

**THE FOLLOWING ARE JAVACODES TO SIMULATE AUTOMATED  
MACHINE TELLER (ATM) FUNCTIONALITIES**

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class ATMGUI extends JFrame {

    private JTextField passwordField;
    private JTextArea transactionArea;
    private double accountBalance = 600000.0; // Initial account balance

    public ATMGUI() {
        super("ATM Simulator");
        setSize(500, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        createGUI();
    }

    private void createGUI() {
        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(4, 2));

        // Language Selection
        JComboBox<String> languageComboBox = new JComboBox<>(new
String[]{"English", "Kiswahili"});
        panel.add(new JLabel("Select Language:"));
        panel.add(languageComboBox);

        // Password Entry
        passwordField = new JPasswordField();
        panel.add(new JLabel("Enter Password:"));
        panel.add(passwordField);

        // Buttons
        JButton withdrawButton = new JButton("Withdraw");
        JButton depositButton = new JButton("Deposit");
        JButton balanceButton = new JButton("Check Balance");

        panel.add(withdrawButton);
        panel.add(depositButton);
```

```

panel.add(balanceButton);

// Transaction Area
transactionArea = new JTextArea();
transactionArea.setEditable(false);
JScrollPane scrollPane = new JScrollPane(transactionArea);

add(panel, BorderLayout.NORTH);
add(scrollPane, BorderLayout.CENTER);

// Thank You Label
JLabel thankYouLabel = new JLabel("Thank you for using GSU CEO
ATM!");
add(thankYouLabel, BorderLayout.SOUTH);

// Event Listeners
withdrawButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        promptAmountAndPerformTransaction("Withdraw");
    }
});

depositButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        promptAmountAndPerformTransaction("Deposit");
    }
});

balanceButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        performTransaction("Check Balance", 0); // No amount needed for
balance check
    }
});
}

private void promptAmountAndPerformTransaction(String
transactionType) {
    String amountStr = JOptionPane.showInputDialog(this, "Enter amount:");

    if (amountStr != null && !amountStr.isEmpty()) {

```

```

        try {
            double amount = Double.parseDouble(amountStr);
            performTransaction(transactionType, amount);
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(this, "Invalid amount. Please
enter a valid number.");
        }
    }
}

private void performTransaction(String transactionType, double amount) {
    String language = ((JComboBox<?>) ((Container)
getContentPane().getComponent(0)).getComponent(1)).getSelectedItem().toString
();

    String password = passwordField.getText();

    // Perform transaction logic here
    if ("Deposit".equals(transactionType)) {
        // Simulate deposit
        accountBalance += amount;
        displayTransactionInfo(language, password, transactionType,
amount);

    } else if ("Withdraw".equals(transactionType)) {
        // Simulate withdrawal
        if (accountBalance >= amount) {
            accountBalance -= amount;
            displayTransactionInfo(language, password, transactionType,
amount);
        } else {
            transactionArea.append("Insufficient funds for withdrawal.\n\n");
        }

    } else if ("Check Balance".equals(transactionType)) {
        // Display balance
        displayTransactionInfo(language, password, transactionType, 0);
    }
}

private void displayTransactionInfo(String language, String password,
String transactionType, double amount) {
    transactionArea.append("Language: " + language + "\n");
    transactionArea.append("Password: " + password + "\n");
    transactionArea.append("Transaction Type: " + transactionType + "\n");
}

```

```

        if (!"Check Balance".equals(transactionType)) {
            transactionArea.append("Amount: $" + amount + "\n");
        }
        transactionArea.append("New Balance: $" + accountBalance + "\n\n");
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            @Override
            public void run() {
                new ATMGUI().setVisible(true);
            }
        });
    }
}

```