```
import java.awt.*;
import java.awt.event.*;
import java.time.LocalDateTime;
import java.util.UUID;
class Transaction {
    private String transactionId;
    private double amount;
    private String cardNumber;
    private LocalDateTime timestamp;
    private boolean suspicious;
    public Transaction(double amount, String cardNumber) {
        this.transactionId = UUID.randomUUID().toString();
        this.amount = amount;
        this.cardNumber = cardNumber;
        this.timestamp = LocalDateTime.now();
        this.suspicious = isCardNumberInvalid(cardNumber) || amount > 50000;
    }
    private boolean isCardNumberInvalid(String cardNumber) {
        return cardNumber.length() != 16 || !cardNumber.matches("\\d{16}");
    }
    public String getTransactionId() {
        return transactionId;
    }
    public double getAmount() {
        return amount;
    }
   public String getCardNumber() {
        return cardNumber;
    }
    public LocalDateTime getTimestamp() {
        return timestamp;
    }
    public boolean isSuspicious() {
        return suspicious;
    }
    @Override
    public String toString() {
        return "Transaction ID: " + transactionId +
                "\nAmount: " + amount +
                "\nCard Number: " + cardNumber +
                "\nTimestamp: " + timestamp +
                "\nSuspicious: " + suspicious;
    }
}
public class Main extends Frame {
    private TextField amountField;
    private TextField cardNumberField;
    private TextArea resultArea;
    public Main() {
        setLayout(new FlowLayout());
        Label amountLabel = new Label("Amount:");
        amountField = new TextField(20);
        Label cardNumberLabel = new Label("Card Number:");
        cardNumberField = new TextField(20);
        Button processButton = new Button("Process Transaction");
        processButton.addActionListener(new ProcessTransactionHandler());
        resultArea = new TextArea(10, 50);
        resultArea.setEditable(false);
        add(amountLabel);
        add(amountField);
        add(cardNumberLabel);
        add(cardNumberField);
        add(processButton);
        add(resultArea);
        setTitle("Transaction Processor");
        setSize(600, 400);
        setVisible(true);
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
    }
    class ProcessTransactionHandler implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            try {
                double amount = Double.parseDouble(amountField.getText());
                String cardNumber = cardNumberField.getText();
                if (cardNumber.isEmpty()) {
                    resultArea.setText("Card number cannot be empty!");
                    return;
                }
                Transaction transaction = new Transaction(amount, cardNumber);
                resultArea.setText(transaction.toString());
                if (transaction.isSuspicious()) {
                    resultArea.append("\n\nWarning: Transaction is marked as suspicious!");
                } else {
                    resultArea.append("\n\nTransaction is safe!");
                }
                // Clear the input fields after processing
                amountField.setText("");
                cardNumberField.setText("");
            } catch (NumberFormatException ex) {
                resultArea.setText("Invalid amount. Please enter a valid number.");
            }
        }
    }

    public static void main(String[] args) {
        new Main();
    }
}
```