = RANDOM.nextInt(9);
            } while (elts[position] != ' ');
            elts[position] = currentPlayer;
            changePlayer();
        }
        return checkEnd();
    }
}
```

Rendering the Board on the Screen

Next step is to create a *BoardView* class to render our Board on the screen. Our *BoardView* will extend the *View* class and we will draw the *Board* and its elements on the *Canvas* object associated. It is a good way to discover how to draw simple shapes on a *Canvas* of a specific *View* too.

Furthermore, we must manage the touch events of the users on the *Board* to let it to play to our Tic-Tac-Toe game. For that, we override the *onTouchEvent* method from the *View* parent class. In that method, we convert a point touched on the screen to a case on our grid. Then, we make the play on the *Board* object. After that, we need to call the gameEnded method of the parent activity if the game is ended to display the win dialog to the user. If not, we make the play for the computer. Like you can see, the heart of the logic game will be located in this method.

This gives us the following code for the *BoardView* object :

```
package
com.ssaurel.tictactoe;
                      import android.content.Context;
                      import android.graphics.Canvas;
                      import android.graphics.Color;
                      import android.graphics.Paint;
                      import android.support.annotation.Nullable;
                      import android.util.AttributeSet;
                      import android.view.MotionEvent;
                      import android.view.View;
                      public class BoardView extends View {
                          private static final int LINE_THICK = 5;
                          private static final int ELT_MARGIN = 20;
                          private static final int ELT_STROKE_WIDTH = 15;
                          private int width, height, eltW, eltH;
                          private Paint gridPaint, oPaint, xPaint;
                          private GameEngine gameEngine;
                          private MainActivity activity;
                          public BoardView(Context context) {
                              super(context);
                          }
                          public BoardView(Context context, @Nullable
                      AttributeSet attrs) {
                              super(context, attrs);
                              gridPaint = new Paint();
                              oPaint = new Paint(Paint.ANTI_ALIAS_FLAG);
                              oPaint.setColor(Color.RED);
                              oPaint.setStyle(Paint.Style.STROKE);
                              oPaint.setStrokeWidth(ELT_STROKE_WIDTH);
                              xPaint = new Paint(oPaint);
                              xPaint.setColor(Color.BLUE);
                          }
                          public void setMainActivity(MainActivity a) {
                              activity = a;
```

```
    }
    public void setGameEngine(GameEngine g) {
        gameEngine = g;
    }
    @Override
    protected void onMeasure(int widthMeasureSpec,
int heightMeasureSpec) {
        height =
View.MeasureSpec.getSize(heightMeasureSpec);
        width =
View.MeasureSpec.getSize(widthMeasureSpec);
        eltW = (width - LINE_THICK) / 3;
        eltH = (height - LINE_THICK) / 3;
        setMeasuredDimension(width, height);
    }
    @Override
    protected void onDraw(Canvas canvas) {
        drawGrid(canvas);
        drawBoard(canvas);
    }
    @Override
    public boolean onTouchEvent(MotionEvent event) {
        if (!gameEngine.isEnded()  &&
event.getAction() == MotionEvent.ACTION_DOWN) {
            int x = (int) (event.getX() / eltW);
            int y = (int) (event.getY() / eltH);
            char win = gameEngine.play(x, y);
            invalidate();
            if (win != ' ') {
                activity.gameEnded(win);
            } else {
                // computer plays ...
                win = gameEngine.computer();
                invalidate();
                if (win != ' ') {
                    activity.gameEnded(win);
                }
            }
        }
        return super.onTouchEvent(event);
    }
    private void drawBoard(Canvas canvas) {
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                drawElt(canvas, gameEngine.elt(i,
j), i, j);
            }
        }
    }
    private void drawGrid(Canvas canvas) {
        for (int i = 0; i < 2; i++) {
            // vertical lines
            float left = eltW * (i + 1);
            float right = left + LINE_THICK;
            float top = 0;
            float bottom = height;
            canvas.drawRect(left, top, right,
bottom, gridPaint);
            // horizontal lines
            float left2 = 0;
            float right2 = width;
            float top2 = eltH * (i + 1);
            float bottom2 = top2 + LINE_THICK;
            canvas.drawRect(left2, top2, right2,
bottom2, gridPaint);
        }
    }
    private void drawElt(Canvas canvas, char c, int
x, int y) {
        if (c == 'O') {
            float cx = (eltW * x) + eltW / 2;
            float cy = (eltH * y) + eltH / 2;
            canvas.drawCircle(cx, cy, Math.min(eltW,
eltH) / 2 - ELT_MARGIN * 2, oPaint);
        } else if (c == 'X') {
            float startX = (eltW * x) + ELT_MARGIN;
```

```
                  float startY = (eltH * y) + ELT_MARGIN;
                  float endX = startX + eltW - ELT_MARGIN
* 2;
                  float endY = startY + eltH - ELT_MARGIN;
                  canvas.drawLine(startX, startY, endX,
endY, xPaint);
                  float startX2 = (eltW * (x + 1)) -
ELT_MARGIN;
                  float startY2 = (eltH * y) + ELT_MARGIN;
                  float endX2 = startX2 - eltW +
ELT_MARGIN * 2;
                  float endY2 = startY2 + eltH -
ELT_MARGIN;
                  canvas.drawLine(startX2, startY2, endX2,
endY2, xPaint);
            }
        }
}
```

Creating the UI for our Game

The biggest part of the user interface of our Tic-Tac-Toe game is managed in the *BoardView* class. So, we just need to set our *BoardView* component into a *RelativeLayout* parent *View* in our layout file :

```
<?xml
version="1.0"
encoding="utf-
8"?>
              <RelativeLayout
              xmlns:android="http://schemas.android.com/apk/res/android"
                  xmlns:app="http://schemas.android.com/apk/res-auto"
                  xmlns:tools="http://schemas.android.com/tools"
                  android:layout_width="match_parent"
                  android:layout_height="match_parent"
                  tools:context="com.ssaurel.tictactoe.MainActivity">
                  <com.ssaurel.tictactoe.BoardView
                      android:id="@+id/board"
                      android:layout_width="match_parent"
                      android:layout_height="match_parent"
                      android:layout_centerInParent="true"/>
              </RelativeLayout>
```

Starting a new Game

To star a new game, the user will have to click on a load item in the action bar of our application. So, we add the item in a main.xml menu file under /res/menu :

```
<?xml
version="1.0"
encoding="utf-
8"?>
              <menu
              xmlns:android="http://schemas.android.com/apk/res/android"
                  xmlns:app="http://schemas.android.com/apk/res-auto">
                  <item
                      android:id="@+id/action_new_game"
                      android:orderInCategory="50"
                      android:title="New Game"
                      app:showAsAction="always" />
              </menu>
```

Assemble all the pieces of the puzzle

Last step is to assemble all the components created previously in the *MainActivity* class. In the *onCreate* method, we create the *Board* object and then we pass it in parameter of the *BoardView* got from the main layout of the application. Then, we connect the new game item of the action bar with the *newGame* method of the *Board* object to create a new game when the user will click on it. Finally, we define the *gameEnded* method which was called in the *BoardView* object.

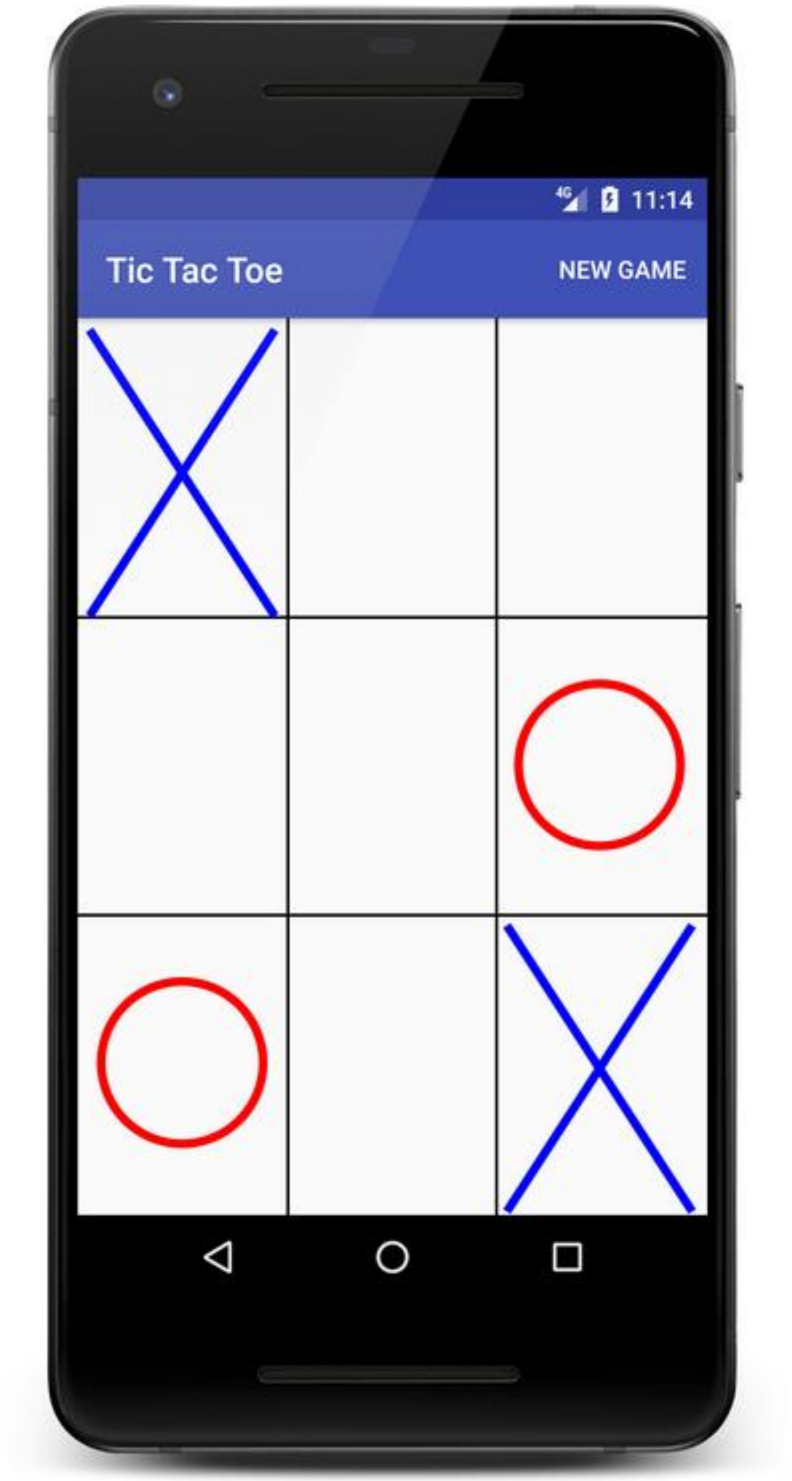
This gives us the following code for our *MainActivity* :

```
package
com.ssaurel.tictactoe;
                      import android.content.DialogInterface;
                      import android.os.Bundle;
                      import android.support.v7.app.AlertDialog;
                      import android.support.v7.app.AppCompatActivity;
                      import android.view.Menu;
                      import android.view.MenuItem;
                      import static com.ssaurel.tictactoe.R.id.board;
                      public class MainActivity extends AppCompatActivity
                      {
                          private BoardView boardView;
                          private GameEngine gameEngine;
                          @Override
                          protected void onCreate(Bundle
                      savedInstanceState) {
                              super.onCreate(savedInstanceState);
                              setContentView(R.layout.activity_main);
                              boardView = (BoardView) findViewById(board);
                              gameEngine = new GameEngine();
                              boardView.setGameEngine(gameEngine);
                              boardView.setMainActivity(this);
```

```
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.main,
menu);
        return super.onCreateOptionsMenu(menu);
    }
    @Override
    public boolean onOptionsItemSelected(MenuItem
item) {
        if (item.getItemId() ==
R.id.action_new_game) {
            newGame();
        }
        return super.onOptionsItemSelected(item);
    }
    public void gameEnded(char c) {
        String msg = (c == 'T') ? "Game Ended. Tie"
: "GameEnded. " + c + " win";
        new AlertDialog.Builder(this).setTitle("Tic
Tac Toe").
                setMessage(msg).
                setOnDismissListener(new
DialogInterface.OnDismissListener() {
                    @Override
                    public void
onDismiss(DialogInterface dialogInterface) {
                        newGame();
                    }
                }).show();
    }
    private void newGame() {
        gameEngine.newGame();
        boardView.invalidate();
    }
}
```

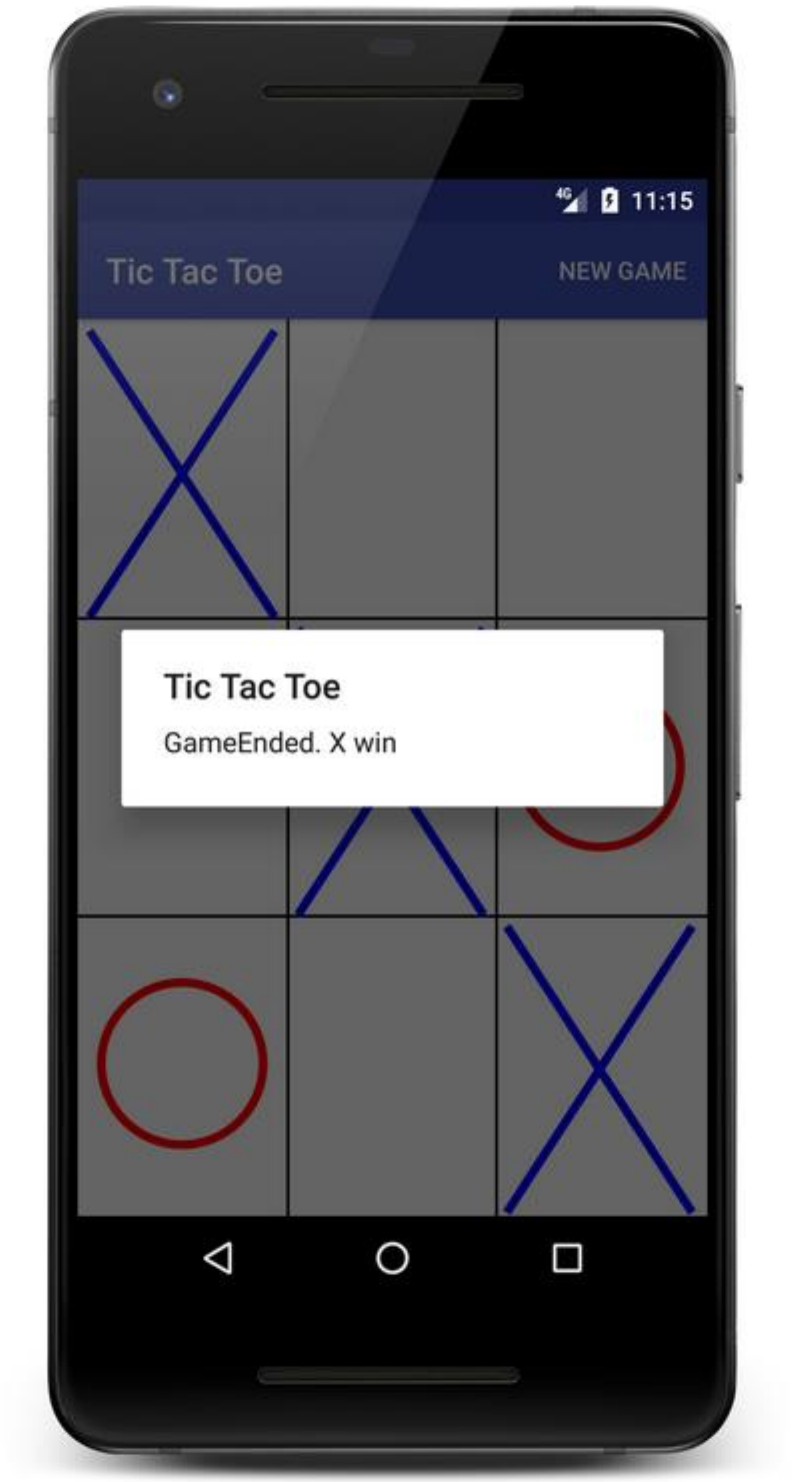
Playing to Tic-Tac-Toe Game

Now, we can try our game and play to the famous Tic-Tac-Toe game :



Playing to the Tic-Tac-Toe Game

The game works great and finally, we win the game against the computer which is logical because our Artificial Intelligence (AI) is really basic :



Tic Tac Toe

GameEnded. X win