Software Requirements Specification

for

Stock Screener

Version 2.0

Prepared by Art Fogiel, Michael Harrison, & Kathleen Connell

CMSC 495 - 6382

February 28, 2018

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Kathleen Connell	1/21/18	First combined draft of SRS requirements	1
Kathleen Connell	2/28/18	Final Draft of SRS requirements	2

1. Introduction

1.1 Purpose

Create a user interface application that will allow a user to create an account, login, and retrieve stock information of the user's choice. The application will then allow a user to screen or filter these stocks by enabling defined metrics. Once the filters have been applied, only the stocks that fall into the categories chosen will be returned to the user.

The application will consist of a user interface that allows a user to create an account and login.

Once a user logs in s/he will be directed to another interface that displays stock filters such as price min and max, Market Cap in millions or billions, and Volume min and max.

In an adjacent column, the screened stocks will display Ticker, Company, Price, Volume, Market Cap, and Change in percent. The final column will allow user selection to watch the specific stock. If the user selects the green double arrow in the Watch column, the stock will be shown in an adjacent column of "Watched Stocks."

1.2 Intended Audience and Reading Suggestions

The intended audience for this document is any user who is interested in using a stock screener for analyzing current stock market trends. This document is also intended for academic purposes, where users in computer science fields can access and understand the development of this product. It is recommended that all users read this document in its entirety. While this is important, it is worth noting that any user who simply wishes to use the stock screener without going into the development process can focus on sections 1 and 2 of the SRS document.

1.3 Product Scope

To provide a simple interface for users to identify potential stocks they would like to invest in.

2. Overall Description

2.1 Product Perspective

The product will be an open source Windows-based application that mirrors the design pattern and functionality of current live stock screeners. It is a Graphical UI application implementing the Model View View-Model. The application provides users the ability to filter stocks based on user-defined specifications.

Main features of the product include:

1. Product capabilities: Compatible with Windows 2.7 desktop.

2. User account: The application allows the user to create a username account and login to view up-to-date stock metrics.
3. Filters: Product will have various quantitative constraints a user can implement when searching for stock metrics. Filter info includes price, market cap, and volume.
4. Screened Stocks: Application displays stocks based on user-defined metrics. Screened stock information displayed includes stock ticker, company name, last price, volume, market cap, percent change, and selection arrow for user to elect to watch that stock.
5. Watching: Application displays stocks user is currently watching, with stock name and percent change displayed, as well as user ability to remove watched stock.
6. Other product features: Application includes 'save as favorite' and 'load favorite' buttons for users to select to save and view favored filter metrics quickly. Application also includes a secure log out feature when user is finished using the application.

2.2 Product Functions

Major functions product must perform include:

1. Create a user/login user
2. Display stock screener metrics
3. Allow user to filter stocks based on user-specific requirements
4. Allow user to save filter metrics and reload them using the favorites options
5. Display current user watched stocks
6. Securely log out user

2.3 User Classes and Characteristics

The user is considered to be any individual with the capabilities of operating applications and have a basic understanding of stock information. User has access to account and any changes made by the user are account-specific. Product Administrators have technical support knowledge to maintain and secure the application with full privilege.

2.4 Operating Environment

In order for the product to operate, it fundamentally requires a Windows Desktop Environment. Product requires .NET Framework 4.5.1, which comes integrated on Windows 8.1 or Windows 10, and can be installed onto Windows 7. The application works on both 32-bit and 64-bit computer architectures. All computers require strong connection to the Internet.

2.5 Design and Implementation Constraints

The product will be completed using tools and techniques as directed by project administrators. Deliverables prescribed are defined in section 3. The following are varying design and implementation constraints:

1. In a client-server GUI environment, each user/machine interaction results in some processing at one or more software servers. Therefore, the process of entering a GUI screen and working on a GUI screen will typically result in many system transactions, most of which are transparent and unknown to the user. Each of these transactions requires the user to wait for a response; therefore, each system interaction has a different set of response expectations by the user.
2. Due to large quantity of stock data that needs to be updated upon user login, decision to initialize stock data upon login is taken into account.
3. It is the responsibility of the project Administrators to submit all aspects of the product at their respective assigned timeframes. All admins are required to adhere to best security and programming standards, and are responsible for the maintenance of the delivered product.

2.6 User Documentation

Due to the nature of the product, user is expected to have basic knowledge of stock screeners, therefore, no external documentation is recommended.

2.7 Assumptions and Dependencies

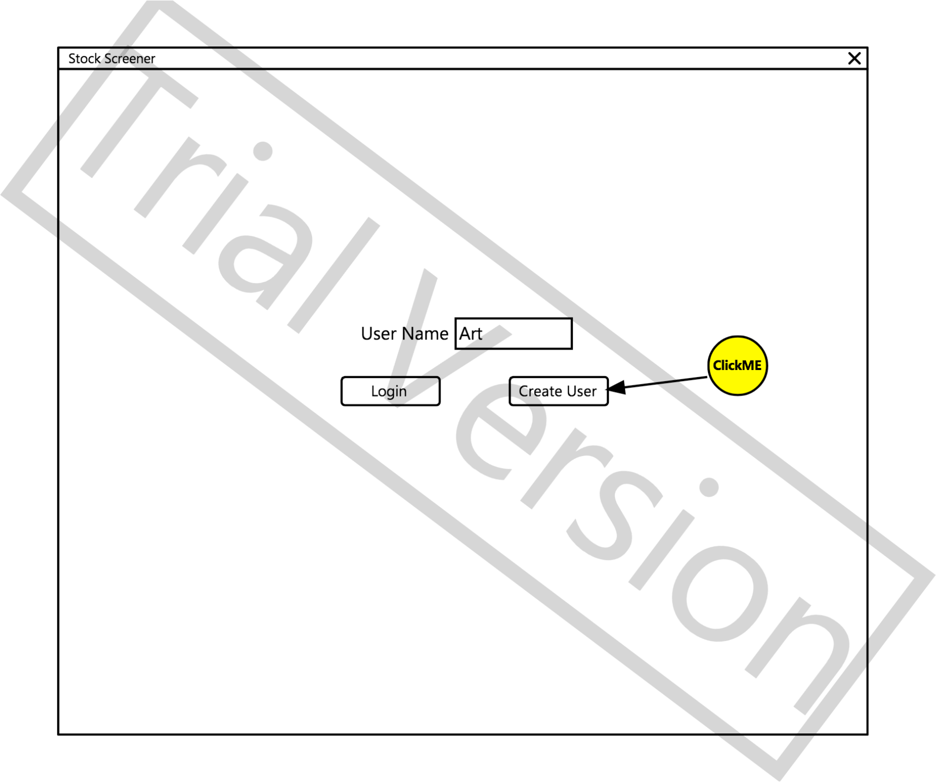
For full product functionality, user is expected to have the proper types and versions of Windows software to utilize product interfaces. User is assumed to have good quality Internet connection on their Windows desktop platform.

3. External Interface Requirements

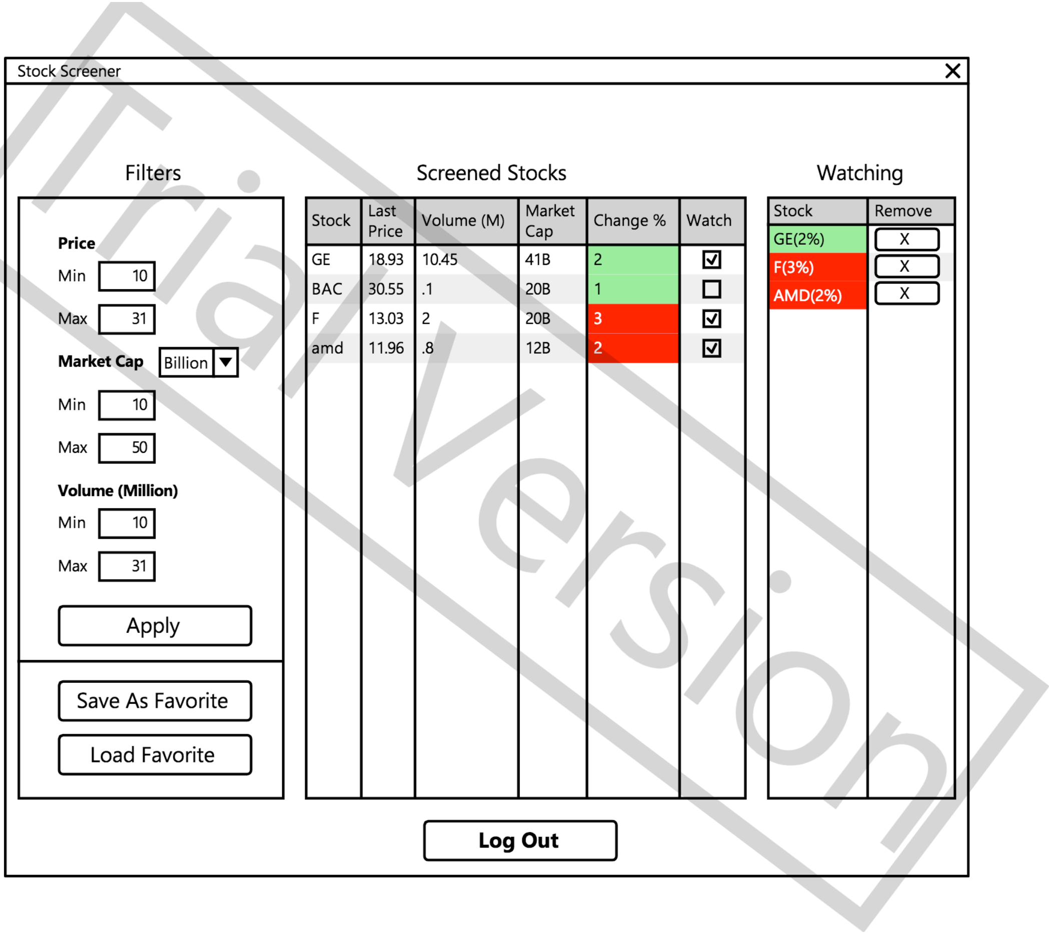
3.1 User Interfaces

The product user interface has been specifically designed with consumer-friendliness in mind, giving them filtered stock metrics in real time. The stock screener UI is designed for easy use, with the User Guide being the only supplemental information on the application itself. The home screen displays the stock screener, showing the stocks and various filter/watch functions that it performs. The user can select stock metrics tailored to their specific needs in the filter section, and then the stock screener is refreshed to display those filtered results. The user can elect to save their filter settings under the ‘favorites’ button, so upon login they can simply apply their favored settings without repeating the filter process. All filter search results pertaining to the queried stocks are displayed. The stocks appear on the screen in a clear and simple format, where the user can then select any stocks they wish to watch.

In the figures below, a mock up of the stock screener product shows the UI from creating a user/login to successfully logging into the account's stock screener page.



User Name Art



Price
Min 10
Max 31

Market Cap Billion ▼
Min 10
Max 50

Volume (Million)
Min 10
Max 31

Stock	Last Price	Volume (M)	Market Cap	Change %	Watch
GE	18.93	10.45	41B	2	☑
BAC	30.55	.1	20B	1	☐
F	13.03	2	20B	3	☑
amd	11.96	.8	12B	2	☑

Stock	Remove
GE(2%)	X
F(3%)	X
AMD(2%)	X

Note: These figures are conceptual, and user interface will change dependent upon desired visuals and implementations.

3.2 Hardware Interfaces

The following physical and logical hardware components are required for product functionality:

1. Standard computer hardware components (power supply unit, motherboard, storage device)
2. Input devices (mouse/touchpad, keyboard)
3. Output devices (computer monitor)
4. Internet connection

3.3 Software Interfaces

The following table lists the software used for stock screener product:

Software Used	**Description**
Operating System	We have chosen Windows OS disk I/O for its support and user-friendly functionality
Developer Platform	To implement the product, we have chosen to use Windows .NET 4.5.1 Framework. This particular framework is open-source, and allows for easy build in CSharp.
Network Capabilities	Our product implements the Windows Networking function, which allows for networking capabilities in the application without making allowances for a particular network provider or physical network implementation.

3.4 Communications Interfaces

The product supports HTTP through Windows WebClient API. Product communicates to https://iextrading.com/developer/docs/#getting-started through their API HTTP calls.

4. System Features

4.1 User Login

4.1.1 Description and Priority

The User Login feature must allow the user to either login to an existing user account, or create a new login. The purpose of a login is to remember the users filter settings and watched stocks. They can then resume where they last used the application next time they login. This feature brings a low risk and low cost to the project with a medium to high impact on the customer's usage.

4.1.2 Stimulus/Response Sequences

1. User Starts the application

2. User enters their username in the text box

3. If the user enters a name in the text box and clicks the "Login" button then if the user exists they will be switched to the screener view.

4. If the user enters a name that does not exist and clicks "Login", a pop up message will appear indicating the user name does not exist. Enter an existing name or click "Create User"

5. If the user clicks "Create User" then the name entered in the text box will be used to generate a new user and the screen will switch to the screener page. If a user

already exists with that name then a pop up will appear indicating user already exists, click "Login" instead.

4.1.3 Functional Requirements

REQ-1: User has ability to enter a user name
REQ-2: User has ability to click a Login button
REQ-3: User has ability to click a Create User button
REQ-4: User must be notified if the username already exists when clicking Create New
REQ-5: User must be notified if the username does not exist when clicking Login
REQ-6: If user logs in with an existing username, the screener view must appear with that users last used filter settings and watched stocks.
REQ-7: If user logs in with a newly created account, the filter setting defaults will be used and no stocks will be in the watch list.
REQ-8: A Log Out button must be available in the screener view to allow the user to log out and change users.
REQ-9: When the user presses the Log Out button then the Login view must be shown, and the user logged out of the system.
REQ-10: Whenever the user applies a filter or modifies the watched stock list, the users settings must be recorded in order to load the settings on next login.

4.2 Filtered Stock Settings

4.2.1 Description and Priority

A logged in user must have a view to set the filter settings to be applied to the master list of stocks. This feature brings the highest priority for the user with a low cost and low risk for development. The settings will be saved per user and allow saving/loading favorite settings as a file. Implementation and design should be straightforward with no unknowns. The actual Filtered Stock View will be discussed in feature 4.3

4.2.2 Stimulus/Response Sequences

Precondition: User has logged in, see Feature 4.1

1. User can interact with a pane on the left side of the application for settings

2. User enters a min/max range for the stock price

3. User enters a unit for the market cap. Either millions or billions

4. User enters a min/max market cap value

5. User enters a min/max volume

6. User Presses Apply

7. Application updates the view of filtered stocks to match the settings.

8. Application saves the users settings to disk for the next time the user logs in.

9. User can optionally press the Save As Favorite to store the settings to disk as a favorite.

10. User can optionally press Load Favorite button to select a favorite setting from disk and have it applied.

4.2.3 Functional Requirements

REQ-1: User has ability to enter a min price value
REQ-2: User has ability to enter a max price value
REQ-3: User has ability to enter a unit of either million or billion for market cap range
REQ-4: User has ability to enter a min market cap value
REQ-5: User has ability to enter a max market cap value
REQ-6: User has ability to enter a min volume value
REQ-7: User has ability to enter a max volume value
REQ-8: User has ability to press an Apply button
REQ-9: User has ability to press a Save as Favorite button
REQ-10: User has the ability to press a Load Favorite button.
REQ-11: When user presses Apply the filtered stocks update to the ranges entered in the settings.
REQ-12: When the user logs into the app a second time, the setting are equal to the last apply.
REQ-13: When the user presses Save as Favorite they are presented with a file save dialog and can select a file name and path to save to.
REQ-14: When the user presses Load Favorite they are presented with a Load File dialog.
REQ-15: When the user selects a file to load through the Load Favorite button, the setting must update to the settings in the file and apply.

4.3 Filtered Stock View

4.3.1 Description and Priority

A logged in user must have a view to see the filtered list of stocks. This feature brings the highest priority for the user with a high cost and high risk for development. There must be implementation done to get current stock prices which carries an unknown in which mechanism to use. It also requires the Filtered Settings to be completed (4.2)

4.3.2 Stimulus/Response Sequences

Precondition: User has logged in, see Feature 4.1

1. User can view a list of filtered stocks

2. The list must have stocks that fit the filter criteria from the filter settings (4.2)

3. User must be able to see the following columns: Stock, Last Price, Volume (Million), Change %, Watch

4. User can click on the header of a column to sort.

4. User can checkbox the stock in the watch column (Feature 4.4 will discuss further)

5. When a stock updates the user should see the change percent value as red or green if negative or positive.

4.3.3 Functional Requirements

REQ-1: User can see a list of the filtered stocks
REQ-2: User has access to the following columns: Stock, Last Price, Volume (Million). Change %, Watch
REQ-3: User can sort a column by clicking on the column header.
REQ-4: The displayed stocks must be within the criteria of the filtered settings
REQ-5: User has ability to select a watch arrow in the table to mark the stock as watching. See feature 4.4 on the watch feature.
REQ-6: Stocks must update in less than 5-minute intervals.

4.4 Watched Stock View

4.4.1 Description and Priority

A logged in user must have a view to see stocks marked as watched. These stocks always appear on the right pane no matter what the filter settings are (4.2). This feature brings a medium priority for the user with a medium cost and low risk for development. If the feature is not complete it will not hold up the release of the application. If time permits, this added feature will be positive for the user and improve the feature set. It requires features 4.1 and 4.3 to be completed. It leverages the user settings to store the watch list and the 4.3 stock list to update the values. Once 4.1 and 4.3 are complete, the work should be low cost and low risk.

4.4.2 Stimulus/Response Sequences

Precondition: User has logged in and marked some stocks as watched, see Feature 4.1 and 4.3

1. User can mark a stock as watched from feature 4.3

2. The checked stock is added to a list of watched stocks that appears on the right side of the application.

3. Watched stocks are saved to the user settings and remembered when users log in.

4. Watched stocks are only removed from the list when unchecked in the watched stock view.

5. Watched stocks update within 1 minute and do not get removed due to filter changes.

4.4.3 Functional Requirements

REQ-1: User can see a list of watched stocks
REQ-2: Any stock marked as watch through feature 4.3 appear in the list of watched stocks
REQ-3: Watched stocks must refresh within 5 minutes
REQ-4: Watched stocks can be removed from the watch list by clicking the remove button in the Remove column in the watched table
REQ-5: Each time a stock is marked or unmarked as watched, the users settings must update so across logins the watched list is persisted.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Requirements for user login:

1. User login button must take user to “logged in” UI in five seconds or less

Requirements for Screener Stock filtering:

1. Stock screener UI buttons must cause action within 2 seconds
2. Stocks must update their values within 5 minutes
3. Logout button must log user out within 5 seconds

Requirements for background stock information:

1. Stocks must be updated every 5 minutes

5.2 Security Requirements

The following security controls and tests should be implemented/verified:
Input Validation

1. CWE-20: Improper Input Validation
2. CWE-89: SQL Injection (If required)
3. CWE-91: XML Injection
4. CWE-78: OS Command Injection
5. CWE-120: Buffer Overflow
6. CWE-22: Path Traversal
7. CWE-807: Reliance on Untrusted Inputs
8. CWE-131: Incorrect Calculation of Buffer Size
9. CWE-134: Uncontrolled Format String
10. CWE-190: Integer Overflow or Wraparound
11. CWE-676: Use of Potentially Dangerous Function

Authentication and Login Process

1. CWE-287: Improper Authentication
2. CWE-306: Missing Authentication for Critical Function
3. CWE-307: Improper Restriction of Excessive Authentication
4. CWE-798: Use of Hard-Coded Credentials

Authorization and Control

1. CWE-22: Path Traversal
2. CWE-250: Execution with Unnecessary Privileges
3. CWE-829: Inclusion of Functionality from Untrusted Control Sphere
4. CWE-862: Missing Authorization
5. CWE-863: Incorrect Authorization
6. CWE-732: Incorrect Permission Assignment for Critical Resource

5.3 Software Quality Attributes

Design Qualities

1. Conceptual Integrity-application should define consistency and coherence of the overall design

2. Maintainability-application should survive changes with relative ease

Run-time Qualities

1. Interoperability-application should communicate with external parties and the host system as seamlessly as possible

2. Performance-latency and responsiveness of the application should be kept as low as possible

3. Security testing should ensure prevention of malicious and accidental actions as well as preventing disclosure and loss of information.

User Qualities

1. Usability-application should meet the requirements of the user and consumer by being intuitive, and result in an overall user experience

   *Reference: https://msdn.microsoft.com/en-us/library/ee658094.aspx

5.4 Business Rules

1. There will be no administrator for this application

2. User roles should be separate ensuring that each user cannot see any information pertaining to another user

3. Generally this application will function as an individual user application and not on an enterprise level