```java
package com.tavor.project;

import java.awt.Dimension;
import java.awt.EventQueue;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseEvent;
import javax.swing.*;
import org.apache.log4j.Logger;
import org.apache.log4j.PropertyConfigurator;

@SuppressWarnings({ "serial", "rawtypes" })
public class View extends JFrame {
//class fields
    private JButton jButtonConvert;
    private JButton jButtonClear;
    private JComboBox jComboBoxFrom;
    private JComboBox jComboBoxTo;
    private JLabel jLabelEnterAmount;
    private JLabel jLabelFrom;
    private JLabel jLabelTo;
    private JLabel jLabelDate;
    private JLabel jLabelResult;
    private JScrollPane jScrollPane1;
    private static JTable jTableRates;
    private JTextField jTextFieldInput;
    private JTextField jTextFieldDate;
    private JTextField jTextFieldResult;
    private static Model model=new Model(); //logic module
    public static Logger myLogger = Logger.getLogger(View.class);//log file

    //constructor
    public View() {
        initComponents();
    }

    @SuppressWarnings("unchecked")
    private void initComponents() {             //creating gui components
        jTextFieldInput = new JTextField();
        jLabelEnterAmount = new JLabel();
        jComboBoxFrom = new JComboBox();
        jLabelFrom = new JLabel();
        jComboBoxTo = new JComboBox();
        jLabelTo = new JLabel();
        jScrollPane1 = new JScrollPane();
        jTableRates = new JTable();
        jButtonConvert = new JButton();
        jTextFieldDate = new JTextField();
        jTextFieldResult = new JTextField();
        jLabelDate = new JLabel();
        jLabelResult = new JLabel();
        jButtonClear=new JButton();

        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        setSize(new Dimension(500,500));
        setLocationRelativeTo(null);
        setResizable(false);
        setTitle("Currency Exchange Rate");
        jTextFieldInput.setText("Enter here");
        requestFocusInWindow(true);
        setVisible(true);
        jLabelEnterAmount.setText("Enter Amount");
        requestFocusInWindow() ;

        jComboBoxFrom.setModel(new DefaultComboBoxModel(new String[] { //set From button options
                "ILS-Israel", "AUD-Australia", "CAD-Canada", "CHF-Switzerland",
                "DKK-Denmark", "EGP-Egypt", "EUR-EMU", "GBP-Great Britain",
                "JOD-Jordan", "JPY-Japan", "LBP-Lebanon", "NOK-Norway",
                "SEK-Sweden", "USD-USA", "ZAR-South Africa" }));

        jLabelFrom.setText("From");

        jComboBoxTo.setModel(new DefaultComboBoxModel(new String[] { //set To button options
                "ILS-Israel", "AUD-Australia", "CAD-Canada", "CHF-Switzerland",
                "DKK-Denmark", "EGP-Egypt", "EUR-EMU", "GBP-Great Britain",
                "JOD-Jordan", "JPY-Japan", "LBP-Lebanon", "NOK-Norway",
                "SEK-Sweden", "USD-USA", "ZAR-South Africa" }));

        jLabelTo.setText("To");
        jLabelDate.setText("Date");
        jLabelResult.setText("Result");

        // table- Currencies Rates
        jTableRates.setModel(new javax.swing.table.DefaultTableModel(
                new Object[][] { { null, null }, { null, null },
                        { null, null }, { null, null }, { null, null },
                        { null, null }, { null, null }, { null, null },
                        { null, null }, { null, null }, { null, null },
                        { null, null }, { null, null }, { null, null } },
                new String[] { "Currency", "In Shekel" }));
        jTableRates.setShowVerticalLines(true);
        jScrollPane1.setViewportView(jTableRates);
        jButtonConvert.setBackground(new java.awt.Color(255, 102, 0));
        jButtonConvert.setText("Convert");
        jButtonClear.setBackground(java.awt.Color.CYAN);
        jButtonClear.setText("Clear");

//add all the component into the container
GroupLayout layout = new GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
    .addGap(24, 24, 24)
    .addGroup(layout.createParallelGroup(GroupLayout.Alignment.LEADING)
    .addComponent(jTextFieldInput, GroupLayout.PREFERRED_SIZE, 121, GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabelEnterAmount))
    .addGap(76, 76, 76)
    .addGroup(layout.createParallelGroup(GroupLayout.Alignment.TRAILING)
    .addGroup(layout.createSequentialGroup()
    .addComponent(jLabelTo)
    .addContainerGap(306, Short.MAX_VALUE))
    .addGroup(GroupLayout.Alignment.LEADING, layout.createSequentialGroup()
    .addGroup(layout.createParallelGroup(GroupLayout.Alignment.TRAILING, false)
    .addComponent(jComboBoxTo, 0, 176, Short.MAX_VALUE)
    .addComponent(jLabelFrom, GroupLayout.Alignment.LEADING)
    .addComponent(jComboBoxFrom, 0, GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    .addGap(0, 0, Short.MAX_VALUE))))
    .addGroup(layout.createSequentialGroup()
    .addGroup(layout.createParallelGroup(GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
    .addContainerGap()
    .addComponent(jScrollPane1, GroupLayout.PREFERRED_SIZE, 179, GroupLayout.PREFERRED_SIZE)
    .addGap(67, 67, 67)
    .addGroup(layout.createParallelGroup(GroupLayout.Alignment.LEADING, false)
    .addComponent(jLabelResult)
    .addComponent(jButtonConvert, GroupLayout.PREFERRED_SIZE, 105, GroupLayout.PREFERRED_SIZE)
    .addComponent(jButtonClear, GroupLayout.PREFERRED_SIZE, 105, GroupLayout.PREFERRED_SIZE)
    .addComponent(jTextFieldResult, GroupLayout.DEFAULT_SIZE, 194, Short.MAX_VALUE)
    .addComponent(jLabelDate, GroupLayout.PREFERRED_SIZE, 70,
```

```java
GroupLayout.PREFERRED_SIZE)
    .addComponent(jTextFieldDate)))
    .addGroup(layout.createSequentialGroup()
    .addGap(177, 177, 177)))
    .addContainerGap(GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);
layout.setVerticalGroup(
    layout.createParallelGroup(GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
    .addContainerGap()
    .addGap(8, 8, 8)
    .addGroup(layout.createParallelGroup(GroupLayout.Alignment.BASELINE)
    .addComponent(jLabelEnterAmount)
    .addComponent(jLabelFrom))
    .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)
    .addGroup(layout.createParallelGroup(GroupLayout.Alignment.BASELINE)
    .addComponent(jTextFieldInput, GroupLayout.PREFERRED_SIZE,
GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
    .addComponent(jComboBoxFrom, GroupLayout.PREFERRED_SIZE,
GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(LayoutStyle.ComponentPlacement.UNRELATED)
    .addComponent(jLabelTo)
    .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(jComboBoxTo, GroupLayout.PREFERRED_SIZE,
GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
    .addGroup(layout.createParallelGroup(GroupLayout.Alignment.TRAILING)
    .addGroup(layout.createSequentialGroup()
    .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED, 18,
Short.MAX_VALUE)
    .addComponent(jScrollPane1, GroupLayout.PREFERRED_SIZE, 253,
GroupLayout.PREFERRED_SIZE))
    .addGroup(layout.createSequentialGroup()
    .addGap(26, 26, 26)
    .addComponent(jButtonConvert, GroupLayout.PREFERRED_SIZE, 67,
GroupLayout.PREFERRED_SIZE)
    .addComponent(jButtonClear, GroupLayout.PREFERRED_SIZE, 67,
GroupLayout.PREFERRED_SIZE)
    .addGap(18, 18, 18)
    .addComponent(jLabelResult)
    .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(jTextFieldResult, GroupLayout.PREFERRED_SIZE,
GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
    .addPreferredGap(LayoutStyle.ComponentPlacement.UNRELATED)
    .addComponent(jLabelDate)
    .addPreferredGap(LayoutStyle.ComponentPlacement.UNRELATED)
    .addComponent(jTextFieldDate, GroupLayout.PREFERRED_SIZE,
GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
```

```
    .addGap(0, 0, Short.MAX_VALUE)))
    .addGap(21, 21, 21))
        );
pack();

// Listeners

        jTextFieldInput.addMouseListener(new java.awt.event.MouseListener() { //jTextFieldInput listener
            @Override
            public void mouseReleased(MouseEvent e) {
                // TODO Auto-generated method stub
            }

            @Override
            public void mousePressed(MouseEvent e) {
                // TODO Auto-generated method stub
            }

            @Override
            public void mouseExited(MouseEvent e) {
                // TODO Auto-generated method stub
            }

            @Override
            public void mouseEntered(MouseEvent e) {
                // TODO Auto-generated method stub
            }

            @Override
            public void mouseClicked(MouseEvent e) {    //delete massage on click jTextFieldInput
                if(jTextFieldInput.getText().equals("Enter here"))
                {jTextFieldInput.setText("");}
            }

        });


        jButtonClear.addActionListener(new ActionListener() {   //jButtonClear listener

            public void actionPerformed(ActionEvent e) {        //clear fields
                jComboBoxFrom.setSelectedItem("ILS-Israel");
                jComboBoxTo.setSelectedItem("ILS-Israel");
                jTextFieldInput.setText("Enter here");
                jTextFieldDate.setText("");
```

```java
                jTextFieldResult.setText("");
                myLogger.info("the user pressed on clear");
            }
        });

        jButtonConvert.addActionListener(new ActionListener() { //jButtonConvert listener

            public void actionPerformed(ActionEvent arg0) {

                String amount = jTextFieldInput.getText();  //get amount input
                String from = jComboBoxFrom.getSelectedItem().toString();
                String to = jComboBoxTo.getSelectedItem().toString();

                from = from.substring(0, 3); //get country code
                to = to.substring(0, 3);    //get country code
                try {                       //Convert the data using model
                    double a = 0;
                    a = Double.parseDouble(amount);
                    jTextFieldResult.setText(Double.toString(model.Convert(from, a, to)));
                    jTextFieldDate.setText(model.getDate());
                    myLogger.info("The user convert " + amount + " " + from+ " to " + to + ".");

                } catch (NumberFormatException e) { //exception cause by illegal input
                    myLogger.error("illegal input");
                    JOptionPane.showMessageDialog(new JFrame(),
                            "Please insert numbers only!");
                } catch (NullPointerException e) {  //exception NullPointerException
                    myLogger.error("Error NullPointerException");

                }
            }
        });
    }// end of initComponents


    //update the table, using model module
    public static void fillTableData() {
        String data[]=model.getData();
        for (int i = 0,j=0; i < data.length; i++,++j) {
            jTableRates.setValueAt(data[i], j, 0);
            jTableRates.setValueAt(data[++i], j, 1);
        }
        myLogger.info("table update!");

    }


    //run application
    private static void Start() {
        PropertyConfigurator.configure(".\\log4j.properties");  //set log path properties
        myLogger.info("application runing");
        try {       //set gui properties

            for (UIManager.LookAndFeelInfo info : UIManager
                    .getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }

        } catch (ClassNotFoundException ex) {
            myLogger.error("gui fail",ex);
        } catch (InstantiationException ex) {
            myLogger.error("gui fail",ex);

        } catch (IllegalAccessException ex) {
            myLogger.error("gui fail",ex);

        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            myLogger.error("gui fail",ex);

        }


        //thread thats run the application
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                new View();     //run gui
                fillTableData(); //update table 1st time

            }
        });

        //separated thread thats update the table every x milliseconds
        Thread t = new Thread(new Runnable() {
            public void run() {
                while (true) {
```

```java
                    try {
                        Thread.sleep(180000);
                        fillTableData();
                    } catch (InterruptedException e) {
                        myLogger.error("Error table",e);

                    }
                }
            }
        });
        t.start();
    }

    public static void main(String args[]) {
        Start();
    }
}

```