Stock Mate

David Hedeen, Chris Nathan

SEIS630-01 (Saturday (b))

# Project 2 Sprint 2

GitHub: <https://github.com/hedeen/tp_2_stockmate>

SHA-1 ID: XXXXXXXXXXXXXXXXXXX

# How to Run

## Setup

1. Download the gitbub repository at the URI located on the title page.
2. Locate the runtime folder within the project directory (save this location for later)

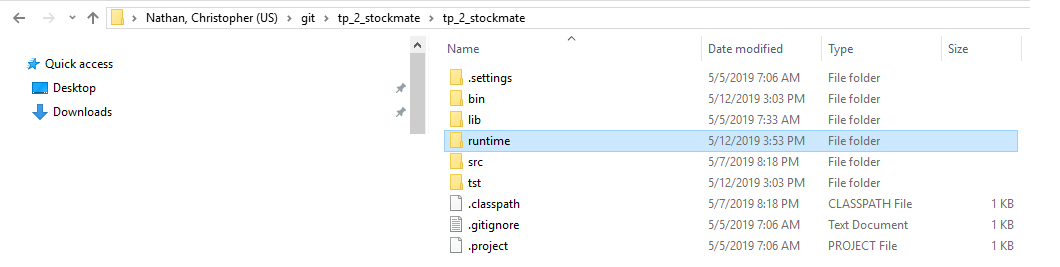


Figure : Runtime Folder

1. Create a temporary folder that will be used to save stock files on your local computer. The default setting will use “C:\Stock Mate”, but you may select any valid directly during program operation
   1. Record the temporary folder created
2. Confirm that you have a valid internet connection when running test cases.
   1. The software will poll data from <http://sec.gov> and will not operate correctly without internet connectitivity

## Starting the software

1. Open up a command line utility (cmd.exe) and navigate to the folder shown above (see below for example)

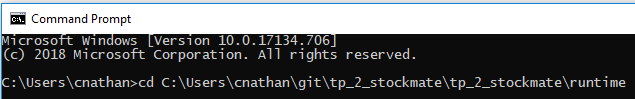


Figure : Change Directory to Runtime Folder

1. Enter in the following “java – jar “Stock Mate.jar” to start the program

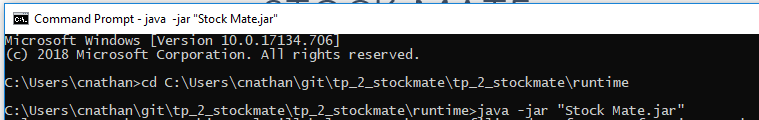


Figure : Run .jar File Using JVM

## Running the Software

1. Follow the prompts that appear on the command line utility and enter in your responses. Press the ‘Enter’ button for each response

### Sequence of Operation & Valid Entries

For some questions, a default response is shown. Press enter without typing any characters to select and use the default response.

1. Select how you want to save files
   1. Enter in a 1 or 2 for .txt or .csv files
2. Enter in a directory where you wish to save stock filing reports.

*NOTE: Enter in the directory that you record in the “Setup” section*.

1. Select the stock ticker you wish to retrieve information about
   1. Enter in any valid stock ticker (e.g. “AAPL” or “GOOG”)
2. Wait for the system to return the number of filings detected. If no filings are detected the software will ask for a new stock ticker
3. Select a filing data tag from the list of support options (earnings per share, earnings per share diluted, income)
4. Select whether you would like to return all filings or just the most recent
5. Wait for the software to retrieve the information
6. Read the message prompt to see the file that was created. Locate the file and review its contents.
7. Continue to use the program by selecting a new stock ticker, repeat as desired.

## Running Unit Test & Evaluating Test COverage

Unit testing can be verified using Eclipse IDE. The software was developed using Version: 2018-12 (4.10.0) although other versions are likely compatible. Confirm that you have a valid internet connection when running test cases.

1. Open the project within eclipse and confirm you can view the project directory on the left side (see below)

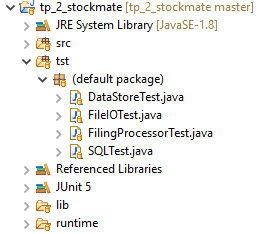
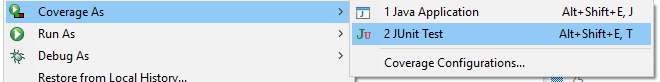


Figure : StockMate Project Directory

1. Locate the ‘tst’ folder and right click it
2. Select “Coverage As > JUnit Test” as shown below



1. Wait for the tests to complete and review their success using the test coverage panel and the test detail panel on the left pane.

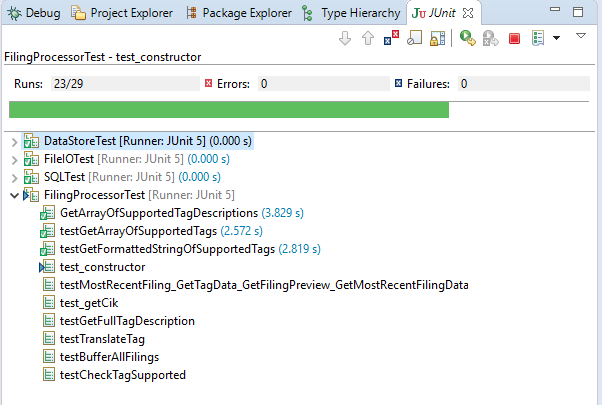


Figure : Test Detail Panel

# Domain Class Diagram

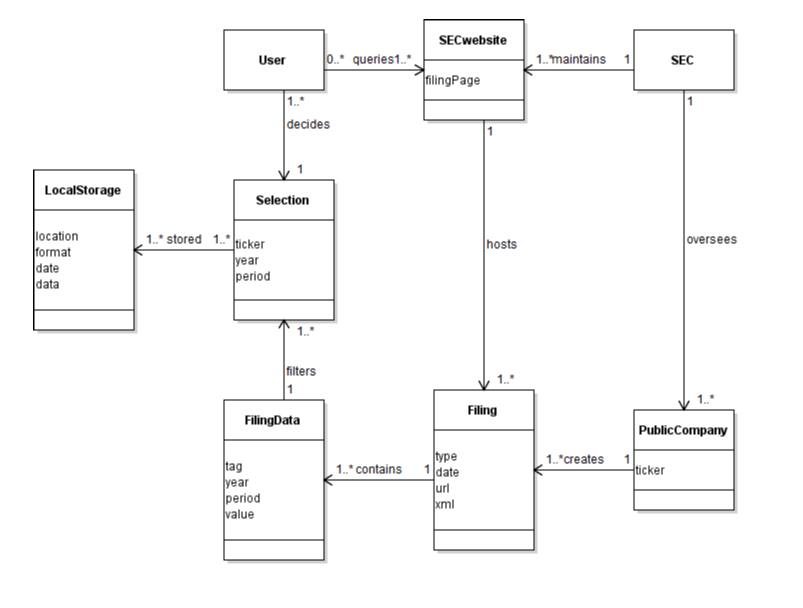


Figure : Domain Class Diagram

# Main Success Scenario (UC1 –Retrieve Stock Information and Save)

**Scope:** Stock trader retrieving and saving stock ticker information

**Level:** User goal

**Stakeholders and Interests:**

Stock trader: Desires easily accessible stock information. Desires accurate stock data in readable format.

Stock trader colleagues: Desires easily readable file format that can be exchanged between PCs.

SEC information technology: Wants stock information to be more easily digested by users.

**Primary Actor:** Stock trader

**Preconditions:** Stock trader has a reliable internet connection and can navigate to <https://www.sec.gov/> homepage. Stock trader is working on a PC with sufficient privileges to create and save files.

**Success Guarantee:** Stock trader has produced a readable .csv file that includes stock data related to the requested ticker.

**Main Success Scenario:**

1. SuD asks user how they would like data to be stored
2. User selects “TXT” file exports and the SuD asks the user to enter in the local folder to save stock files.
3. User enters in a filepath
4. SuD confirms the folder is valid, saves the information and asks user which stock ticker they would like to retrieve

[ -- Repeat until user closes application

1. SuD asks the user which stock ticker there are interested in
2. User enters in a stock ticker
3. SuD returns number of filings present, outputs available (supported) stock tags (earnings per share, income, etc.), and asks which information they would like to retrieve.
4. User enters in a stock tag selection
5. SuD retrieves information from sec.gov website, displays a preview of the data and saves the stock information to the selected local folder.

--- ]

**Extensions:**

1a. User closes the console application (they have no more stocks that they are interested in)

4a. Local folder is invalid

7a. HTTP request to sec.gov fails

9a. HTTP request to sec.gov fails

# External System Events

Two external system events are defined/modeled below:

User Requests a Directory for Saving Files

checkDirectory(String dir) returns true/false for valid directory

User Requests a Stock Ticker:

getFilings(String ticker) returns integer for # of filings located

# GRASP

## TODO

We implemented XXXXXXXX