

---

**Certified LabVIEW Developer Examination**

---

Examinee \_\_\_\_\_ Date: \_\_\_\_\_

Administrator \_\_\_\_\_ Date: \_\_\_\_\_

**Instructions:**

If you did not receive this exam in a sealed envelope stamped “NI Certification,” **DO NOT ACCEPT** this exam. Return it to the proctor immediately. You will be provided with a replacement exam.

- **Please do not detach the binding staple of any section. If any part of the exam paper is missing or detached when returned to National Instruments, you will be deemed to have failed the exam.**
- This examination may not be taken from the examination area or reproduced in any way. You may not keep any portion of this exam after you have completed it.
- Please do not ask the proctor for help. If you believe the intent of any part of the exam is not clear, you may make appropriate assumptions. Please document your assumptions either on the question paper or on the LabVIEW block diagram.
- The exam requires you to develop a LabVIEW application based on a set of specifications.
- A computer with a standard installation of LabVIEW is the only reference allowed for the examination. Externally developed code and third party tools are not allowed in the exam.
- You may use LabVIEW design patterns, templates, and examples available in the development environment as a guide/resource for the application development.
- The application must be specifically developed for the exam submission.
- Submit your finished application on the disk provided.
- Total time allocated for the exam: 4 hours
- Exam passing grade: 75%

**Grading:**

The application development exam consists of a total of 40 points which are allocated as follows:

- Programming style (**15 points**)
- Functionality (**15 points**)
- Documentation (**10 points**)

**IMPORTANT:**

- **When you have completed the exam, place the exam document, the disk with the saved application, and any deliverables in the envelope provided.**
- **Please SEAL the envelope.**
- **Give the sealed envelope to your proctor.**

**Application Development**  
**Section I: General Requirements**

The Certified LabVIEW Developer exam tests your ability to develop a LabVIEW application.

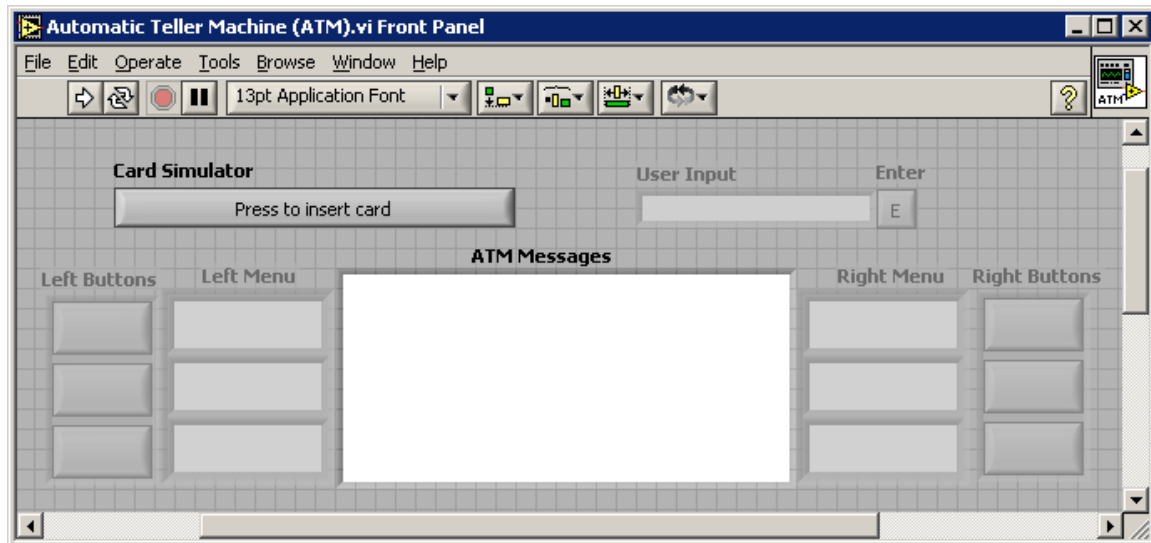
The application should do the following:

- Function as specified in Section II of this document.
- Conform to LabVIEW coding style and documentation standards (found in LabVIEW documentation – *LabVIEW Development Guidelines*).
- Be created expressly for this exam using VIs and functions available in LabVIEW. Templates, examples, or code developed outside the bounds of this exam are not acceptable for use in the application.
- Be hierarchical in nature. All major functions should be performed in subVIs.
- Use a state machine that either uses a type defined enumerated control, queue, or Event structure for state management.
- Be easily scalable to add more states / features without having to manually update the hierarchy.
- Minimize the use of excessive structures, variables (locals / globals) and property nodes.
- Respond to front panel controls (within 100 ms) and not utilize 100 % of CPU time.
- Close all opened references and handles where used.
- Be well documented and include the following:
  - Labels on appropriate wires within the main VI and subVIs.
  - Descriptions for each algorithm.
  - Documentation in **VI Properties » Documentation** for both main VI and subVIs.
  - Tip strips and descriptions for front panel controls and indicators.
  - Labels for constants

**Application Development**  
**Section II: Application Requirements**  
**Automated Teller Machine (ATM)**

**Task:**

Design an Automated Teller Machine (ATM) controller using LabVIEW. The front panel of the controller should look similar to the following front panel.

**General Operation:**

The ATM controller simulates the control system of an Automated Teller Machine. The user interacts with controls and indicators on the front panel to perform common ATM functions such as deposit funds, withdraw funds, and inquire about the balance of funds from the user account.

The ATM controller has access to user account information stored in the `Accounts.txt` file. You, the developer, are responsible for creating the `Accounts.txt` file, which the ATM controller reads from or writes to, depending on user-initiated transactions. Refer to the *ATM Accounts File specification* section for more information about the initial content and data format of the `Accounts.txt` file.

The controller should perform the following general operations:

- Obtain user input through the **User Input**, **Left Buttons**, and **Right Buttons**.
- Manage the **Left Menu** and **Right Menu** strings and take appropriate action by responding to the corresponding **Left Buttons** and **Right Buttons**.
- Prompt user actions and confirm transactions by displaying appropriate messages in the **ATM Messages** indicator.
- Monitor user inactivity and terminate the user session after the specified time expires.

**Sequence of Operation:**

**Start (Application Run):** When the application starts, the **User Input**, **Enter (E)**, **Left Menu**, **Right Menu**, **Left Buttons**, and **Right Buttons** should be disabled.

The **Card Simulator** button should be enabled and the **ATM Messages** indicator should display the *Welcome Message*.

Refer to the *ATM Message Types* section of the specification for message types and font specifications for the messages.

**Insert ATM card:** Click the **Card Simulator** button. The button should remain pressed and the Boolean text should display **Card Inserted**. This action enables the **Enter (E)** button and the **User Input** string control, focuses the cursor to the **User Input** control and waits for the user to enter the account number in the control. The **Card Simulator** button should remain pressed until the user completes the session or a timeout (10 seconds) occurs.

After the user completes keying in the account number and presses the **Enter (E)** button, the ATM controller should access the ATM Accounts file and verify that the account exists.

If the account does not exist, the **ATM Messages** indicator should display the *Account Verification Failed Message* and prompt the user to re-enter the account number.

If the account exists, the ATM controller should enable the **Left Menu**, **Right Menu**, **Left Buttons** and **Right Buttons**, and display the *Main Menu Message* on the **ATM Messages** indicator. The ATM controller should also populate the **Left Menu** and **Right Menu** with the Main Menu and wait for user action.

Refer to the *ATM Menus* section of the specification for more information about the strings that appear in the menus.

**Note:** If, at anytime after the insertion of the ATM card, the ATM controller does not detect use of front panel controls for 10 seconds, the **Card Simulator** button should be released and the user session should be terminated by stopping the application. The **ATM Messages** indicator should display the *Session Terminate Message*.

**Deposit:** Click the button next to the corresponding **Right Menu** item. This action should display the *Deposit Message*, focus the cursor to the **User Input**, and allow the user to enter the amount to be deposited. After the user enters an amount in the **User Input** and presses the **Enter (E)** button, the *Deposit Complete Message* should display on the **ATM Messages** indicator and the ATM Accounts file should update with the new balance.

**Withdraw:** Click the button next to the corresponding **Right Menu** item. This action should display the *Withdrawal Message*, focus the cursor to the **User Input** control, and allow the user to enter the amount to withdraw from the account. After the user enters an amount in the **User Input** control and presses the **Enter (E)** button, the ATM controller should check if the account has sufficient funds to complete the transaction. If the account has sufficient funds, the controller should deduct the keyed in amount from the current balance, update the ATM Accounts file with the new balance, and display the *Withdrawal Complete Message* on the **ATM Messages** indicator.

If the account has insufficient funds, the *Withdrawal Failed Message* should display on the **ATM Messages** indicator.

**Fast Cash \$50:** Click the button next to the corresponding **Right Menu** item. The ATM controller should check if the account has sufficient funds to complete the transaction. If the account has at least \$50, the ATM controller should deduct \$50 from the current balance, update the ATM Accounts file with the new balance, and display the *Withdrawal Complete Message* on the **ATM Messages** indicator.

If the account has insufficient funds, the *Withdrawal Failed Message* should display on the **ATM Messages** indicator.

**None:** This button is reserved for future use. Clicking the button next to the corresponding **Left Menu** item should not produce any results. The **ATM Messages** indicator should continue to display the current message.

**Balance Inquiry:** Click the button next to the corresponding **Left Menu** item to get the current balance from the ATM Accounts file. The **ATM Messages** indicator should display the *Balance Inquiry Message*.

**Return Card and Terminate:** Click the button next to the corresponding **Left Menu** item to terminate the user session. The **ATM Messages** indicator should display the *Session Terminate Message*. The ATM controller should release the **Card Simulator** button to indicate the return of the ATM card, disable the **User Input** control, and terminate the application.

**Description of Controls/Indicators:**

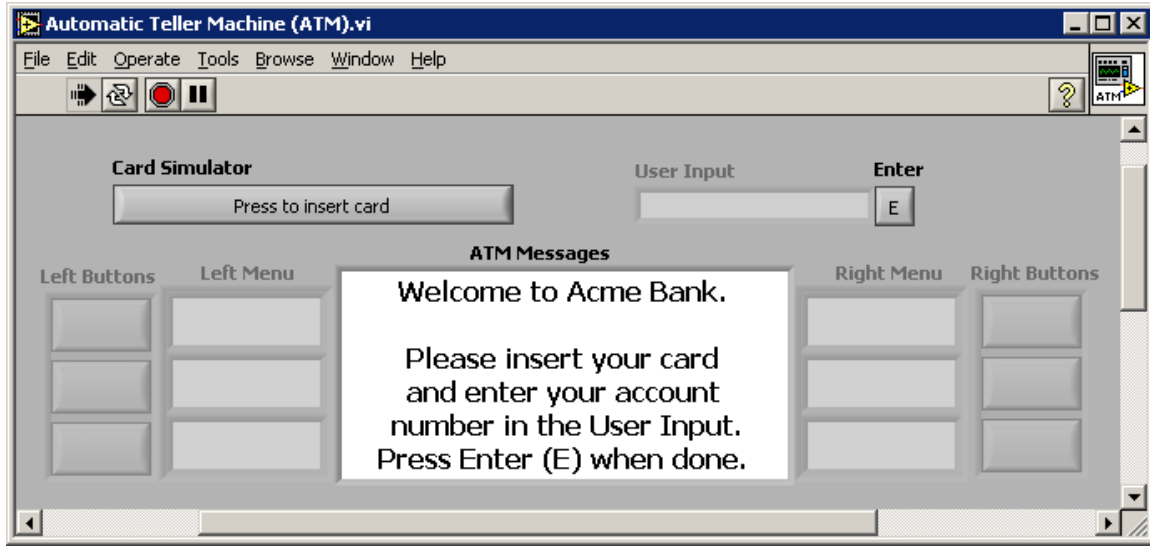
Control Name	Control Description - Function
<b>Card Simulator</b>	Button – Simulates insertion of an ATM card OFF text: Press to inset card ON text: Card inserted
<b>Left Buttons</b>	Cluster of Buttons – Allows user to select ATM transaction
<b>Right Buttons</b>	Cluster of Buttons – Allows user to select ATM transaction
<b>User Input</b>	String – Allows user to enter Account number, Deposit, and Withdrawal amounts.
<b>Enter Button</b>	Button – Commits the entry in the <b>User Input</b>

Indicator Name	Indicator Description - Function
<b>ATM Messages</b>	String – Displays messages to guide and provide feedback to user. Refer to the <i>ATM Message Types</i> section for the list of messages
<b>Left Menu</b>	Cluster of strings – Used to display menu. Refer to the <i>ATM Menus</i> section for the list of menus.
<b>Right Menu</b>	Cluster of strings – Used to display menu. Refer to the <i>ATM Menus</i> section for the list of menus.

**ATM Message Types:**

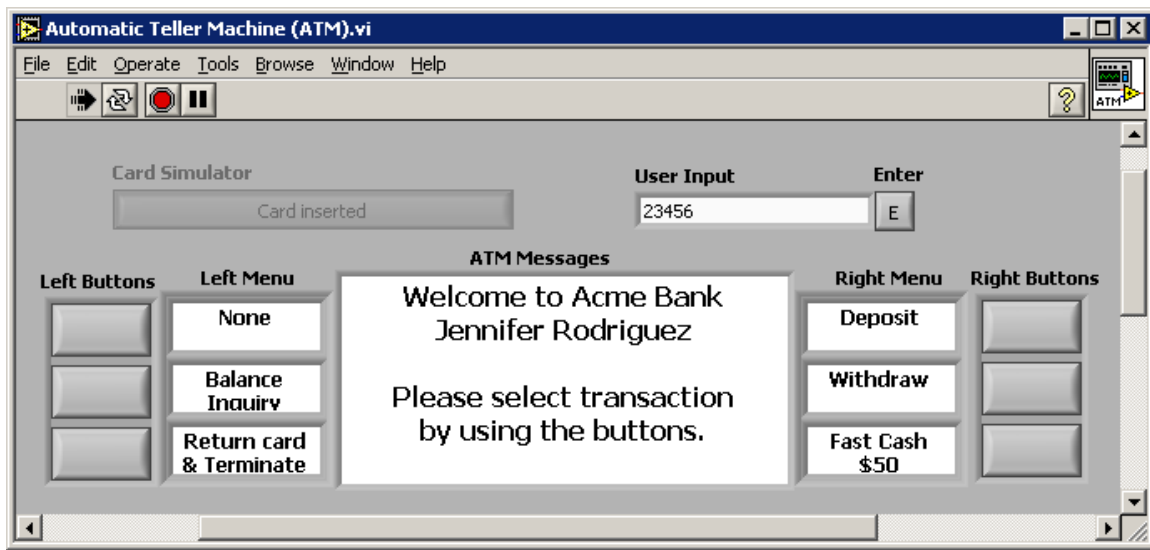
<b>Message Type</b>	<b>Message</b>
Welcome Message	Welcome to Acme Bank. Please insert your card and enter your account number in the User Input.  Press Enter (E) when done.
Main Menu Message	Welcome to Acme Bank First Name, Last Name (from file)  Please select transaction by using the buttons.
Account Verification Failed Message	Account Information Incorrect Please Re-Enter Account Number. Press Enter (E) when done
Session Terminate Message	Your session has been terminated due to inactivity or menu selection. Please take your card  Goodbye!
Deposit Message	Please enter amount to deposit and press Enter (E) when done
Deposit Complete Message	\$ (Amount) Deposited
Withdrawal Message	Please enter amount to withdraw and press Enter (E) when done
Withdrawal Complete Message	\$ (Amount) Withdrawn
Withdrawal Failed Message	Insufficient funds in account Please check your balance and try again
Balance Inquiry Message	Your Balance Is: \$ (Balance Amount from file)

The font for the **ATM Message** indicator should be 14 point and bold. The message should be centered in the display, as shown in the following figure.

**Sample screen shot for ATM messages:****ATM Menus:****Main Menu:**

Left Menu (Strings)	Right Menu (Strings)
None	Deposit
Balance Inquiry	Withdraw
Return Card and Terminate	Fast Cash \$50

**Note:** The application should be scalable to add other menus without major modifications to the application.

**Sample screen shot with Main Menu displayed:**

**ATM Accounts File specification:**

The file should be a text file containing ATM account information in a Comma Separated Value (CSV) format. The file should be located in the same folder as the main VI. The ATM controller should not prompt the user for a filename or prompt the user to update the file.

The account information should be in the following format:  
Account Number (5 digits), First Name, Last Name, Balance

Each account should be identified by a unique account number only.

The following four records should exist to test the operation of the ATM:

12345,John,Doe,550

23456,Jennifer,Rodriguez,1000

34567,Bryan,Smith,750

45678,Julie,Ramirez,900