**PROGRAM DESIGN**

* One Class called StockActivity
* Another Class called StockDisplay
* Have two top level windows
  + First for storing & searching of stock information
  + Second for display of Stock Activity
* In the first window has:
  + widget for **date** of transaction input from the user (yyyy-mm-dd)
  + widget for **symbol** of transacted stock (e.g., AMZN, FLT) no spaces
  + widget for **transaction** type (i.e., BUY or SELL)
  + widget for stock **quantity** involved (e.g., 100)
  + widget for **price** of transacted stock (e.g., 12.34)
* In the second window has:
  + Summary button displays:
    - Oldest transaction date with stock name (SELECT symbol, MIN(date) FROM stocks;)
    - Newest transaction date with stock name (SELECT symbol, MAX(date) FROM stocks;)
    - Number of unique stock symbols (SELECT symbol, COUNT(symbol) FROM stocks GROUP BY symbol;)
    - Cheapest price paid for a stock with name (SELECT symbol, MIN(price) FROM stocks WHERE trans=’BUY;)
    - Most expensive price paid for a stock with name (SELECT symbol, MAX(price) FROM stocks WHERE trans=’BUY’;)
    - The stock symbol for most traded stock (SELECT symbol FROM stocks
  + Activity button displays:
    - List of all activities from oldest to latest (e.g., 2021-03- GME BUY 100 $123.00) (SELECT \* FROM stocks ORDER BY date;)
  + Export button
    - Outputs all user activities to a file named *stock\_activity.txt*

**Code:**

* Create StockEntry & StockDisplay classes
* Have two windows called master & second
* In the StockEntry class
  + Have a CreateWidgets Method that handles the frame, labels, entry widgets & buttons for the master window
* In the StockDisplay class
  + Have a CreateWidgets method that handles the frame, labels, entry widgets & buttons for the second window
* In the main function create the stocks table and set the columns, date text, symbol text, trans text, quantity integer, price real
* In the Search method located in the StockEntry class, write a query that get all the stocks from the stocks tables given all the entries the use has inputted
* In the Record method located also in the StockEntry class, write a query that inserts the user’s entry into the stocks tables on the condition that they input the right data types in the respective entry widgets. If the user input a wrong data type in any of the entry widgets, then a pop up is displayed to warn the use that they have inputted a valid data type
* Reset function clears everything in the in the entry widgets
* iExit function attempts to kill the application but before it proceeds, the application asks the user to confirm that they want to exit
* there’s a date button in the StockEntry class that when clicked puts the date selected from the calendar into an entry widget to either be used to search the database or add to the database
* In the StockDisplay class, the Summary method, gives a summary of the stocks in the database
* Also, in the StockDisplay class, the Activities method simply shows all the stocks in the database, where the quantity is greater than 1
* The summary reset button handles clearing the info that has been displayed from pressing the Stock Summary button
* The activity reset button handles clearing the info that has been displayed from pressing the Stock Activity button
* The exit button does the same thing as the one in StockEntry class

**Terminal**

* Run the command, “python andy0002-2101-Project2.py” or “python3 andy0002-2101-Project2.py” depending on the version of python you have
* The Tkinter windows open up but the master overlaps the second window so the master window has to be moved a bit