ATM Machine

Left off on:

* Documenting in the following locations
  + Current Features
  + Future Features
  + Classes
* Coding
  + ATMMenu.java
    - Check this for what’s missing

To Do:

* Functional requirements documentation (current features and classes)
* User Documentation
  + Includes details on what to do from start to finish for using the software
  + Include screenshots

Current Features:

* Login system
  + Username and PIN verification
  + Checks for login requirements
    - Username and PIN cannot be any symbol
    - Username cannot be greater than 30 characters
    - PIN cannot be anything other than 6 digits
* …

Future Features:

* Add PIN encryption
* Create an Admin user 🡪 This should have a new ATMMenu.java class (ATMMenuAdmin)
  + This will have new ATM Menu options
    - Register new user system
      * Check for account creation requirements
        + Username and PIN cannot be any symbol
        + Username cannot be greater than 30 characters
        + PIN cannot be anything other than 6 digits
      * Check for duplicate user
    - Delete user
      * …
    - Modify user
      * Change username
      * Change PIN
        + Should have power to delete old PIN and create a new one without knowing what the PIN was
* …

Classes:

* Main.java
* SignIn.java
* …

// Requriements:

// Asks for username and PIN

// Type in username, press Enter, then PIN and press Enter (figure this out)

// Check if username exists in DB --> Pull this from a file with name and PIN in

// CSV format

// If either name or PIN don't match, return "Error with username or PIN"

// If both are correct, show ATM options:

// Quit

// Withdraw

// Deposit

// Transfer

    // Different Accounts (work on this after other functionality)

// Transaction History

    // Maintain the last 30 transactions and use for statistical information

    //-------------------------------- Everything below, place in Account Stats (DataHandler.java)

    // Show transactions from greatest to smallest and have an option that does the

    // reverse

    // Show greatest withdrawl

    // Show greatest deposit

// Account Stats (use last 30 transactions)

    // Show the average withdrawn

    // Show the average of Deposited

    // Show percentage of money deposited vs overall funds at that time

    // Show percentage of money withrawn vs overall funds at that time

    // Percentage of withdraws vs deposits

// test cases

// Can't withdraw more than balance $1,000

// Can't transfer more than $1,000 within the first 30 transactions

// edge cases

// Can't deposit negative numbers

// Deposit only up to the hundreths place

// Can't deposit anything over $10,000

// Classes

// Main - driver class

// Messages class //This will make sure that our verbage is the same across the

// entire platform --> ATMMenu.java

// Greeting/Ending

// Error messages

// Coding specific

// Account specific (ex. "not enough funds")

// SignIn class

// Accounts class - This will read and write to CSV file for the following:

// username, PIN, Balance Amount, [30 transactions] --> later

// ATMMenu class

// Formatting purposes, certain messages should have new lines

        // \*\*\*\*\*NOTE: Can have multiple transactions

            //\*COMPLETE\* After 1 transaction, it sends you back to main menu

            //\*COMPLETE\* Change "selection" variable in ATMMenu.java to all lowercase

                // Add or clause to if statement for if someone typed out the entire selection instead of a number

        // Sign In criteria

            //\*COMPLETE\* PIN should take only numbers and be 6 digits

            //\*COMPLETE\* Username should have a 16 character limit

        //\*COMPLETE\* Do we close the csv file handlers? maybe making new ones will work? --> This may fix itself, try closing the files AFTER we write to it.

        //\*COMPLETE\* ATMMenu.java: Make a new method (ATMMenu.java) that checks if user input a string that says "back" (in any casing). If yes, break.

        //\*COMPLETE\* SignIn.java: Add requirement to "checkRequirements()" that checks if user input is a string that says "quit" (in any casing) or an inavlid entry. Handle this with if-else statements inside same method

        // Custom error messages in "checkRequirements()" for each wrong requirement

        // Add a Create an account feature? --> At sign in, look for "quit", "username" or "register" and add this to signin message

            // Check "checkRequirements()" for valid username and pin criteria

                // Test "checkDupUsers()" --> Place this after the "checkRequirements()"

        // Fix DataHandler.java FOR use with showing balances

        // Round all numbers to the hundreths place down (.floor())