Test Driving a CLD

Anton Sundqvist

ASTEMES
Test and Measurement Solutions

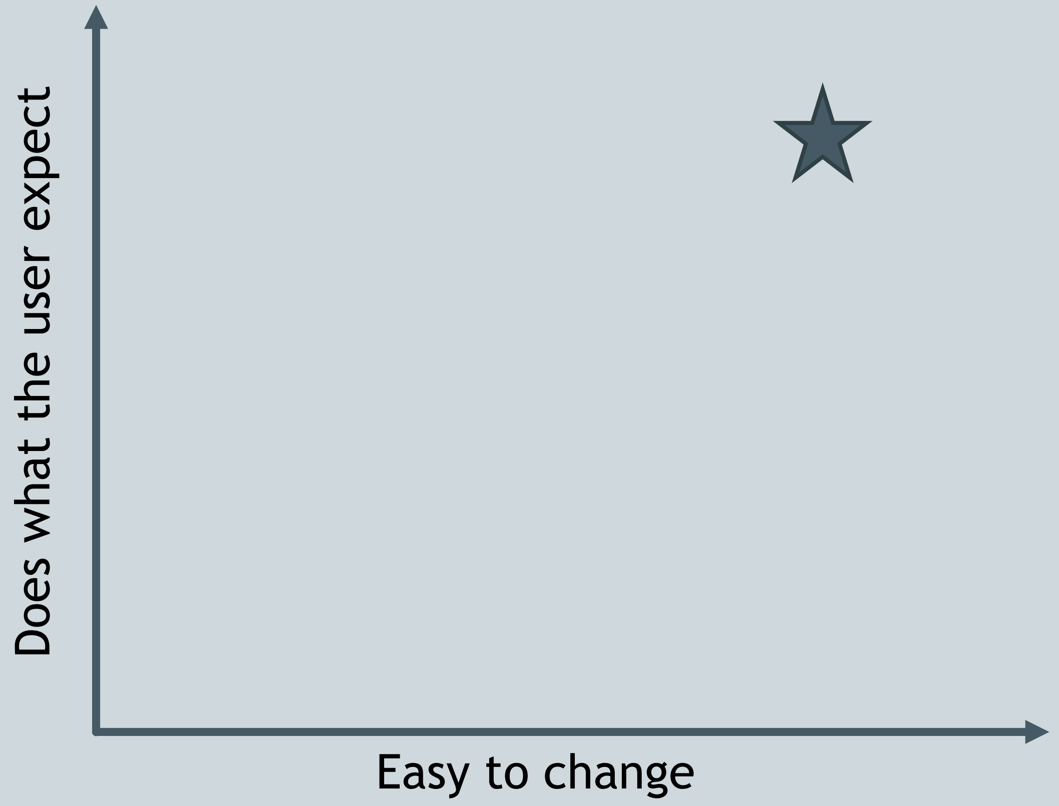
Introduction

- Test Driven Development and Design
- ATM CLD Exam
- Design Process
- Design Analysis
- Conclusions

https://github.com/astemes/astemes-glasummit-2022

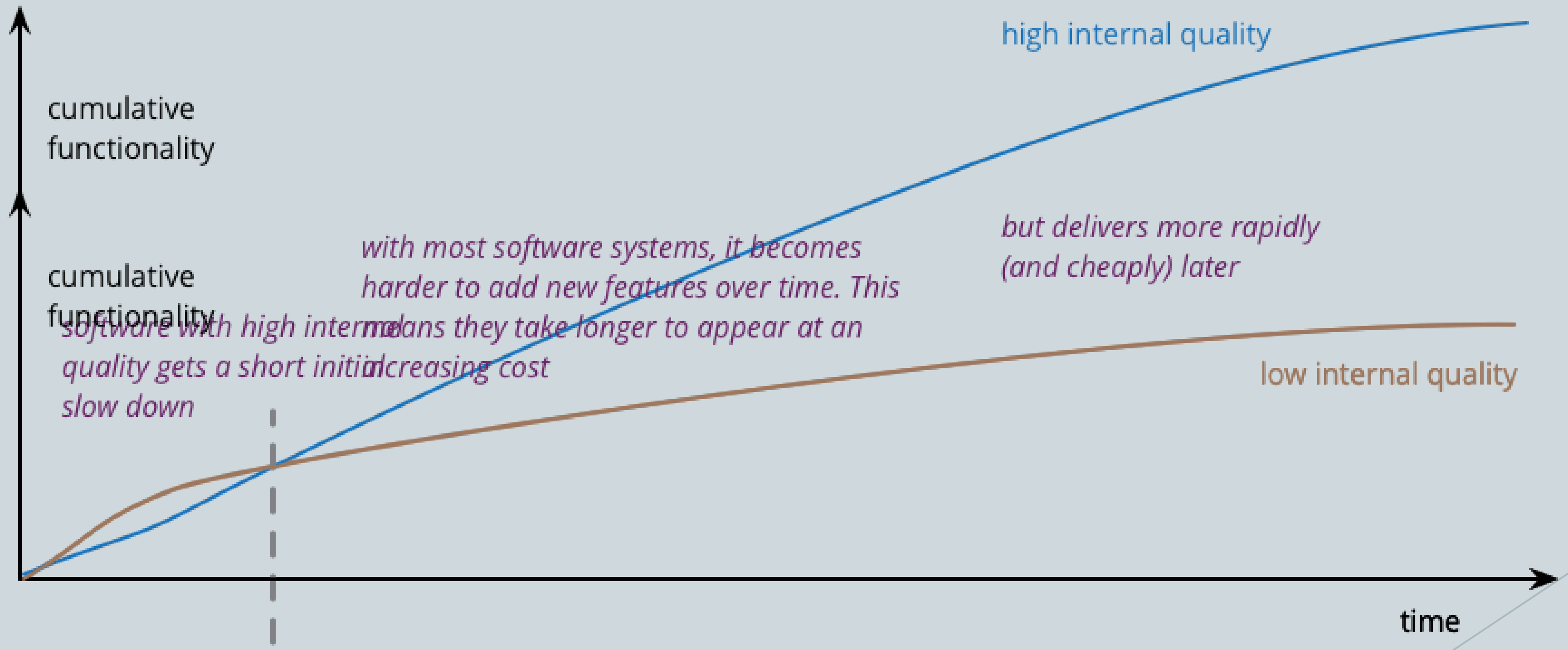
ASTEMES
Test and Measurement Solutions

Software Quality



How to deliver high quality code at a high pace?

- The only way to go fast is to not make a mess while going

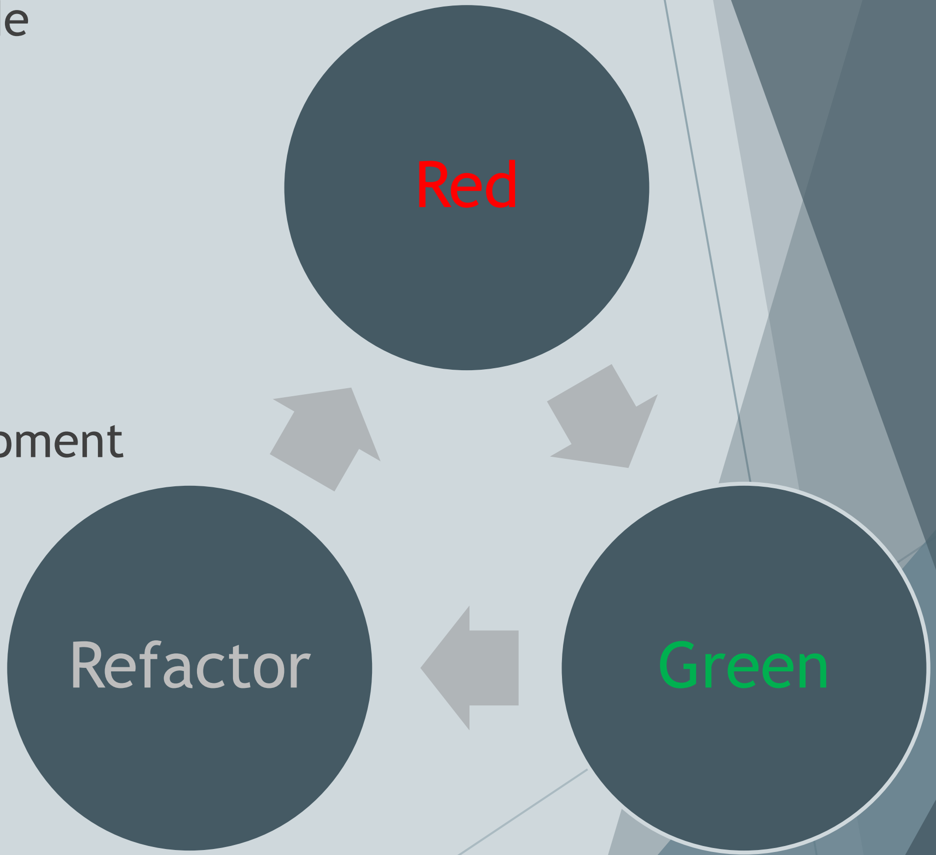


Source: https://martinfowler.com/articles/is-quality-worth-cost.html

ASTEMES Test and Measurement Solutions

Test-Driven Development

- A test is created **before** writing **any** production code
- Code is developed to make the test pass
- Refactor to reduce duplication and clean up
- Red–Green–Refactor **cycle**
- Cycle time on the order of **minutes**
- Writing tests is not a separate activity from development



Why bother with Test Driven Development?

1. Improved Design
2. Self Testing Code
3. Less stress
4. Documentation

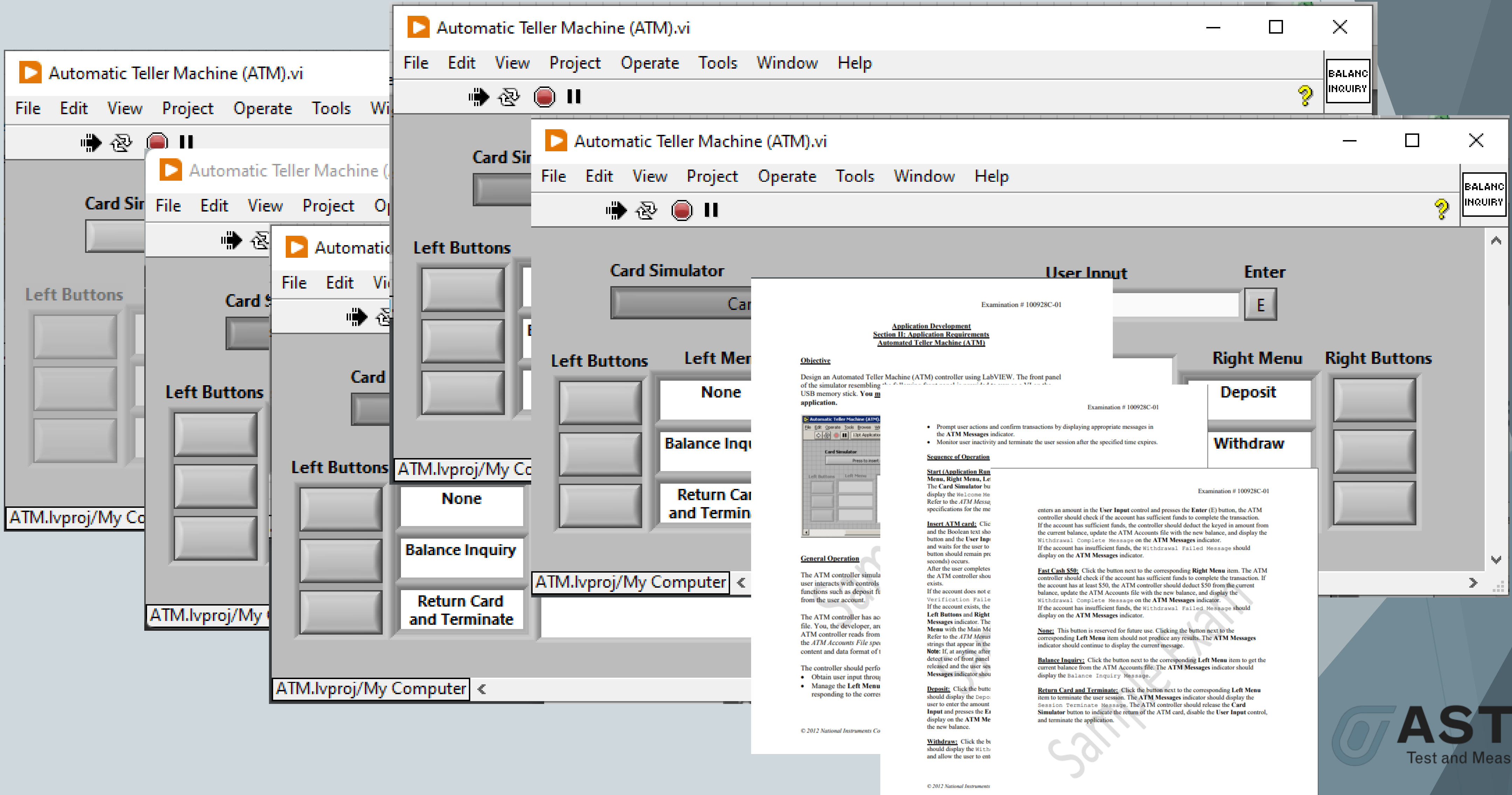
ASTEMES
Test and Measurement Solutions

Test Driven Design

- What makes code testable?
  - Loose coupling
  - Clear APIs
  - Well managed dependencies
  - Clear responsibilities
  - Limited side effects
- What if we start with writing a test?

ASTEMES
Test and Measurement Solutions

The ATM CLD Exam



Examination # 100928C-01

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

Objective

Design an Automated Teller Machine (ATM) controller using LabVIEW. The front panel of the simulator resembling
USB memory stick. **You m**
application.

General Operation

The ATM controller simula
user interacts with controls
functions such as deposit fu
from the user account.

The ATM controller has ac
file. You, the developer, are
ATM controller reads from
the *ATM Accounts File spec*
content and data format of t

The controller should perfo
- Obtain user input throu
- Manage the **Left Menu** responding to the corres

© 2012 National Instruments Co

Examination # 100928C-01

- 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
Menu, Right Menu, Le
The **Card Simulator** bu
display the `Welcome Me`
Refer to the *ATM Messa*
specifications for the me

Insert ATM card: Clic
and the Boolean text sho
button and the **User Inp**
and waits for the user to
button should remain pre
seconds) occurs.
After the user completes
the ATM controller shou
exists.
If the account does not e
`Verification Faile`
If the account exists, the
Left Buttons and **Right**
Messages indicator. The
Menu with the Main Me
Refer to the *ATM Menus*
strings that appear in the
Note: If, at anytime after
detect use of front panel
released and the user ses
Messages indicator shou

Deposit: Click the butto
should display the `Depo`
user to enter the amount
Input and presses the **E**
display on the **ATM Me**
the new balance.

Withdraw: Click the bu
should display the `With`
and allow the user to ent

© 2012 National Instruments

Examination # 100928C-01

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.

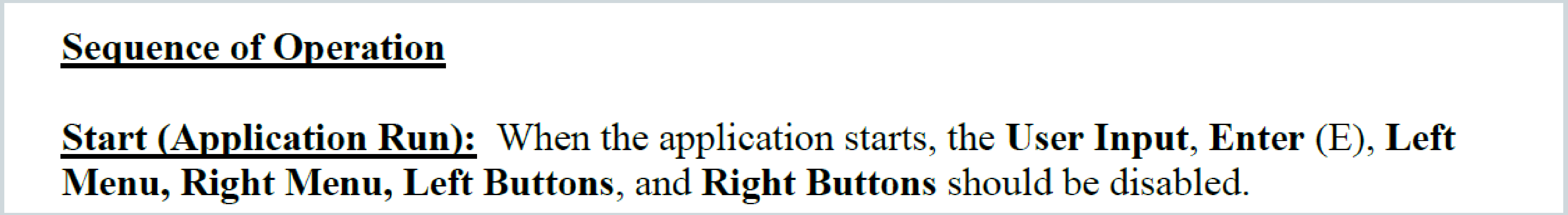
ASTEMES
Test and Measurement Solutions

Reality check

- Could this be a real application?
- Could these be a real Requirements?
- Where is the hardware?
- But it is still a toy example, right?

ASTEMES
Test and Measurement Solutions

Start with a test…

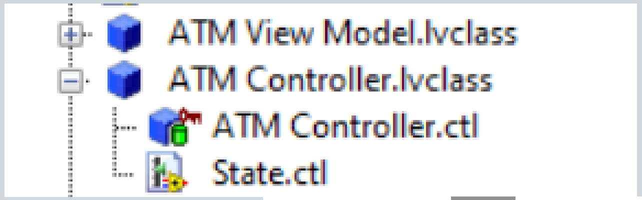

<u>Sequence of Operation</u>

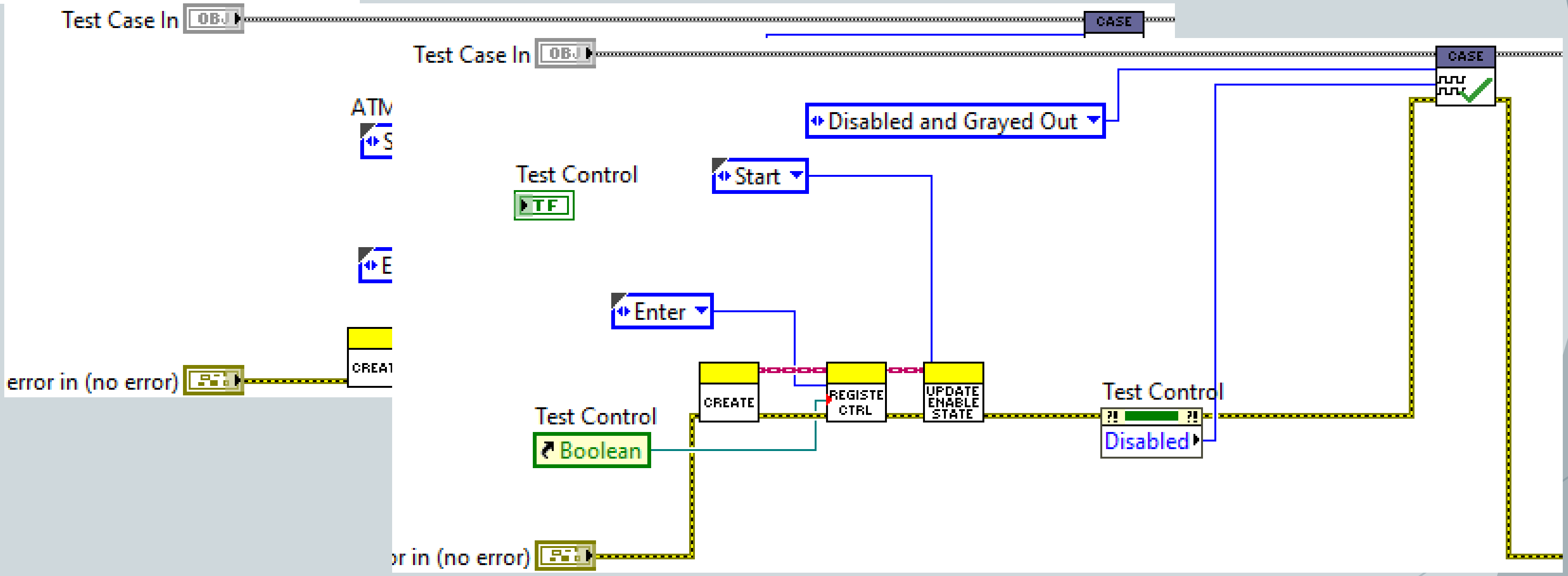
<u>Start (Application Run):</u> When the application starts, the **User Input**, **Enter** (E), **Left Menu, Right Menu, Left Buttons**, and **Right Buttons** should be disabled.

- UI testing…
- When something is difficult to test – get rid of the stuff which makes testing difficult
- ViewModel abstraction

ASTEMES
Test and Measurement Solutions

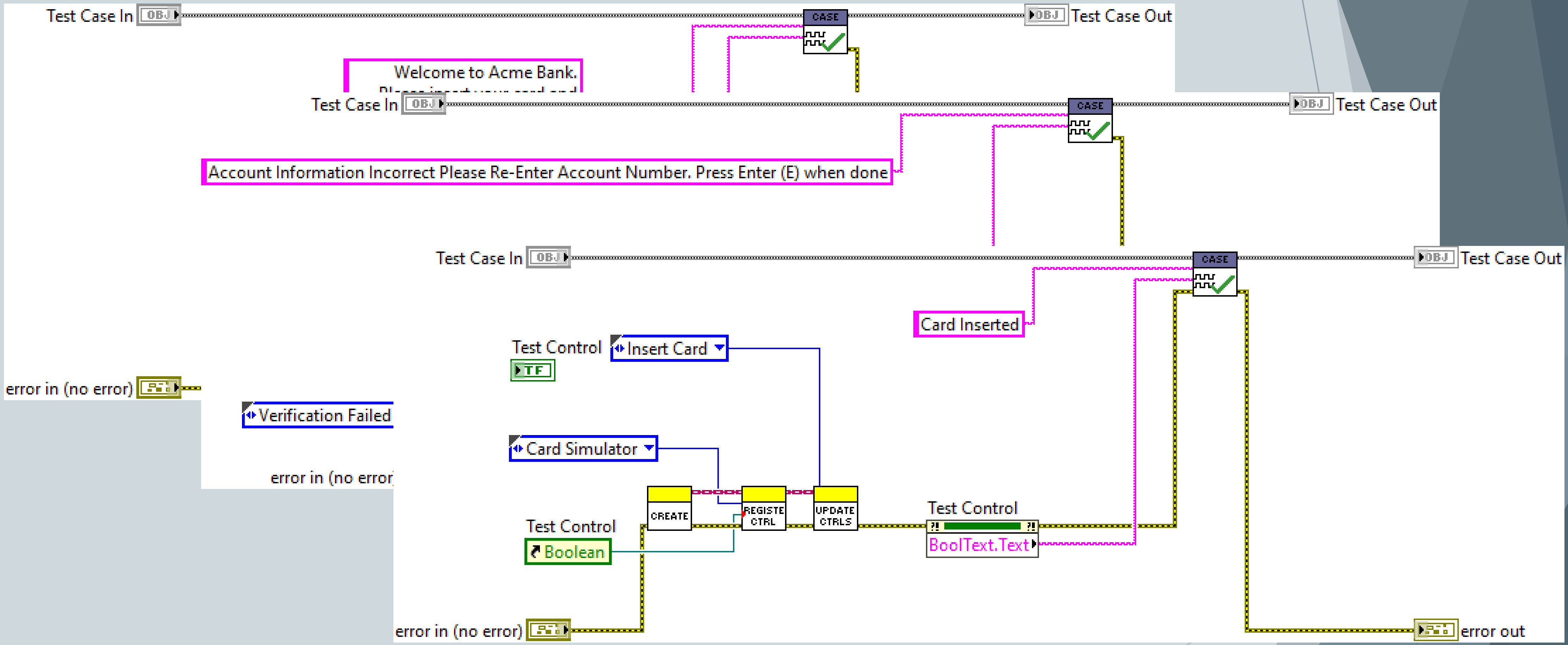
The View Model





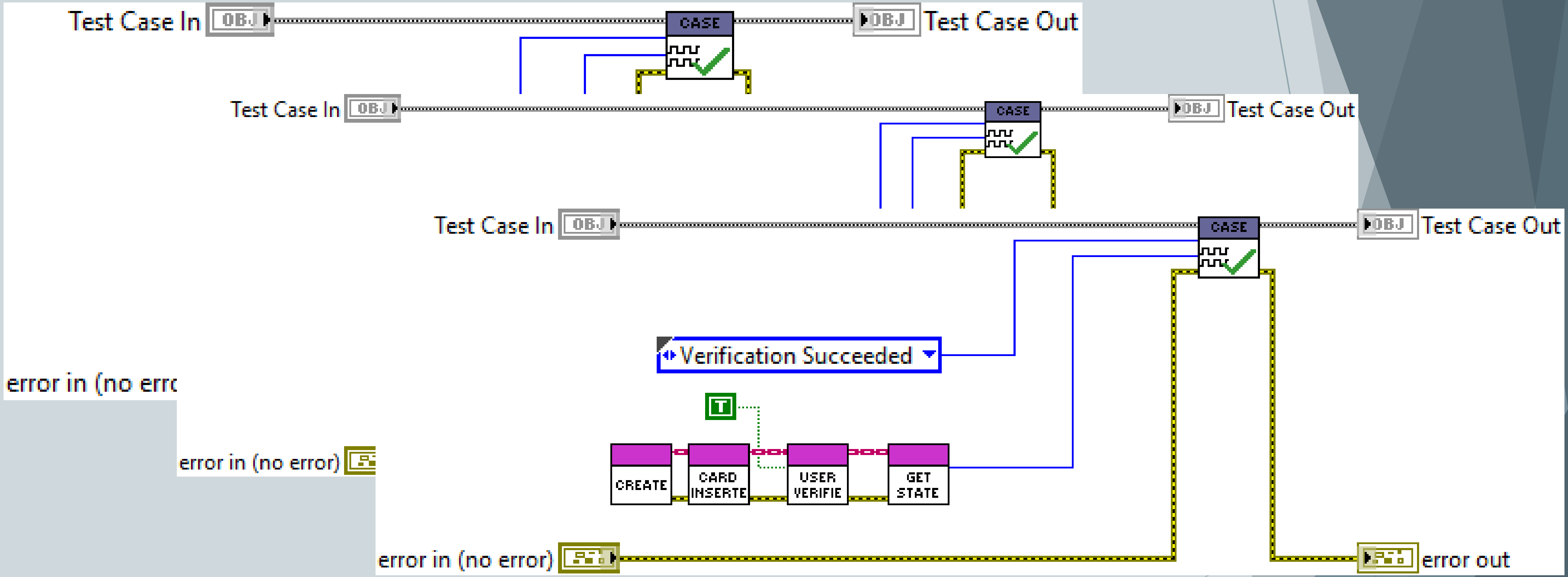
ASTEMES
Test and Measurement Solutions

View Model Test by Test



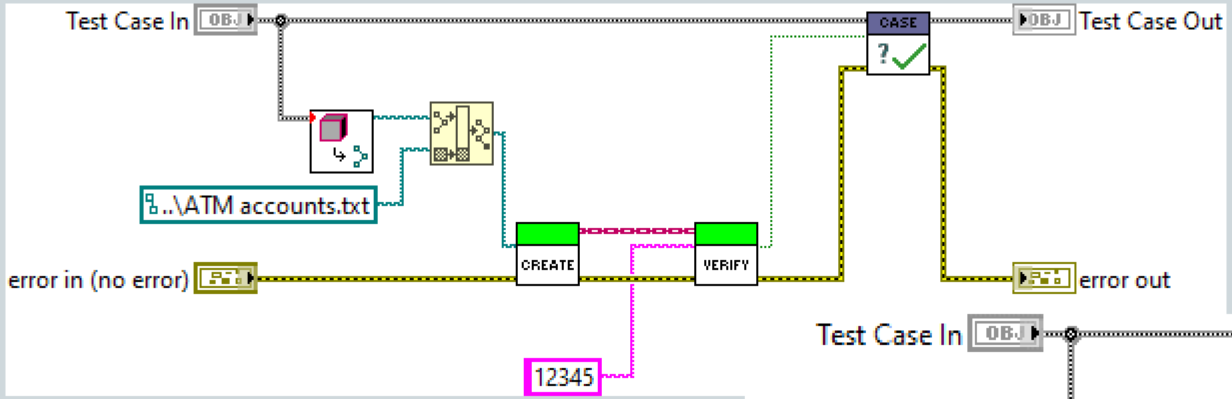
ASTEMES
Test and Measurement Solutions

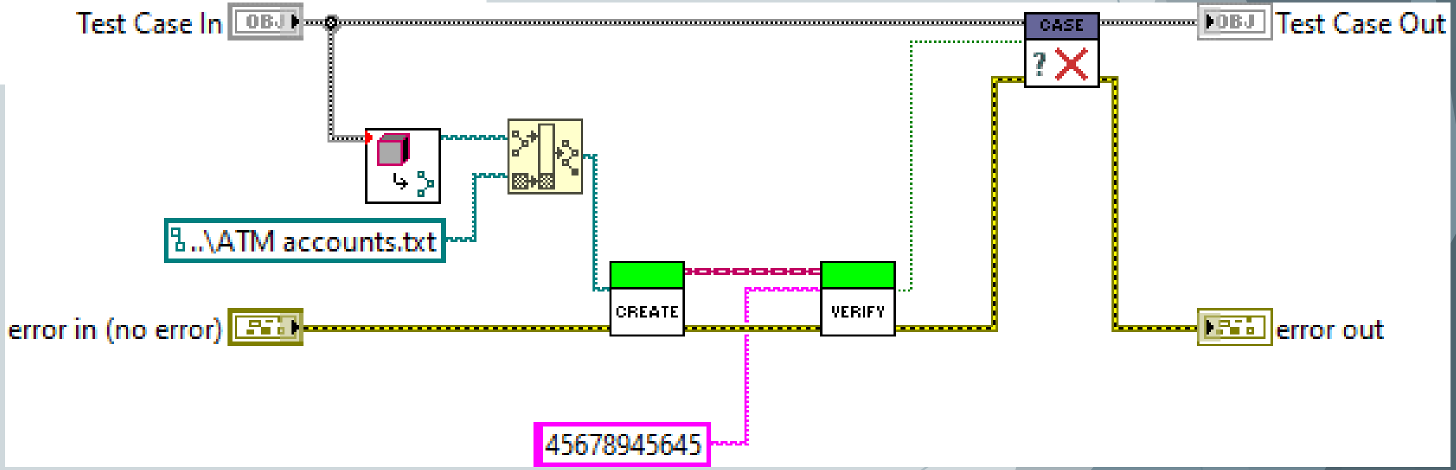
Controller



ASTEMES
Test and Measurement Solutions

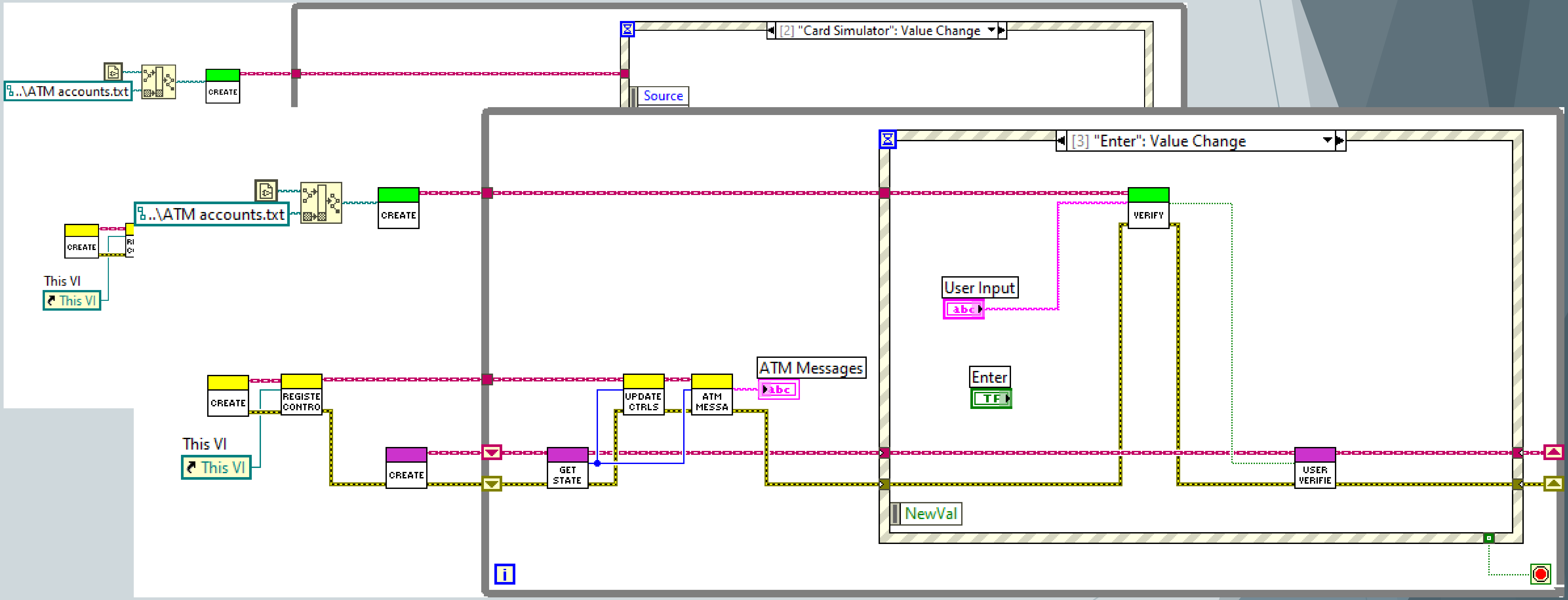
Database





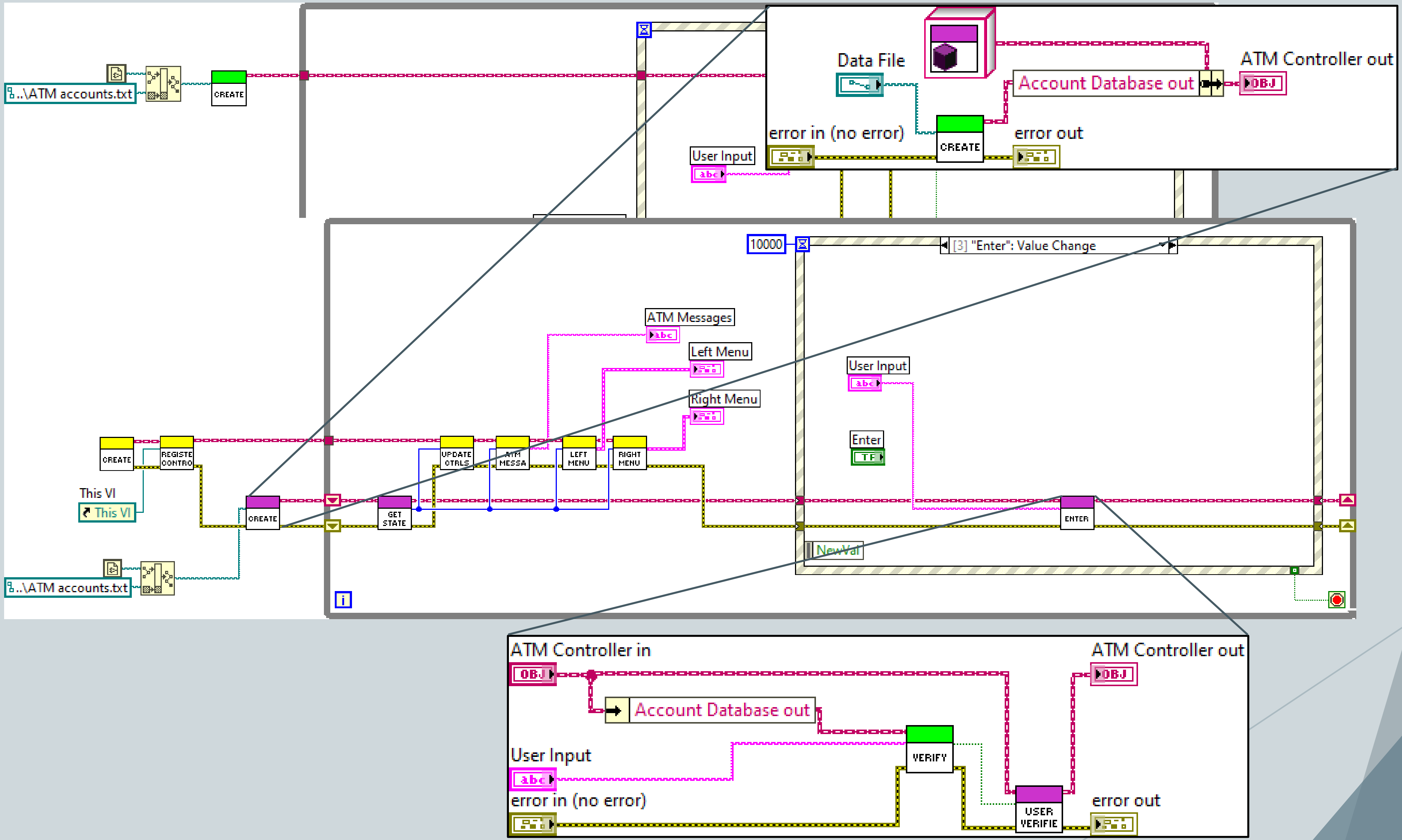
ASTEMES
Test and Measurement Solutions

Integration



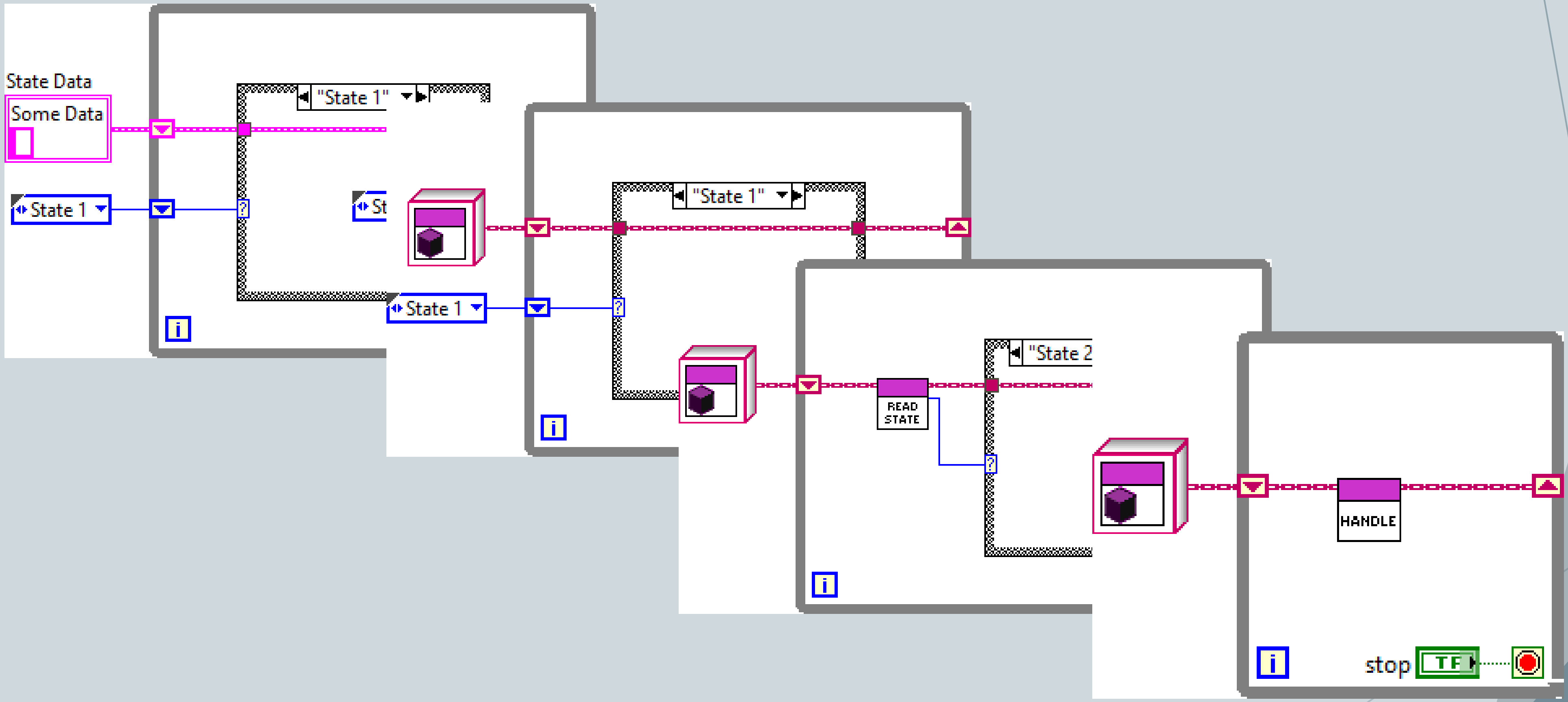
ASTEMES
Test and Measurement Solutions

Refactoring



ASTEMES
Test and Measurement Solutions

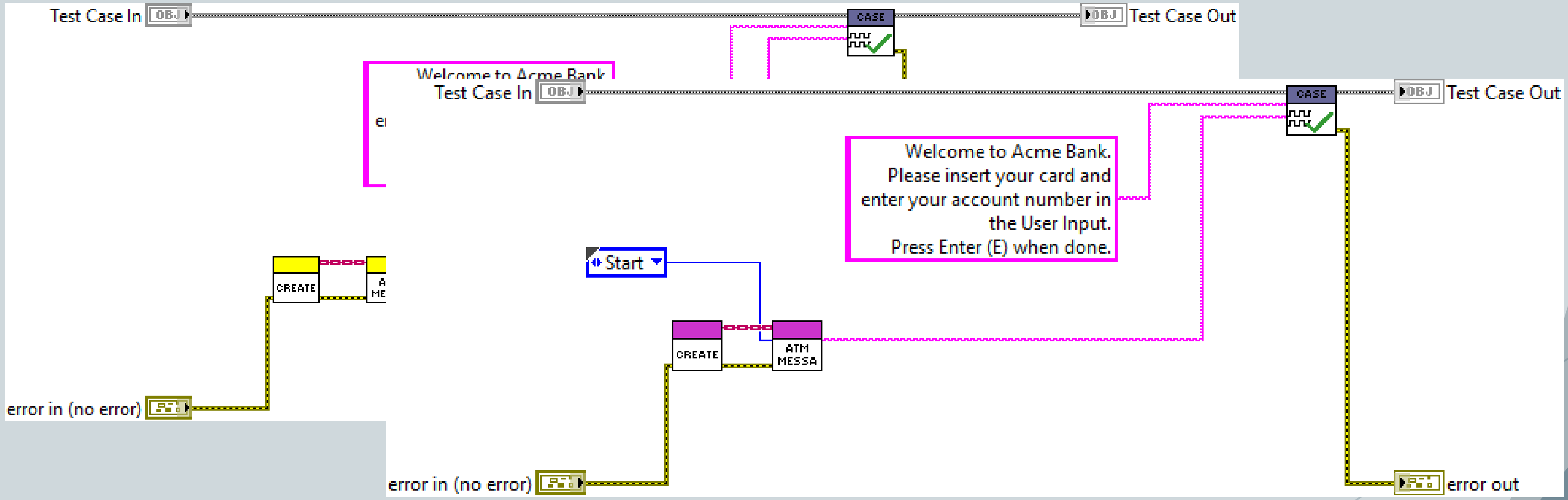
What about state machines?



ASTEMES
Test and Measurement Solutions

Requirements changes…

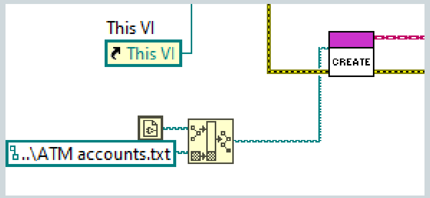
- Some ATM messages contain data from the database
- ViewModel knows nothing about databases
- Should to move responsibility from View Model to Controller



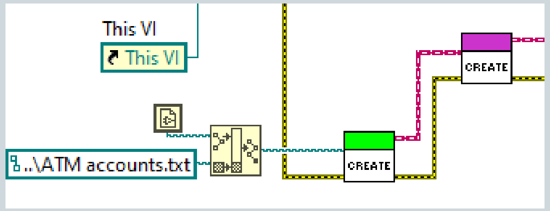
Welcome to Acme Bank.
Please insert your card and
enter your account number in
the User Input.
Press Enter (E) when done.

Testing with the Database

- Needed to test making updates to Database



Refactor

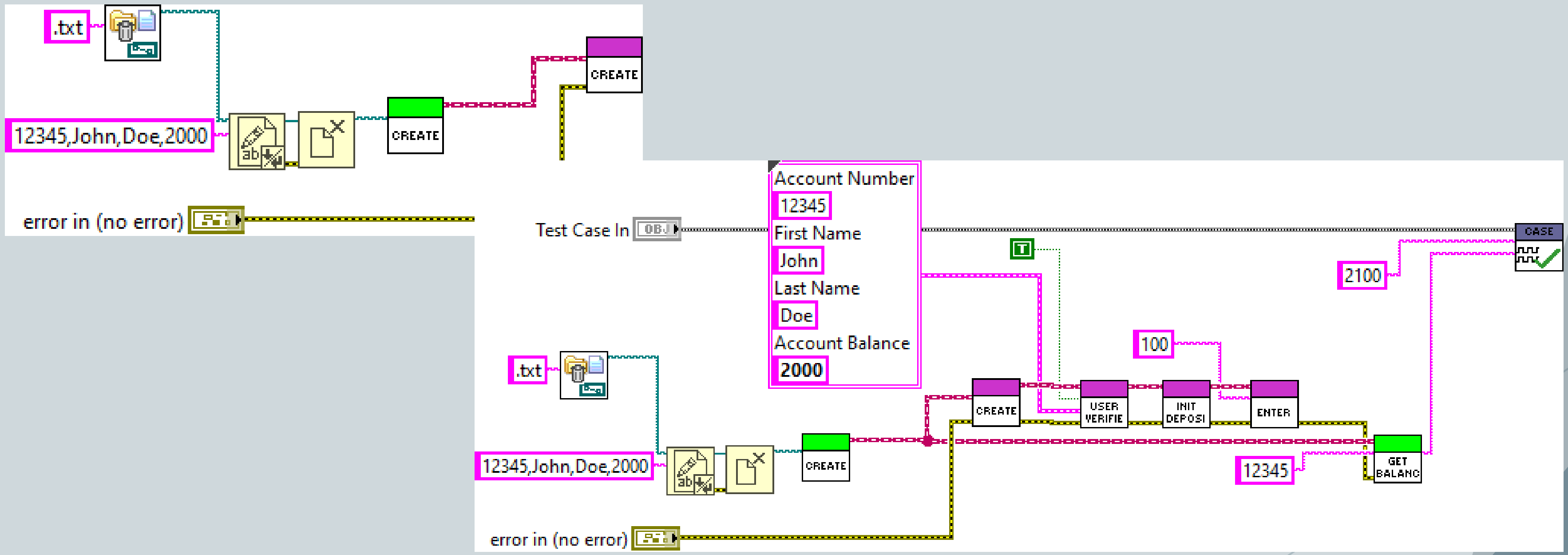


Can now replace database with test double during testing!

ASTEMES
Test and Measurement Solutions

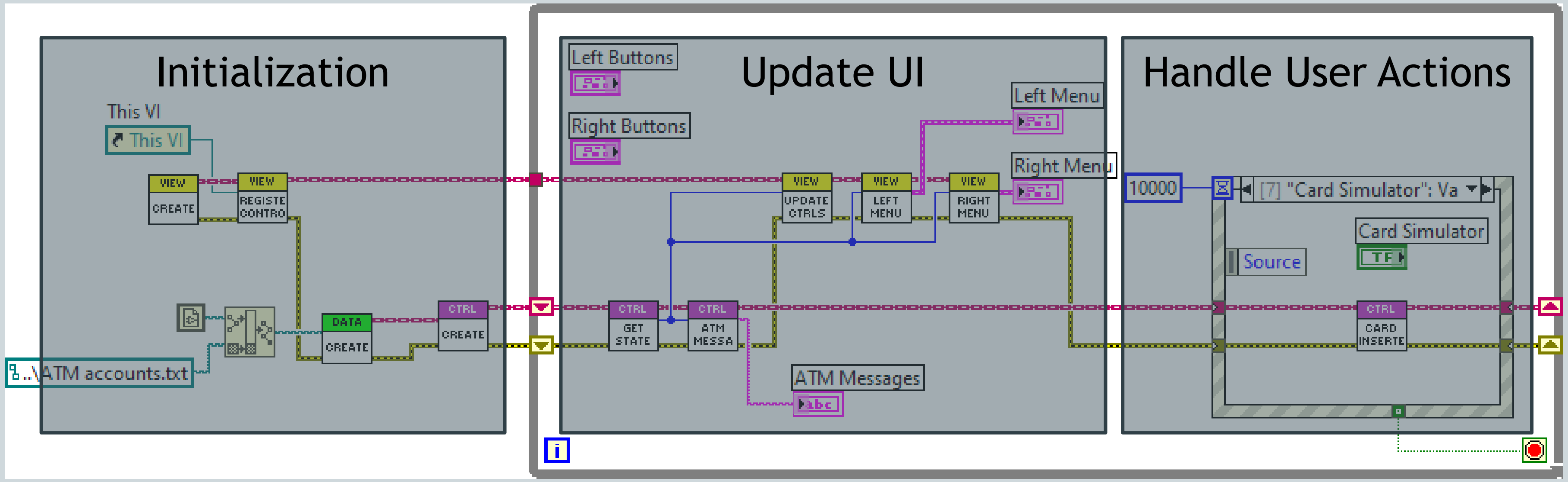
Simple is always best

- Abandoned idea of writing a test double
- Used actual database instead with test data



ASTEMES
Test and Measurement Solutions

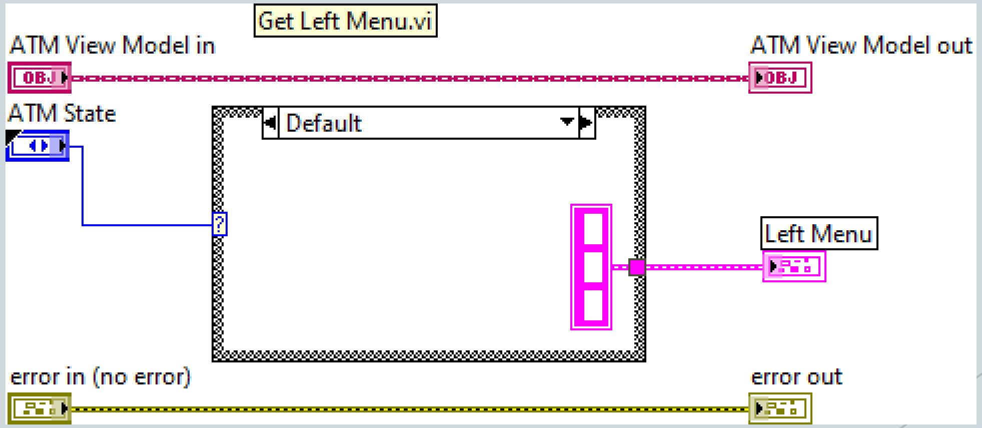
Final Main VI



ASTEMES
Test and Measurement Solutions

Finished, or?

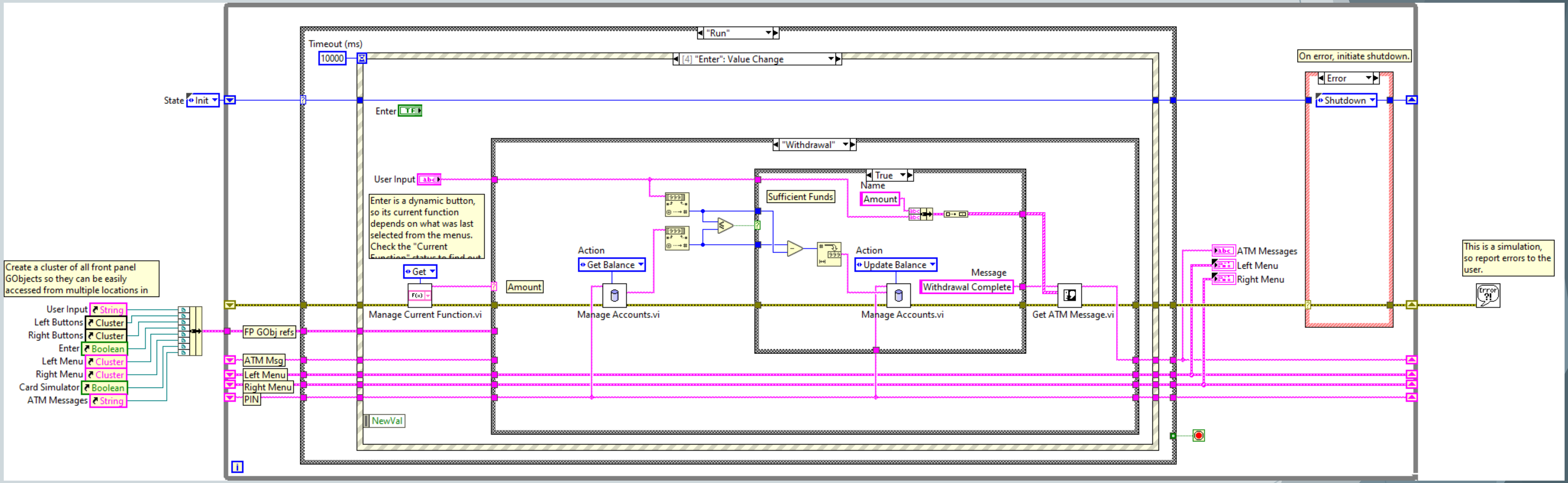
- All requirements were tested and passed – but had never run the main vi
- Discovered implicit requirements
  - Controls to remain enabled between states
  - Menu to remain visible
  - Front Panel initialization



ASTEMES
Test and Measurement Solutions

A non-TDD solution

NI:s provided solution:



Enter is a dynamic button, so its current function depends on what was last selected from the menus. Check the "Current Function" status to find out

Create a cluster of all front panel GObjects so they can be easily accessed from multiple locations in

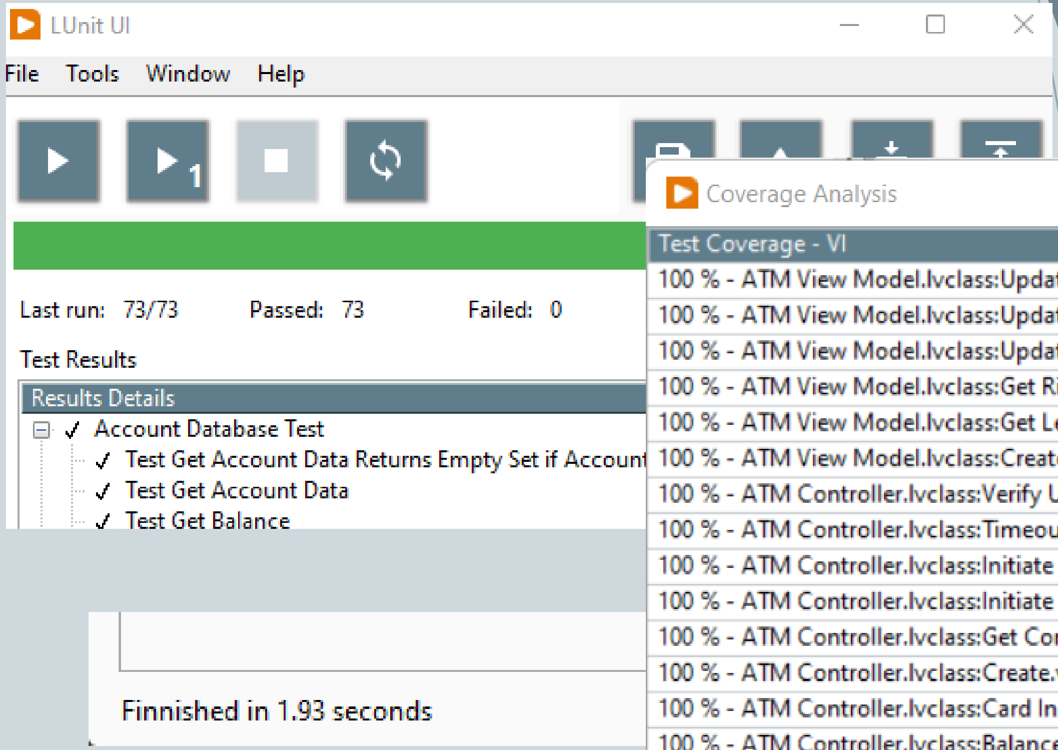
On error, initiate shutdown.

This is a simulation, so report errors to the user.

ASTEMES Test and Measurement Solutions

Statistics

- Exercise solved in ~4h
- 73 Test VI:s
- 100's of Test executions
- Most code exercised by tests
- Almost no debugging



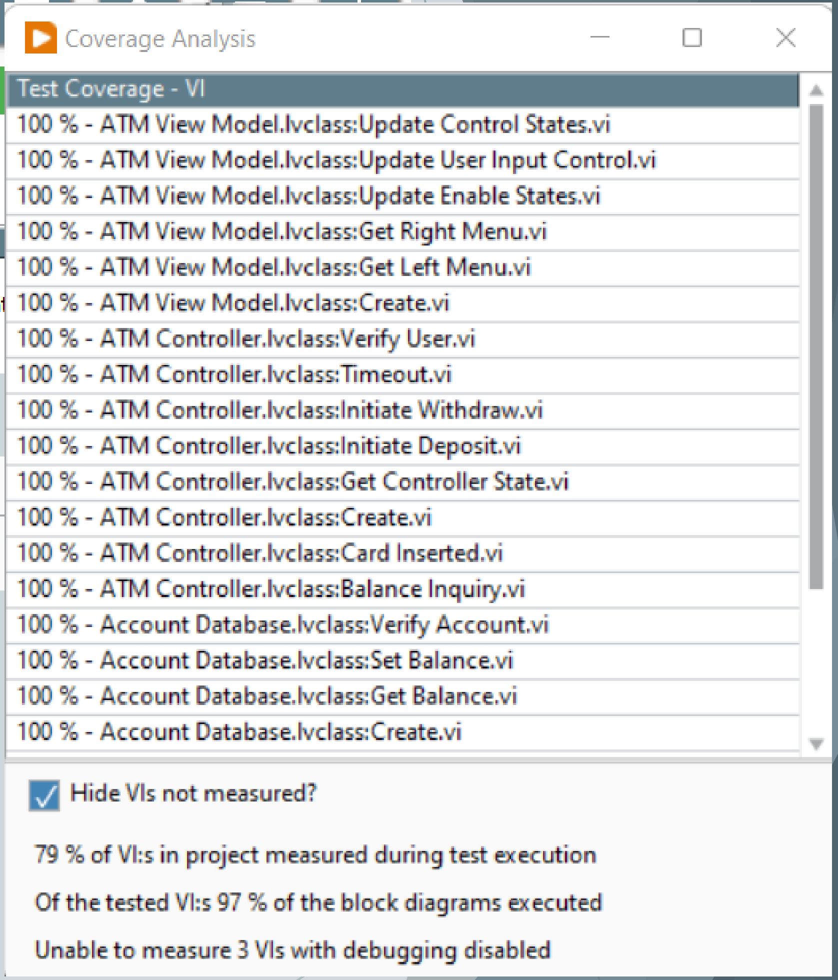
Last run: 73/73 Passed: 73 Failed: 0

Test Results

Results Details

- ✓ Account Database Test
  - ✓ Test Get Account Data Returns Empty Set if Account
  - ✓ Test Get Account Data
  - ✓ Test Get Balance

Finnished in 1.93 seconds



Test Coverage - VI

100 % - ATM View Model.lvclass:Update Control States.vi
100 % - ATM View Model.lvclass:Update User Input Control.vi
100 % - ATM View Model.lvclass:Update Enable States.vi
100 % - ATM View Model.lvclass:Get Right Menu.vi
100 % - ATM View Model.lvclass:Get Left Menu.vi
100 % - ATM View Model.lvclass:Create.vi
100 % - ATM Controller.lvclass:Verify User.vi
100 % - ATM Controller.lvclass:Timeout.vi
100 % - ATM Controller.lvclass:Initiate Withdraw.vi
100 % - ATM Controller.lvclass:Initiate Deposit.vi
100 % - ATM Controller.lvclass:Get Controller State.vi
100 % - ATM Controller.lvclass:Create.vi
100 % - ATM Controller.lvclass:Card Inserted.vi
100 % - ATM Controller.lvclass:Balance Inquiry.vi
100 % - Account Database.lvclass:Verify Account.vi
100 % - Account Database.lvclass:Set Balance.vi
100 % - Account Database.lvclass:Get Balance.vi
100 % - Account Database.lvclass:Create.vi

✓ Hide VIs not measured?

79 % of VI:s in project measured during test execution

Of the tested VI:s 97 % of the block diagrams executed

Unable to measure 3 VIs with debugging disabled

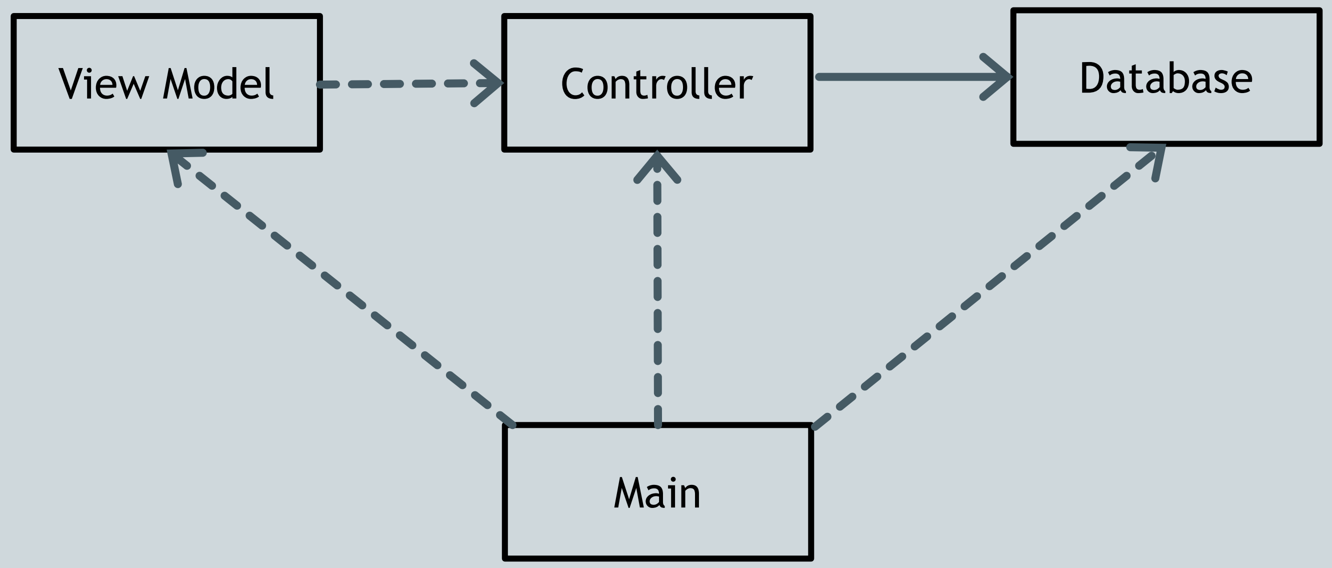
ASTEMES
Test and Measurement Solutions

What does it feel like test driving?

- Less stress
- High Confidence
- Feeling of being in control
- Addictive…
- Slow… but is it?

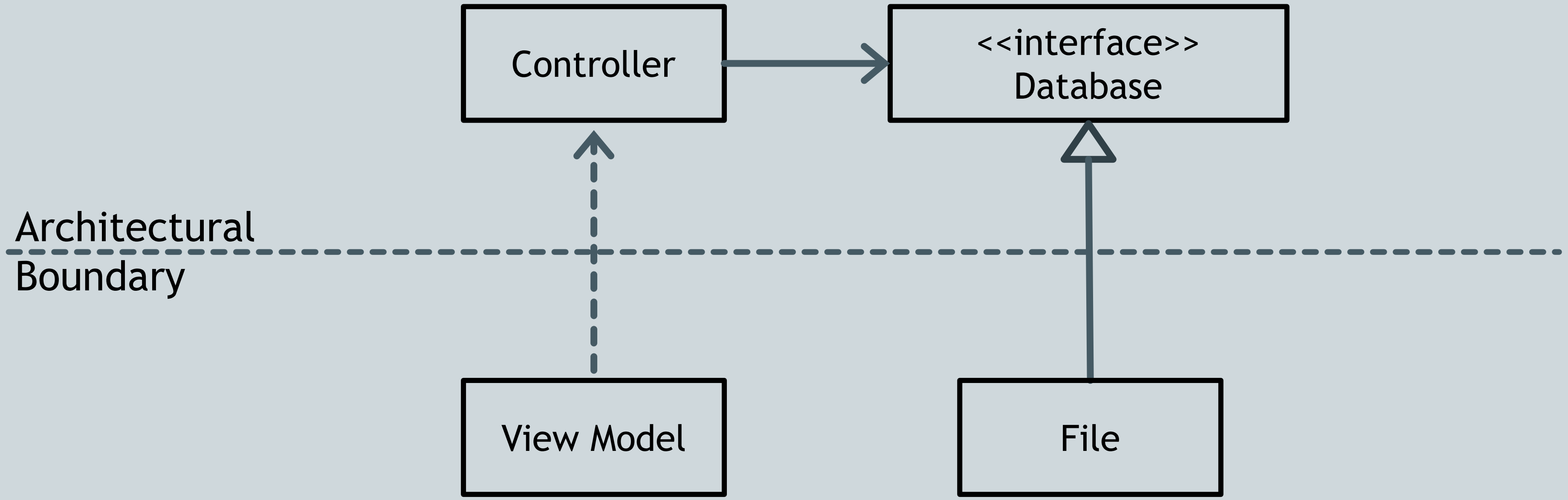
ASTEMES
Test and Measurement Solutions

What about architecture?

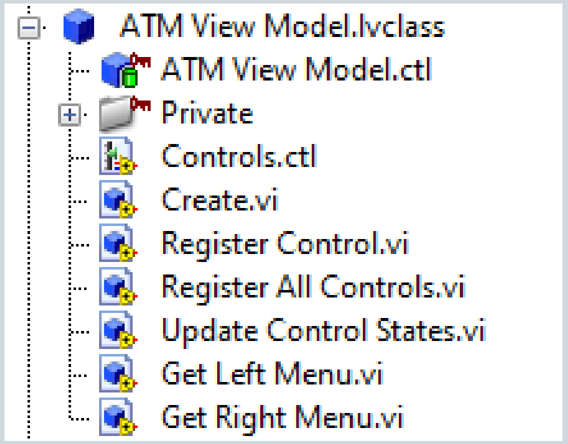


ASTEMES
Test and Measurement Solutions

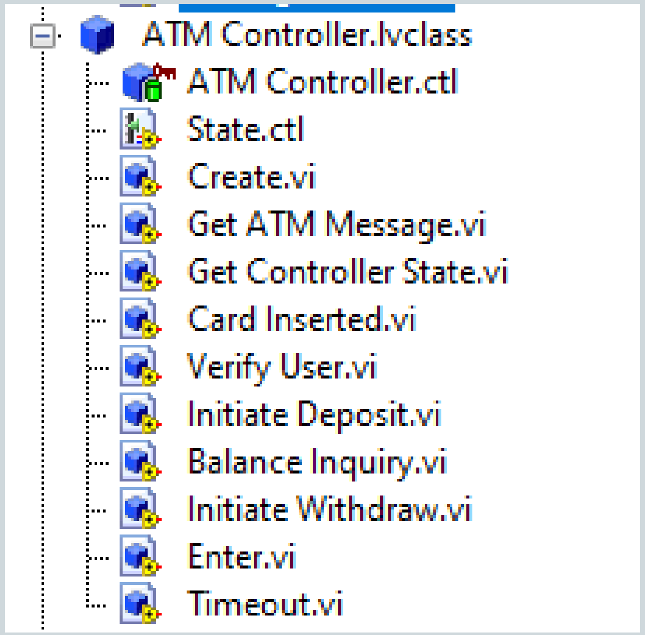
Refactored Architecture



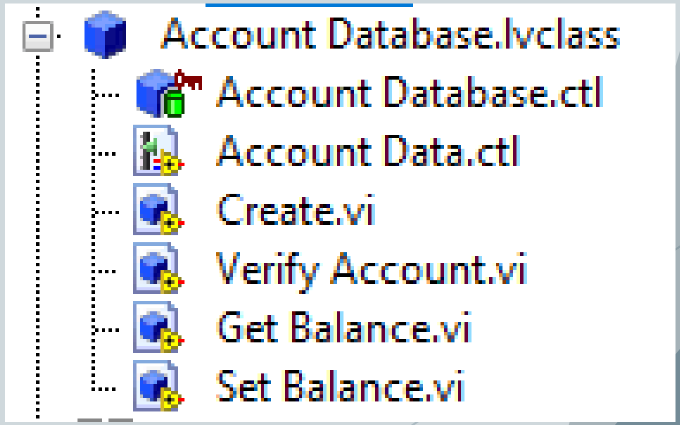
What about the APIs?



- ATM View Model.lvclass
  - ATM View Model.ctl
  - Private
  - Controls.ctl
  - Create.vi
  - Register Control.vi
  - Register All Controls.vi
  - Update Control States.vi
  - Get Left Menu.vi
  - Get Right Menu.vi



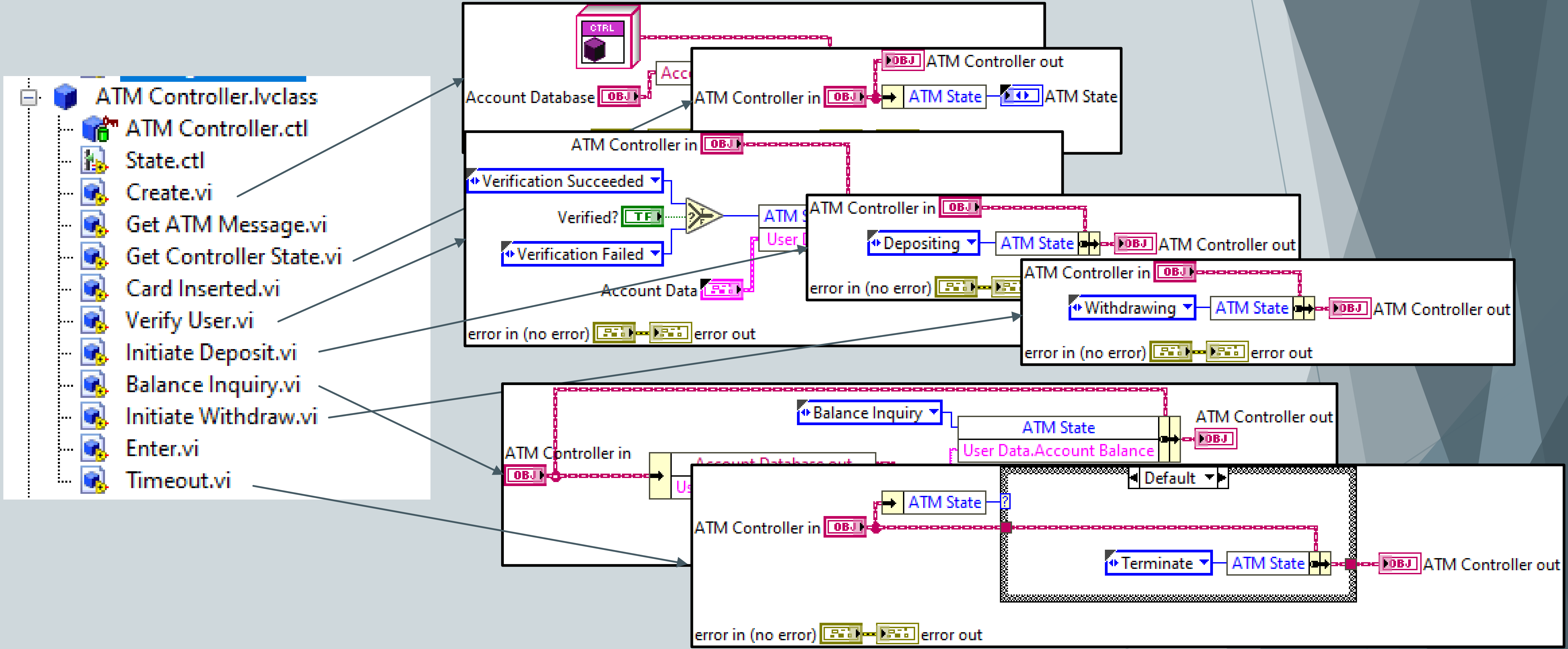
- ATM Controller.lvclass
  - ATM Controller.ctl
  - State.ctl
  - Create.vi
  - Get ATM Message.vi
  - Get Controller State.vi
  - Card Inserted.vi
  - Verify User.vi
  - Initiate Deposit.vi
  - Balance Inquiry.vi
  - Initiate Withdraw.vi
  - Enter.vi
  - Timeout.vi



- Account Database.lvclass
  - Account Database.ctl
  - Account Data.ctl
  - Create.vi
  - Verify Account.vi
  - Get Balance.vi
  - Set Balance.vi

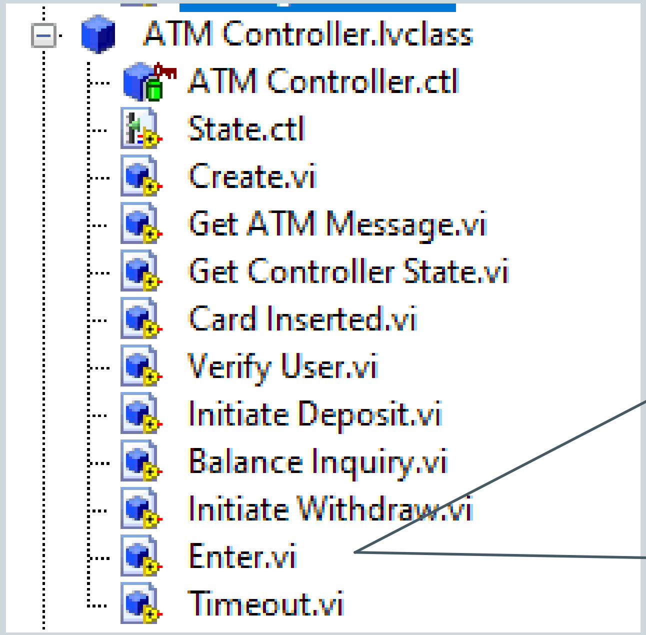
ASTEMES
Test and Measurement Solutions

And what is behind the API:s?

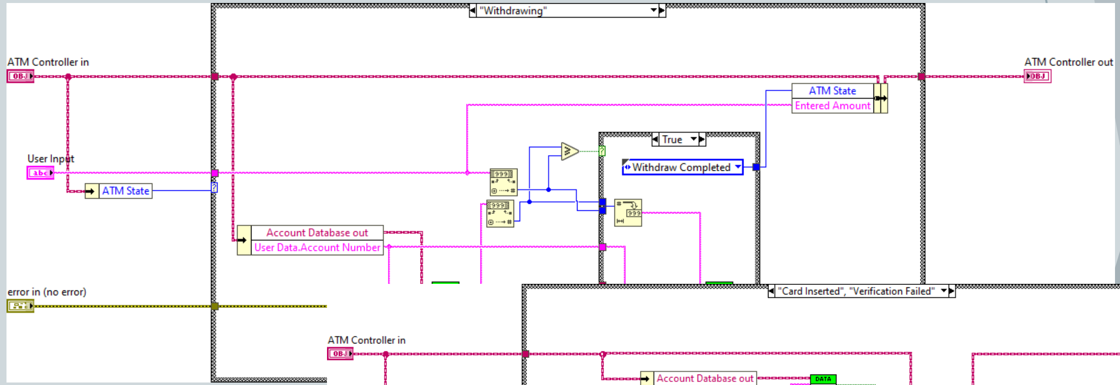


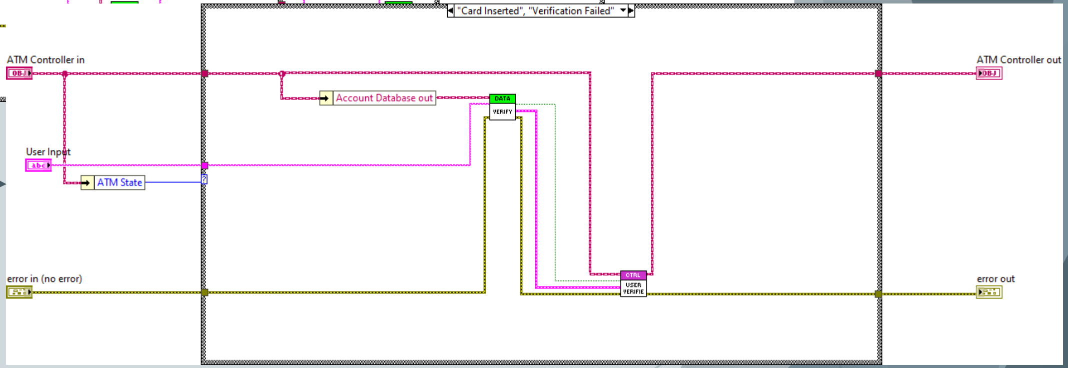
- ATM Controller.lvclass
  - ATM Controller.ctl
  - State.ctl
  - Create.vi
  - Get ATM Message.vi
  - Get Controller State.vi
  - Card Inserted.vi
  - Verify User.vi
  - Initiate Deposit.vi
  - Balance Inquiry.vi
  - Initiate Withdraw.vi
  - Enter.vi
  - Timeout.vi

And what is behind the API:s?



ATM Controller.lvclass
- ATM Controller.ctl
- State.ctl
- Create.vi
- Get ATM Message.vi
- Get Controller State.vi
- Card Inserted.vi
- Verify User.vi
- Initiate Deposit.vi
- Balance Inquiry.vi
- Initiate Withdraw.vi
- Enter.vi
- Timeout.vi





ASTEMES
Test and Measurement Solutions

Where to go from here?

- Code for ATM CLD solution available on GitHub

  https://github.com/astemes/astemes-glasummit-2022

- TDD requires practice
  - Solve practice problems
  - Pair with someone more experienced
  - Start by religiously following the discipline

- For a demonstration of the TDD process – See my Snake game demo
- Handling Hardware when doing TDD – See my GDevCon 2022 presentation