Certified LabVIEW Architect 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 on the LabVIEW block diagram of the appropriate VI.
- The exam requires you to architect a LabVIEW application based on a set of requirements.
- 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 architecture must be specifically developed for the exam submission.
- Submit your LabVIEW application on the disk provided.
- Total time allocated for the exam: 4 hours
- Exam passing grade: 70%

Grading:

The application architecture development exam consists of a total of 100 points which are allocated as follows:
- User interface and block diagram style   : 10 points
- Documentation   : 20 points
- Requirements coverage   : 30 points
- Architecture development   : 40 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.**

Project Deliverables

For the purposes of this exam, you are required to <u>design</u> an architecture that covers all of the Project Specifications listed below. Your architecture should include implementations of the following key components:
- Project hierarchy
- User interface(s)
- Top-level loops
- Design patterns (state machines, user interface handlers, etc)
- Modules and Application Programmers Interfaces (APIs) for all modules
- Important data structures
- Inter-process communication mechanisms
- Error handling strategy
- Shutdown strategy

Assume that specific algorithms, states, and functions will be implemented by a secondary developer. These do not need to be implemented as part of the test. However, for each requirement which is not implemented, you should include localized instructions or comments in the most appropriate location on the block diagram. These instructions or comments are directed for a developer and describe a method for implementing the requirement.

Requirements Tracking
In order to demonstrate coverage of a requirement, you must include the ID of the requirement in the documentation of your architecture. Each requirement's ID precedes the requirement text in the specification below. You may cover requirements in any part of your architecture's documentation, including:
- VI Documentation Property
- Control Documentation Property
- Project or Library Documentation Property
- Comments on the front panel or block diagram

A single requirement may be covered by multiple sections of code if all of those sections are necessary to fulfill the requirement.

To cover a requirement, include the following text in the documentation of your code:
`[Covers: ID]`

A requirements tracking tool will used to verify your program, therefore, please be careful to use this exact syntax. You should also double-check that each requirement ID is correct.

Example:
Consider the following requirement:

> SD2. The database shall provide a mechanism to search for a record based upon an account number.

You could cover this requirement in your code by including the text `[Covers: SD2]` in the Documentation property of a SubVI which implements the requirement. Alternately, you could place a comment on your block diagram with instructions to a developer to implement

the search. In this case, you would place the text `[Covers: SD2]` somewhere in the comment.

<u>**Project Specifications**</u>
<u>**Automatic Teller Machine Simulator (ATM)**</u>

Overview
An ATM machine is an unmanned, computer-controlled device that performs the functions of a bank teller. The functions include balance inquiries, accepting deposits, and issuing cash for withdrawal requests.

ATM transactions are initiated by a customer inserting a bank card and entering a pin code on a keypad. Upon authentication, the ATM controller allows the customer to initiate transaction functions by pressing the appropriate buttons, and responds accordingly by displaying text messages.

Definitions
Simulation Interface – A program which is run to test ATM algorithms in a software only environment. This program incorporates the Simulated Console, Simulated Database, and ATM Controller.

User Console – The interface by which the user interacts with the ATM Controller. All references to the User Console may apply to the Simulated Console or Physical Console.

Physical Console – A hardware interface which includes the physical buttons and displays that allow a user to interact with the ATM.

Simulated Console – A software interface which mimics a Physical Console.

Account Database – The interface by which the ATM accesses and changes account information and balances. All references to the Account Database may apply to the Enterprise Database or Simulated Database.

Enterprise Database – A centrally accessible database which provides account information to multiple ATMs and bank systems.

Simulated Database – A software interface which mimics access to an Enterprise Database.

ATM Controller – A software component that manages users, accounts, and transactions for the ATM.

Session – A session is initiated when an ATM user inserts a card. It is complete when the card is returned.

Simulation Interface
Due to the unavailability of a physical setup or enterprise data storage your client has requested that you create simulations of external hardware and databases. These simulations will be used to validate the ATM controller logic and algorithms.

The Simulated Console will be replaced with a prototype Physical Console in a subsequent project phase. The Simulated Database will be replaced with network communication to an Enterprise Database in a subsequent project phase.
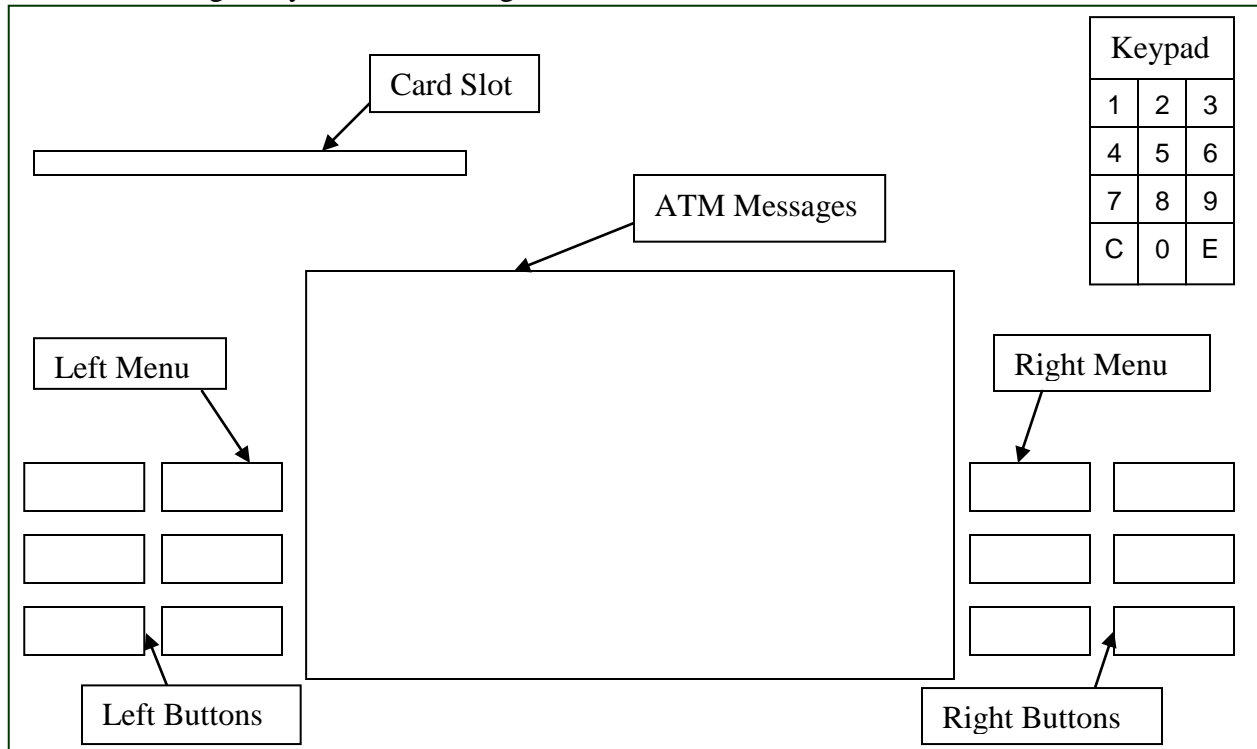
Simulation Interface Requirements
SI1. The components of the Simulation Interface shall be separated from the ATM controller by defined interfaces which mimic communication with the Physical Console and Enterprise Database.

SI2. The Simulation Interface shall automatically initiate the Simulated Console, Simulated Database, and ATM Controller upon startup.

SI3. The Simulation Interface simulates a single session and terminates when the user completes all activity and requests return of the bank card.

Simulated Console Requirements
Your client has given you the following illustration of the ATM User Console.

UI1. The simulation interface shall mimic the illustration as closely as possible
The simulation interface shall include the following components and functionality

UI2. The simulation interface shall include a **Card Slot** component.

> UI2A. Card slot component shall simulate acceptance of a card from a user.

> UI2B. When the Card Slot component is enabled, inserting a simulated card shall notify the ATM Controller.

> UI2C. Card slot shall visibly change to indicate that a card is inserted.

UI3. The simulation interface shall include a **Keypad** component

> UI3A. When enabled, the keypad component shall allow the user to enter a number.

>> UI3A1. Pressing one of the numeric keys on the keypad will cause the number to be displayed on an indicator. The number should be appended to any existing numbers.

> UI3B. Pressing the C button on keypad shall clear any digits entered.

> UI3C. The ATM Controller may clear the keypad by signaling the User Console.

> UI3D. Pressing the E button on keypad shall signal to the ATM Controller that numeric entry is complete.

UI4. The simulation interface shall include a **ATM Messages** component

UI4A. ATM messages will display text messages when prompted by the ATM Controller.

UI4B. When a new message is received, any previous messages shall be cleared.

UI5. The simulation interface shall include a **Left and Right Buttons** component.

UI5A. When the buttons are enabled, the user console will notify the ATM Controller each time a button is pressed.

UI6. The simulation interface shall include a **Left and Right Menu** component.

UI6A. The Left and Right Menu items will display text items when prompted by the ATM Controller.

UI7. The ATM Controller shall have the ability to disable the function of any component on the User Console.

UI7A. A disabled component will perform no action if the user presses or interacts with it. Disabling a display component such as the ATM Messages or Left and Right Menu has no effect.

Note For this phase of the project, you need not simulate the mechanism for inserting money during a deposit or dispensing money after a withdrawal.

Simulated Database Requirements

SD1. The database shall store records where each record contains the following elements.

SD1A. Account Number – 5 Digit Integer number

SD1B. First Name – Text, maximum of 256 characters

SD1C. Last Name – Text, maximum of 256 characters

SD1D. Balance – Fractional number with at least 9 integer digits and 2 decimal digits

SD1E. Locked – Boolean flag

Note The above descriptions merely describe the type and limits of each element. You may choose to store and/or implement each element with any data type that meets the requirements.

SD2. The database shall provide a mechanism to search for a record based upon an account number.

SD2A. Upon locating a record the database shall check the locked element of the record.

SD2A1. If the record is locked, the database shall return an appropriate notification. In this scenario the account information is not returned.

SD2A2. If the record is not locked, the database shall change the element to locked and return the record.

SD2B. If a matching record cannot be located, the database shall return an appropriate notification.

SD2C. If the database interface successfully searches for and locks a record, the interface is said to "own" that record.

SD2C1. The database interface should keep a temporary copy of the data in a record it owns.

SD2C2. The database interface should own only one record at a time.

SD2C2A. Attempting to own a second record should return a Session Error.

SD3. The database shall provide a mechanism to modify the account balance of a record.

SD3A. This functionality will only be available on a record that is owned by the interface.

SD3B. Modifying the account balance in this way will only affect the local copy of the owned record.

SD4. The database shall provide a mechanism to access any element of a record.

SD4A. This functionality will only be available on a record that is owned by the interface.

SD5. The database shall provide a mechanism to commit a record.

SD5A. This functionality will only be available on a record that is owned by the interface.

SD5B. When a record is committed, the database interface shall overwrite the balance of the original record with the balance of the local record copy.

SD6. The database shall provide a mechanism to unlock a record.

SD6A. This functionality will only be available on a record that is owned by the interface.

SD6B. When a record is unlocked, the database interface is no longer considered to own the record, and should clear any local copies of the record data.

SD7. It must be somehow possible for a tester to populate the simulated database. However, this may be through an external program or editor.

ATM Controller

The ATM Controller is the software component that performs transactions and provides appropriate feedback as requested through the user console. The ATM Controller performs the following main functions:

- Get transaction commands and data from the user console
- Authenticate users and perform requested transactions
- Output messages to the ATM Messages display
- Implement security features

User Message Requirements

UM1. The ATM Controller shall store a list of messages, as shown in the ATM Messages Table below. The Message Name is an identifier that is used throughout this document to efficiently refer to a given message.

UM1A. The message table shall be stored in such a way that messages can be added, removed, or changed without recompiling the ATM Controller code.

UM2A. Some messages contain parameters in *italic* that shall be dynamically replaced with data from the Account database or keypad when displayed.

UM2. The ATM Controller will send a message to the User Console as necessary to fulfill other requirements.

ATM Messages

Message Name	Message
Welcome Message	Welcome to Acme Bank. Please insert your card and enter your pin number on the keypad. Press Enter (E) when done.
Main Menu Message	Welcome to Acme Bank *First Name from DB*, *Last Name from DB* Please select a 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 DB*)
System Error Message	This terminal is out of service, please use another terminal or try again later.

ATM Menu Requirements

AM1. The ATM Controller shall store a list of menu states, as shown in the ATM Menus Table below. The Menu State is an identifier that is used throughout this document to efficiently refer to a given menu.

    AM1A. The menu table shall be stored in such a way that menu items and states can be added, removed, or changed without recompiling the ATM Controller code.

AM2. The ATM Controller will apply a menu state to the User Console as necessary to fulfill other requirements.

    AM2A. Each table item within the menu state corresponds to one of the left or right menu locations on the User Console. Applying a state displays the corresponding text on the corresponding menu location. A menu location with the text "None" displays nothing.

ATM Menus

Left Menu	Right Menu
Menu State: Main	
None	Deposit
Balance Inquiry	Withdraw
Return Card and Terminate	Fast Cash $50
Menu State: Cancel	
None	None
None	None
None	Cancel
Menu State: Return	
None	None
None	None
None	Done
Menu State: Blank	
None	None
None	None
None	None

ATM Button Requirements

AB1. When a requirement specifies that the ATM Controller should wait for a user button the ATM Controller should wait for the User Console to indicate that a button has been pressed.

AB2. Any buttons pressed before the ATM Controller begins waiting should be ignored.

AB3. When pressed, a button indicates the function associated with it in the current menu state.

AB4. Any buttons with a menu state of none should be ignored.

✎ **Note** The operation of the machine is described in states. This is a convenient and descriptive format for showing the order of operations and response to inputs and is not intended to constrain the architecture to any specific design, logic, or states.

Initial State Requirements

IS1. On the user console, disable the **Keypad**, **Left Buttons**, and **Right Buttons**. Enable the **Card Slot**.

IS2. Set the Menu State to Blank.

IS3. Display `Welcome Message`.

IS4. Wait for the User Console to indicate that a card has been inserted into the **Card Slot**.

IS5. Transition to the Login State.

Login State Requirements

LO1. On the user console, disable the **Card Slot**, **Left Buttons**, and **Right Buttons**. Enable the **Keypad**.

LO2. Clear the **Keypad**.

LO3. Set the Menu State to Blank.

LO4. Display `Welcome Message`.

LO5. Wait for the User Console to indicate that a number has been entered into the keypad.

LO6. Query the Account Database for the number

   LO6A. If the Account is not found in the database or is locked, this is considered an invalid login attempt. See SC1.

      LO6A1. In the event of an invalid login, display the `Account Verification Failed Message.`

      LO6A2. In the event of an invalid login, restart the Login state.

   LO6B. If the Account is found and not locked clear the **Keypad** and transition to the Main Menu State.

Main Menu State Requirements

MM1. On the user console, disable the **Card Slot** and **Keypad**. Enable the **Left Buttons**, and **Right Buttons**.

MM2. Display `Main Menu Message`.

MM3. Set the Menu State to Main Menu.

MM4. Wait for a user button press.

   MM4A. Pressing the Balance Inquiry button transitions the system to the Balance Inquiry state.

   MM4B. Pressing the Deposit button transitions the system to the Deposit state.

MM4C. Pressing the Withdraw button transitions the system to the Withdraw state.

MM4D. Pressing the Fast Cash $50 button shall cause the system to check if the balance of the Account Database record is greater or equal to $50.00.

MM4D1. If there are sufficient funds, modify the balance of the Account Database record and commit the change.

MM4D2. If there are insufficient funds, make no change to the database and display the Withdrawal Failed Message in the subsequent Withdrawal Complete state.

MM4D3. Transition to the Withdrawal Complete state after Fast Cash $50 is pressed, whether the transaction was successful or not.

MM4E. Pressing the Return Card and Terminate button transitions the system to the Terminate state.

Balance Inquiry State Requirements

BI1. On the user console, disable the **Card Slot** and **Keypad**. Enable the **Left Buttons** and **Right Buttons**.

BI2. Clear the **Keypad**.

BI3. Display `Balance Inquiry Message.`

BI4. Set the Menu State to Return.

BI5. Wait for a user button press.

BI5A. Pressing the Done button transitions the system to the Main Menu state.

Deposit State Requirements

DP1. On the user console, disable the **Card Slot**. Enable the **Keypad**, **Left Buttons** and **Right Buttons**.

DP2. Display `Deposit Message.`

DP3. Set the Menu State to Cancel.

DP4. Wait for the User Console to indicate a user button press or a keypad entry.

DP4A. Pressing the Cancel button transitions the system to the Main Menu state.

DP4B. A keypad entry triggers a deposit

DP4B1. In the event of a deposit, modify the balance of the Account Database record and commit the change.

DP4B2. In the event of a deposit, transition to the Deposit Complete State

Deposit Complete State Requirements

DC1. On the user console, disable the **Card Slot** and **Keypad**. Enable the **Left Buttons** and **Right Buttons**.

DC2. Clear the **Keypad**.

DC3. Display `Deposit Complete Message.`

DC4. Set the Menu State to Return.

DC5. Wait for a user button press.

DC5A. Pressing the Done button transitions the system to the Main Menu state.

Withdrawal State Requirements

WD1. On the user console, disable the **Card Slot**. Enable the **Keypad**, **Left Buttons** and **Right Buttons**.

WD2. Display `Withdrawal Message.`

WD3. Set the Menu State to Cancel.

WD4. Wait for the User Console to indicate a user button press or a keypad entry.

WD4A. Pressing the Cancel button transitions the system to the Main Menu state.

WD4B. A keypad entry triggers a withdrawal

WD4B1. In the event of a withdrawal, compare the balance of the Account Database record to the withdrawal amount.

WD4B1A. If there are sufficient funds, modify the balance of the Account Database record and commit the change.

WD4B1B. If there are insufficient funds make no change to the database and display the Withdrawal Failed Message in the subsequent Withdrawal Complete state.

WD4B2. In the event of a withdrawal or failed withdrawal, transition to the Withdrawal Complete state.

Withdrawal Complete State Requirements

WC1. On the user console, disable the **Card Slot** and **Keypad**. Enable the **Left Buttons** and **Right Buttons**.

WC2. Display `Withdrawal Complete Message` or `Withdrawal Failed Message` depending upon whether the withdrawal was successful.

WC3. Set the Menu State to Return.

WC4. Wait for a user button press.

WC4A. Pressing the Done button transitions the system to the Main Menu state.

Terminate State Requirements

TS1. Disable all of the components of the User Console.

TS2. Signal the User Console to return any inserted card to the user.

TS3. Set the Menu State to Blank.

TS4. Display `Session Terminate Message.`

Security Requirements

SC1. The machine transitions to the Terminate State after three consecutive invalid login attempts.

SC2. The machine transitions to the Terminate State after an inactivity period of ten seconds during which no input from the console is detected.

Error Handling Requirements

EH1. The system shall define three levels of error with associated actions.

EH1A. A Minor Error shall be invisible to the user. The system should continue any operations without interruption.

EH1B. A Session Error will cause the system to enter the Terminate State. No additional feedback to the user is necessary.

EH1B1. A session error shall cause any records locked by the ATM Controller to be unlocked. No changes should be committed.

EH1C. A System error will lock out the system.

EH1C1. A system error shall cause any records locked by the ATM Controller to be unlocked. No changes should be committed.

EH1C2. A system error shall disable all User Console controls.

EH1C3. A system error shall set the menu state to Blank.

EH1C4. A system error shall display `System Error Message`.

EH2. The error handling system shall maintain a log file which can be checked by ATM maintenance personnel.

EH2A. The error log shall include a description of each error that occurs.

EH2B. The error log shall include the level of the error, as specified by EH1.

EH2C. The error log shall include the time and date at which the error occurred.

EH2D. The error log shall include the account number of any user logged in at the time the error occurred.

Note When an error level is not specifically noted in this document, the system designer shall select the appropriate error level based upon the impact of the error.

The system designer may, at their discretion, ignore errors that are expected behavior or errors that are entirely handled by the system. Such errors do not need to be noted in the error log.