SCIT

School of Computing and Information Technology

Faculty of Engineering & Information Sciences

CSCI321 - Project

Project Topic: FYP-20-S3-09

Topic Code: Financial Market Simulator

Project Requirements Documentation

4 July 2020

UOW ID:	Name:	Email:
6096438	G ARAVINTHAN	ga001@mymail.sim.edu.sg
6648265	JOHAN SETIAWAN	js004@mymail.sim.edu.sg
6355444	TEO MING HONG	mhteo004@mymail.sim.edu.sg
6354567	FOO LI LING	llfoo002@mymail.sim.edu.sg
6344926	LIM JIA MIN KIMBERLY	jmklim002@mymail.sim.edu.sg

Document Control

Title: Software Requirements Specification

Document Name: Application Software Requirements_ FYP-20-S3-09, Version 1.0

Information	
Document ID	FMS 2020/001
Document Owner	Teo Ming Hong
Issue date	13/07/2020
Last saved date	28/07/2020
File name	FYP-20-S2-17_Application_Software_Requirements.docx

Version	**Issue Date**	**Changes**
1.0	13/07/2020	Initial document creation

Authorization

The Product Design Leader has authorized the project and has agreed that it be initialized

Product Design Leader	**Customer**
Name: Aravin Ganeshan	Name: Sujati Sastro
Position: Product Design Leader	Position:
Date:__/__/____	Date:__/__/____

Distribution List

Name	Title/Role	Where
Sujati Sastro	Project Supervisor	SIM-UOW
Tian Soh Lui	Project Assessor	SIM-UOW
G Aravinthan	Project Leader	SIM-UOW
Johan Setiawan	Team Member	SIM-UOW
Teo Ming Hong	Team Member	SIM-UOW
Foo Li Ling	Team Member	SIM-UOW
Lim Jia Min Kimberly	Team Member	SIM-UOW

Content Page

1. Introduction

1.1 Purpose

The purpose of this document is to present the detailed requirements of MarketForStarters, a web application that simulates a trading platform for the stock market. It will explain the features required of the system, the interfaces of the system, what the system will do, and how the system will react to external stimuli. This document covers the description of most of the software features. This document is intended for software developers and the product design team of the system and will be proposed to the sponsor for her approval.

1.2 Statement of Scope of Software Development

This project will develop application software that provide beginners to the stock market a stock market simulator called MarketForStarters. This application will have basic financial market simulator features such as Stock Price Simulation, Virtual Trading Platform, Personal Trading Portfolio and Market Watchlist. Furthermore, this simulator will have added unique features of Beginner Friendly Interface,Education on the Impacts of Major Historical Events, and Key Financial information that will allow beginners to make an informed decision when buying or selling stocks. These unique features will position the application to compete with existing players, such as Wall Street Survivor, HowTheMarketWorks and MarketWatch

1.3 Intended Audience

This document is written for the product design team, application developers, users and testers. This is to learn about the requirements.

1. Project Manager
   Project Manager should be regularly updated on the progress and changes of this document, to be informed about the main components and progression of the application development to make more informed decisions.

2. Application Developers
   Application Developers should refer to this document to have a clear understanding of the main goals and features of the system before adding new features or making changes.

3. Application Testers
   Application Testers should refer back to this document to design test plans based on the application requirements and execute functional tests to validate the proposed functions.

4. Application Users
   Application Users use this document as a guiding material on how the business requirements are translated into the proposed functions.

2. Product Requirements

2.1 User Classes and Characteristics

There will be two types of users allowed in this project, traders and administrator. Traders are allowed to access the trading platform complete with its functions and all the learning materials available. The administrators are able to access the traders' privileges and back-end of the system.

2.1.1 Traders

Functions allowed to be performed by the traders are:

- Create, manage and delete personal account
- Buy and sell stocks
- Create and manage personal watchlist
- Manage trading portfolio
- Access educational material

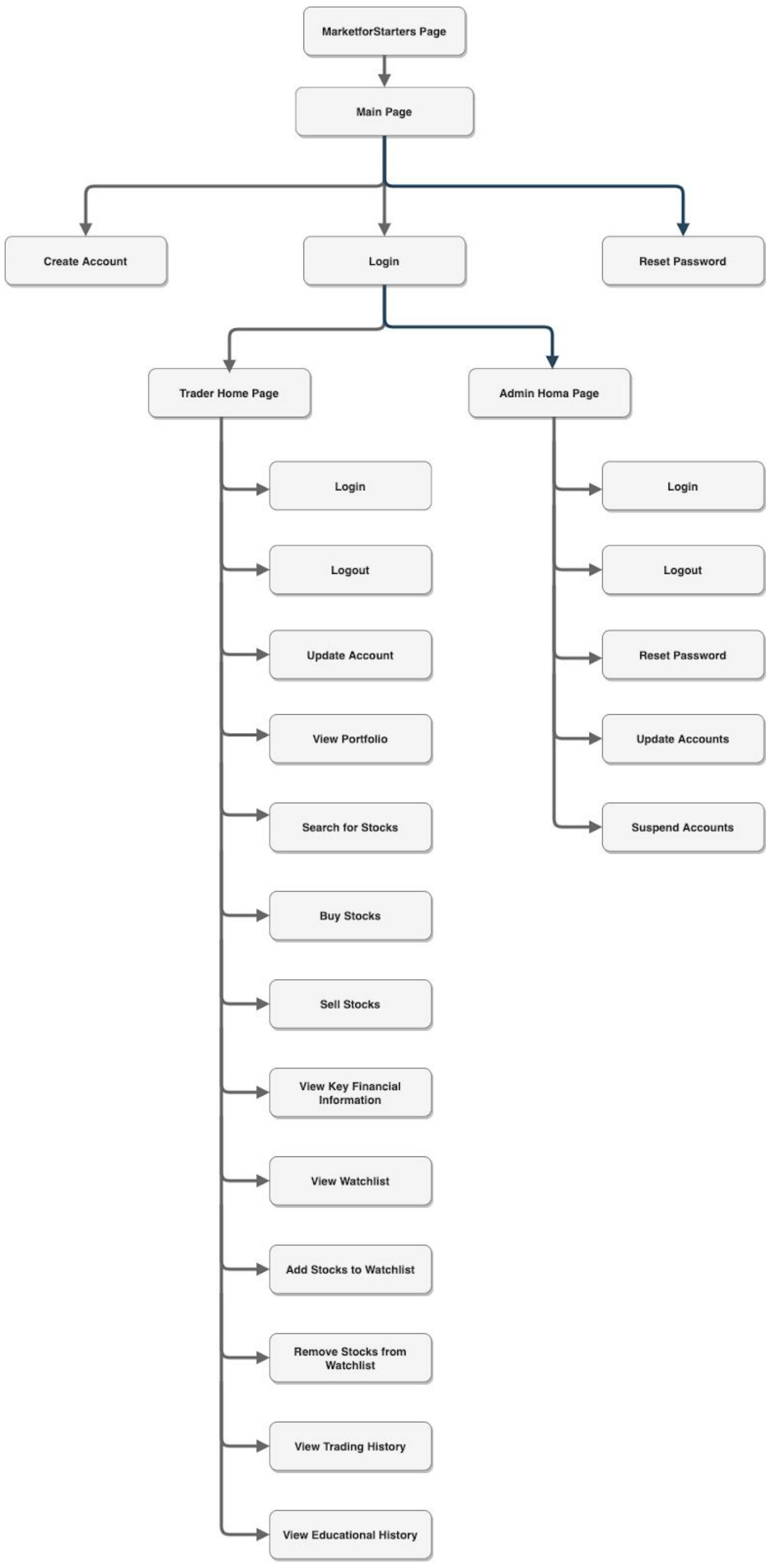
2.1.2 Administrator

Functions allowed to be performed by the administrators are:

- Manage user accounts
- Maintain database

2.2 Overview of Product Structure

The overall product structure for the above user classes is shown below:



2.3 Product Features

Our simulation will follow the Singapore Exchange (SGX) operation time which allows local users to transact in a normal working operation time.

2.3.1 Stock Price Simulation

For stock price simulation, it is required to have real-time updated data with a clear and robust data representation so that the simulation can be interpreted correctly and accurately by the end users.

2.3.2 Virtual Trading Platform

On the trading platform, responsiveness and reliability are the utmost qualities to have. Every buy and sell has to be executed in a very fast and accurate manner so that the calculation of total virtual net worth of a trader is as accurate as possible. The trader will be very satisfied if there is no lagging in every transaction done as every second matters for the stock pricing.

2.3.3 Personal Trading Portfolio

Every trader's trading portfolio is required to display all the stocks that he/she has with the right amount, right ordered price and some other information.

2.3.4 Market Watchlist

A good market watchlist is required to provide information accurately regarding the stocks and from every trader so that every trader can fetch the monitor the right stocks.

2.3.5 Beginner Friendly Interface

To have a beginner friendly interface, means the application to be designed in such a way that is less overwhelming with less informative tabs and page components

2.3.6 Education on the Impacts of Major Historical Events

On the educational page, it is required to have all the information as accurate and updated as possible.

2.3.7 Key Financial Information

Key financial information is required to be as reliable and updated as possible so that every trader can make the right and informed decision on every transaction.

3. Requirements

3.1 Functional Requirements

Here is described the functions that are required on every stakeholder's end on the application.

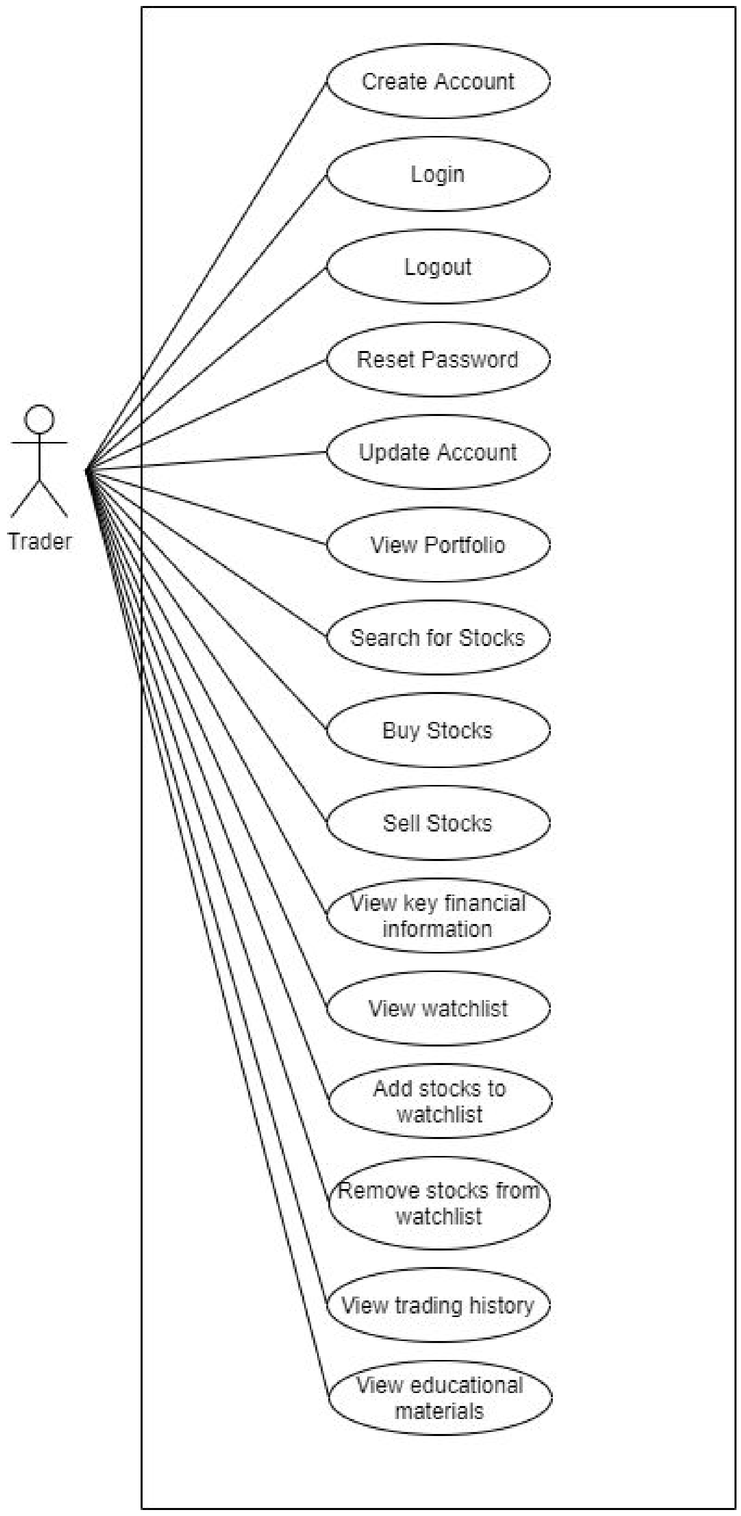
3.1.1 Functional Requirement Descriptions

Functional Requirement ID	Functional Requirement Description	Priority
FR-T1	As a trader, I want to create an account so I can use the web application	High
FR-T2	As a trader, I want to be able to reset my account password so that I use the application again	Medium
FR-T3	As a trader, I want to update my account so that I can update my latest credentials and information	Medium
FR-T4	As a trader, I want to log in so I can use the application	High
FR-T5	As a trader, I want to log out so I can stop using the application	High
FR-T6	As a trader, I want to view my portfolio so I can manage my trades and view the performance of my stocks	High
FR-T7	As a trader, I want to search for the stock of my choice to view its details	Medium
FR-T8	As a trader, I want to buy stocks and decide on the quantity	High
FR-T9	As a trader, I want to sell stocks and decide on the quantity	High
FR-T10	As a trader I want to view key financial information so that I can make an informed decision on every stock that I want to transact on	High
FR-T11	As a trader, I want to view my watchlist to monitor closely my favourite stocks	High

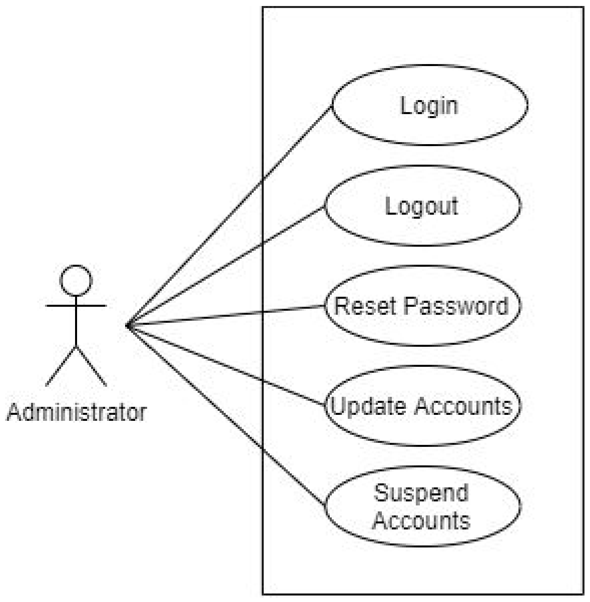
Functional Requirement ID	**Functional Requirement Description**	**Priority**
FR-T12	As a trader, I want to add stocks to my watchlist so that I can monitor closely my favourite stocks at that moment	High
FR-T13	As a trader, I want to remove stocks from my watchlist so that I can stop monitoring irrelevant stocks	High
FR-T14	As a trader, I want to view my trading history so I can track my equity and performance well	High
FR-T15	As a trader, I want to view some educational materials so that I can keep up with the trading world and strengthening my knowledge	High
FR-A1	As an administrator, I want to log in to the application to manage the application	High
FR-A2	As an administrator, I want to log out to end my session	High
FR-A3	As an administrator, I want to reset my account password so I can manage the application again	High
FR-A4	As an administrator, I want to update traders' account to its newest changes	Medium
FR-A5	As an administrator, I want to suspend any accounts that have been detected for serious offenses	High

3.1.2 Use Case Diagram

3.1.2.1 Trader



3.1.2.2 Administrator



Use Case description is documented at Appendix A and Detailed Process Flow is at Appendix B.

3.2 Non- Functional Requirements

3.2.1 Safety Requirements

In the case of database failure which would result in any amount of data loss, there is backed up data on a different set of storage disks. Transactions done up to database failure will be logged and redone to the new set of databases.

3.2.2 Security Requirements

Database Security

Database security is important, especially on protecting user credentials so that their credentials will not fall on the wrong hand. Thus, choosing the right vendor for the database system is crucial.

Web Application Security

The web application itself should be secure enough to cover up common web application vulnerabilities such as SQL injection, Cross-Site Scripting, Broken Session Management, etc.

3.2.3 Performance Requirements

A certain degree of hardware capabilities are required to accommodate the speed of updating the database and the web application live. Not only view updating, accommodating possibly a hundreds of thousands to a million traders's account requires a good server. Besides space accommodation, a good amount of server bandwidth is required so that every transaction from the future traders can be executed smoothly.

On the performance requirement, there is a need to touch on concurrent processes handling capacity. It will be determined by performing load testing on the server. The server will be upgraded accordingly afterwards.

When discussing about web application, on the user-end, to have the full experience, they have to have a good internet speed and connection, which is a factor that can't be controlled by the developer

3.2.4 Software Quality Attributes

- **Adaptability**
  The web application should be able to be used in any kind of web browsers and any devices available.

- **Availability**
  The web application must be available to everyone at a reasonable amount of rate.

- **Correctness**
  The web application must present the data correctly and error-free.

- **Maintainability**
  The web application shall be built and designed in such a way that if there are new team members of developers, they would not ask too many questions on how to proceed on maintaining and developing the application.

- **Usability**
  The web application must be easy to navigate, efficient and visually acceptable to all types of customers.

3.2.5 Normalisation

Normalisation is a process of database design that eliminates data redundancy and undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization rules separate larger tables into smaller tables and link them using relationships.

Normalisation is accomplished in the purpose of reducing information redundancy and increasing efficiency in storing data in the database, which would lead to an overall lighter system to run. Normalisation should be done after we finalize the first phase of the database development and if we find any redundancy or anomalies.

Progress flow diagram is documented at Appendix C.

Appendix A

Use Case Descriptions for the functional requirements

Function Name: Create Account	**ID:** FR-T1
Stakeholders and Goals: Trader - wants to create an account	
Actors: Trader	
Pre-condition: The trader must be on the homepage of the web application	
Post-condition: The trader has successfully created an account	
Trigger: The trader wants to use the application	
Normal Flow: 1. The trader clicks on the “Sign Up” button 2. The application redirects the trader to the “Sign Up” page 3. The trader inputs all the required credentials and clicks on the “Register” button 4. The application validates the credentials 5. The application redirects the trader to the trader home page 6. End.	
Sub-flows:	
Alternative/Exceptional Flows: Existing account: The system informs the trader that the username or email address has been registered. Repeat the normal flow	

Function Name: Reset Password	**ID:** FR-T2
Stakeholders and Goals: Trader - wants to reset the account password	
Actors: Trader	
Pre-condition: The trader must be on the login page	
Post-condition: The trader has successfully reset his/her password	
Trigger: The trader wants to reset his/her account password	
Normal Flow: 1. The trader clicks on the “Forget Password” button on the login page 2. The application redirects the trader to the “Forget Password” page 3. The trader enters his/her email address 4. The trader clicks on the “Submit” button 5. The application validates the email address from the database 6. The application sends a link to the trader’s email address to redirect them to the “Reset Password” page 7. The trader clicks on the link and gets redirected to the page 8. The trader inputs his/her new password 9. The trader clicks on the “Update” button 10. The application updates the user’s new password in the database 11. End	
Sub-flows: 1. The trader clicks on the “Back to login” button 2. The application redirects the trader to the login page 3. End	
Alternative/Exceptional Flows: Invalid email address: The application displays the error for invaliemail address. Repeat normal flow from step 3	

Function Name: Update Account	**ID:** FR-T3
Stakeholders and Goals: Trader - wants to update his/her account	
Actors: Trader	
Pre-condition: The trader must be logged in to his/her account	
Post-condition: The trader has successfully updated his/her account	
Trigger: The trader wants to update his/her profile	
Normal Flow: 1. The trader clicks on the “Account Setting” button 2. The application fetches his/her relevant credentials from the database 3. The trader makes relevant changes to his/her profile 4. The trader clicks on the “Update” button 5. The application updates the investor’s profile in the database 6. Ends	
Sub-flows: -	
Alternative/Exceptional Flows:-	

Function Name: Log In	**ID:** FR-T4
Stakeholders and Goals: Trader - wants to log in to the web application	
Actors: Trader	
Pre-condition: The trader must be on the homepage of the website and has created an account	
Post-condition: The trader has success logged in to the application	
Trigger: The trader wants to use the application	
Normal Flow: 1. The trader clicks on the “LogIn” button on the main page 2. The trader types his or her credentials 3. The trader clicks on the “login” button 4. The application checks the submitted credentials with the database 5. The application redirects the trader to the trader’s home page 6. Ends	
Sub-flows: -	
Alternative/Exceptional Flows: Invalid credentials: The application redirects the trader to the login page and prompts for credentials	

Function Name: Log Out	**ID:** FR-T5
Stakeholders and Goals: Trader - wants to log out to the web application	
Actors: Trader	
Pre-condition: The trader must be logged in to his/her account	
Post-condition: The trader has successfully logged out of the application	
Trigger: The trader wants to end his or her trading session	
Normal Flow: 1. The trader clicks on the “logout” button 2. The application ends the trader session 3. The application redirects the trader to the application’s home page 4. End	
Sub-flows: -	
Alternative/Exceptional Flows: -	

Function Name: View Portfolio	**ID:** FR-T6
Stakeholders and Goals: Trader - wants to view his/her portfolio	
Actors: Trader	
Pre-condition: 1. The trader must be logged in to his/her account 2. The trader is on the trader home page	
Post-condition: The trader has successfully viewed his/her portfolio	
Trigger: The trader wants to view his/her portfolio	
Normal Flow: 1. The trader clicks on the “Portfolio” button 2. The application shows the trader’s portfolio 3. End	
Sub-flows: -	
Alternative/Exceptional Flows: -	

Function Name: Search Stock	**ID:** FR-T7
Stakeholders and Goals: Trader - wants to search for stocks	
Actors: Trader	
Pre-condition: 1. The trader must be logged in to his/her account 2. The trader is on the trader home page	
Post-condition: The trader has successfully searched for the stocks	
Trigger: The trader wants to search for stock	
Normal Flow: 1. The trader types in the stock symbol on the search bar 2. The trader clicks on the “Search” button 3. The database fetches the stock data and displays the searched stock 4. End	
Sub-flows: -	
Alternative/Exceptional Flows: -	

Function Name: Buy Stock	**ID:** FR-T8
Stakeholders and Goals: Trader - wants to buy stocks	
Actors: Trader	
Pre-condition: 1. The trader must be logged in to his/her account 2. The trader is on the stock's page	
Post-condition: The trader has successfully purchased the stocks	
Trigger: The trader wants to buy stocks	
Normal Flow: 1. The trader select on the specific stock name 2. The application will redirect the trader to the stock's page 3. The trader specifies how many stocks to buy 4. The trader clicks "Buy" 5. The system updates the database 6. The application updates the user's data by adding the stock to the portfolio or increase the number of stock he/she has 7. The system redirects the trader to the stock detail and notify that the user has successfully purchased the stock 8. The system displays the updated data 9. End	
Sub-flows: 1. The trader goes to the "Watchlist" page 2. The trader clicks on the stock name on the watchlist 3. The system redirects the trader to the stock page	
Alternative/Exceptional Flows: Market closed: the application will not allow any transaction beyond market operation time. The application will display the respective error message Insufficient trading balance: the application will not allow the purchase. The application will display the respective error message	

Function Name: Sell Stock	**ID:** FR-T9
Stakeholders and Goals: Trader - wants to sell stocks	
Actors: Trader	
Pre-condition: 1. The trader must be logged in to his/her account 2. The trader is on the stock’s page	
Post-condition: The trader has successfully sold the stocks	
Trigger: The trader wants to sell his/her stocks	
Normal Flow: 1. The trader clicks on the stock 2. The application redirects the trader to the stock’s page 3. The trader specifies the amount of stock he/she will be selling 4. The trader clicks “Sell” 5. The system updates the database 6. The application updates the user’s data by subtracting the stock to the portfolio or increase the number of stock he/she has 7. The system redirects the trader to the stock detail and notify that the user has successfully sold the stock 8. The system displays the updated data 9. End	
Sub-flows: -	
Alternative/Exceptional Flows: Market closed: the application will not allow any transaction beyond market operation time. The application will display the respective error message Invalid number of stocks: The application will not allow the sale of stocks. The system will display the respective error message	

Function Name: View Key Financial Information	**ID:**FR-T10
Stakeholders and Goals: Trader - want to view key financial information	
Actors: Trader	
Pre-condition: 1. The trader must be logged in to his/her account 2. The trader is on the stock's page	
Post-condition: The trader has successfully view the key financial information of a certain stock	
Trigger: The trader want to get more information about the stock	
Normal Flow: 1. The trader clicks on the "View Key Financial Info" button 2. End	
Sub-flows: -	
Alternative/Exceptional Flows: -	

Function Name: View Watchlist	FR-T11
Stakeholders and Goals: Trader - wants to view personal watchlist	
Actors: Trader	
Pre-condition: 1. The trader must be logged in to his/her account 2. The trader must be on the trader home page	
Post-condition: The trader has successfully viewed his/her watchlist	
Trigger: The trader wants to monitor his/her favourite stocks	
Normal Flow: 1. The trader clicks on "Watchlist" tab 2. End	
Sub-flows: -	
Alternative/Exceptional Flows: -	

Function Name: Add Stocks on Watchlist	**ID:**FR-T12
Stakeholders and Goals: Trader - wants to add stock on his/her watchlist	
Actors: Trader	
Pre-condition: 1. The trader must be logged in to his/her account 2. The trader is on “Watchlist” page	
Post-condition: The trader has successfully added stock(s) on his/her watchlist	
Trigger: The trader wants to monitor a specific stock(s).	
Normal Flow: 1. The trader clicks on “Add stock(s)” button 2. The trader searches for the specific stock(s) 3. The trader clicks on the dimmed star button on the right hand corner on the specific stock name 4. End	
Sub-flows: -	
Alternative/Exceptional Flows: 1. The trader is on a stock’s page 2. The trader clicks on the dimmed star button on the top right hand side of the page	

Function Name: Remove Stocks on Watchlist	**ID:**FR-T13
Stakeholders and Goals: Trader - wants to add stock on his/her watchlist	
Actors: Trader	
Pre-condition: 1. The trader must be logged in to his/her account 2. The trader is on “Watchlist” page	
Post-condition: The trader has successfully removed stock(s) on his/her watchlist	
Trigger: The trader wants to stop monitoring a specific stock(s).	
Normal Flow: 1. The trader clicks on lighted star button 2. End	
Sub-flows: -	
Alternative/Exceptional Flows: 1. The trader is on a stock’s page 2. The trader clicks on the lighted star button on the top right hand side of the page	

Function Name: View Trading History	FR-T14
Stakeholders and Goals: Trader - wants to see his/her trading history	
Actors: Trader	
Pre-condition: 1. The trader must be logged in to his/her account 2. The trader must be on the trader home page	
Post-condition: The trader views his/her trading history	
Trigger: The trader wants to track his/her buys and sells	
Normal Flow: 1. The trader clicks on the "Trading History" tab 2. End	
Sub-flows: -	
Alternative/Exceptional Flows: -	

Function Name: View Educational Materials	**ID:**FR-T15
Stakeholders and Goals: Trader - wants to view some educational materials	
Actors: Trader	
Pre-condition: The trader must be on the trader home page	
Post-condition: The trader views provided educational materials	
Trigger: The trader wants to learn something new related to the stock market	
Normal Flow: 1. The trader clicks on "Education" tab 2. The trader clicks on any article on the list available 3. End	
Sub-flows: -	
Alternative/Exceptional Flows: -	

Function Name: Log In	FR-A1
Stakeholders and Goals: Administrator - wants to log in to the web application	
Actors: Administrator	
Pre-condition: The administrator must be on the homepage of the web application	
Post-condition: The administrator has successfully logged in to the web application	
Trigger: The administrator wants to use the application	
Normal Flow: 1. The administrator clicks on the "LogIn" button on the main page 2. The administrator types his or her credentials 3. The administrator clicks on the "login" button 4. The application checks the submitted credentials with the database 5. The application redirects the administrator to the administrator home page 6. Ends	
Sub-flows: -	
Alternative/Exceptional Flows: -	

Function Name: Log Out	**ID:**FR-A2
Stakeholders and Goals: Administrator - wants to log out of the web application	
Actors: Administrator	
Pre-condition: The administrator must be logged in to his/her account	
Post-condition: The administrator has successfully logged out of the application	
Trigger: The administrator wants to end his or her session	
Normal Flow: 1. The administrator clicks on the "logout" button 2. The application ends the administrator session 3. The application redirects the administrator to the application's home page 4. End	
Sub-flows: -	
Alternative/Exceptional Flows: -	

Function Name: Reset Password	**ID:**FR-A3
Stakeholders and Goals: Administrator - wants to reset the account password	
Actors: Administrator	
Pre-condition: The administrator must be on the login page	
Post-condition: The administrator has successfully reset his/her password	
Trigger: The administrator wants to reset his/her account password	
Normal Flow: 1. The administrator clicks on the “Forget Password” button on the login page 2. The application redirects the administrator to the “Forget Password” page 3. The administrator enters his/her email address 4. The administrator clicks on the “Submit” button 5. The application validates the email address from the database 6. The application sends a link to the administrator’s email address to redirect him to the “Reset Password” page 7. The administrator clicks on the link and gets redirected to the page 8. The administrator inputs his/her new password 9. The administrator clicks on the “Update” button 10. The application updates the administrator's new password in the database 11. End	
Sub-flows: 1. The administrator clicks on the “Back to login” button 2. The application redirects the administrator to the login page 3. End	
Alternative/Exceptional Flows: Invalid email address: The application displays the error for invalid email address. Repeat normal flow from step 3	

Function Name: Update Traders' Account	**ID:**FR-A4
Stakeholders and Goals: Administrator - wants to update a trader's account	
Actors: Administrator	
Pre-condition: 1. The administrator must be logged in to his/her account 2. The trader's account must exist in the database	
Post-condition: The administrator has successfully updated the trader's account	
Trigger: The administrator needs to help to update a user's account if they are unable to	
Normal Flow: 1. The administrator goes to the "Update User's Account" page 2. The application displays all the existing accounts 3. The administrator update the necessary data 4. The administrator clicks on "Update" button 5. The application validates the input fields 6. The application updates the account details in the database 7. End	
Sub-flows: -	
Alternative/Exceptional Flows: -	

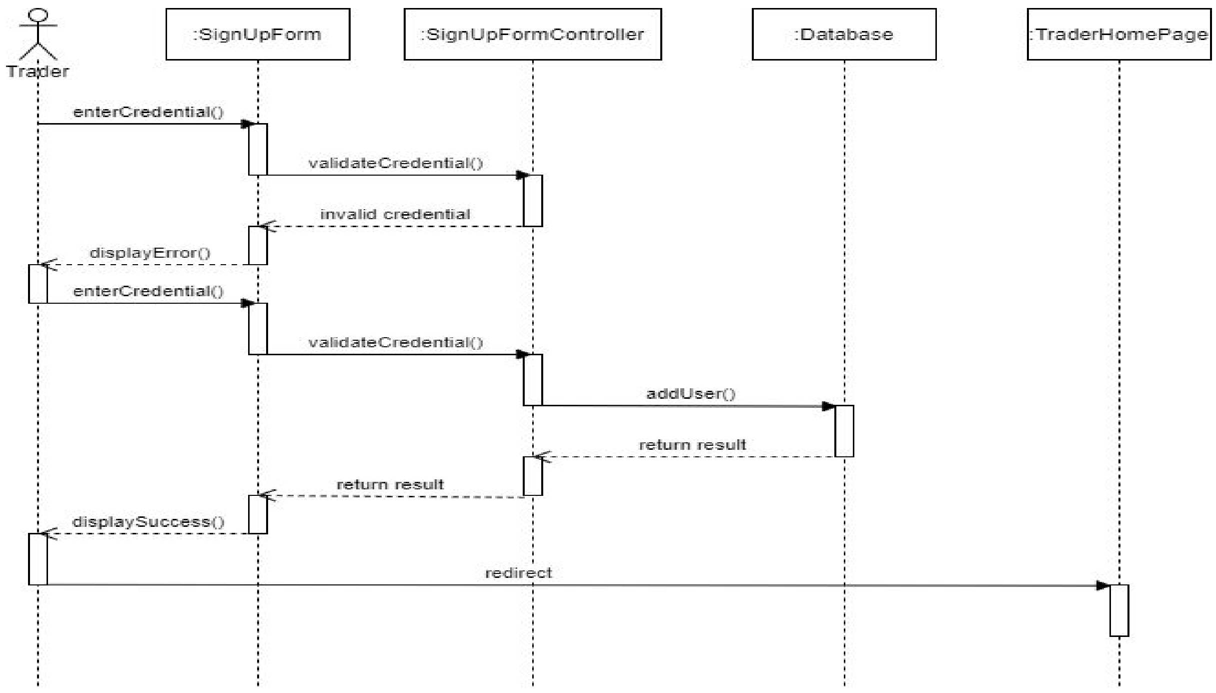
Function Name: Suspend Traders' Account	**ID:**FR-A5
Stakeholders and Goals: Administrator - wants to suspend a trader(s) account	
Actors: Administrator	
Pre-condition: 1. The administrator must be logged in to his/her account 2. The trader's account must exist in the database	
Post-condition: The administrator has successfully suspended the trader(s) account	
Trigger: Administrator wants to stop the trader's account which shows unusual behaviour	
Normal Flow: 1. The administrator goes to "Update User's Account" page 2. The application displays all the existing accounts 3. On the "status" column, the administrator changes the status from **Active** to **Suspended** 4. The administrator clicks on "Update" button 5. The application validates the input fields 6. The application updates the account details in the database 7. End	
Sub-flows: -	
Alternative/Exceptional Flows: -	

Appendix B

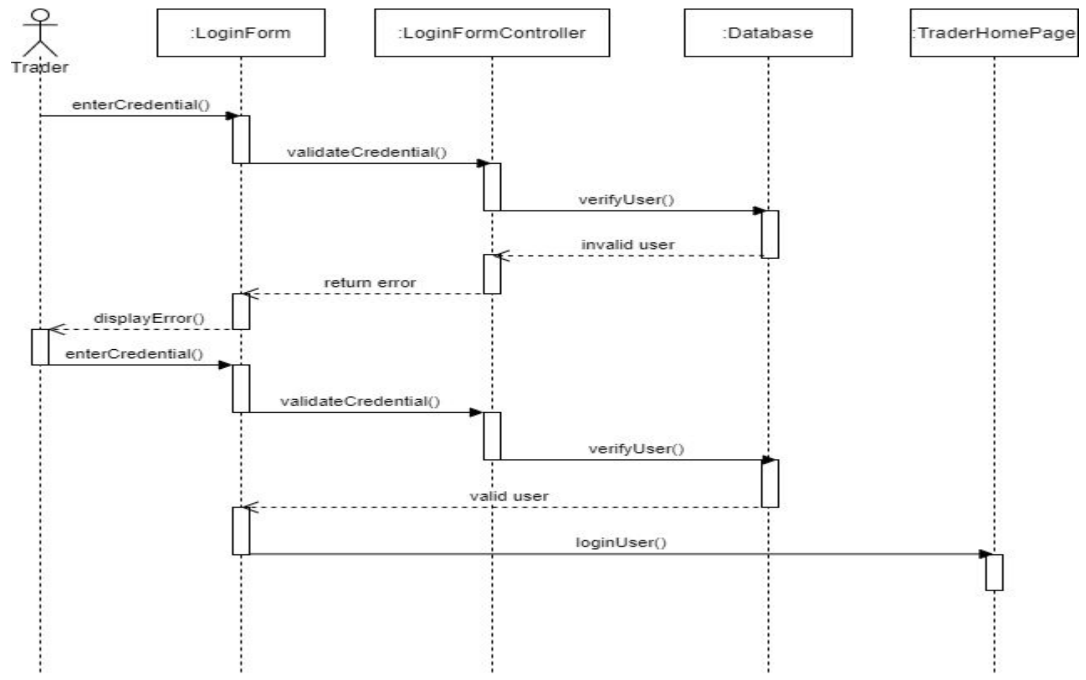
Sequence Diagrams

1. Trader

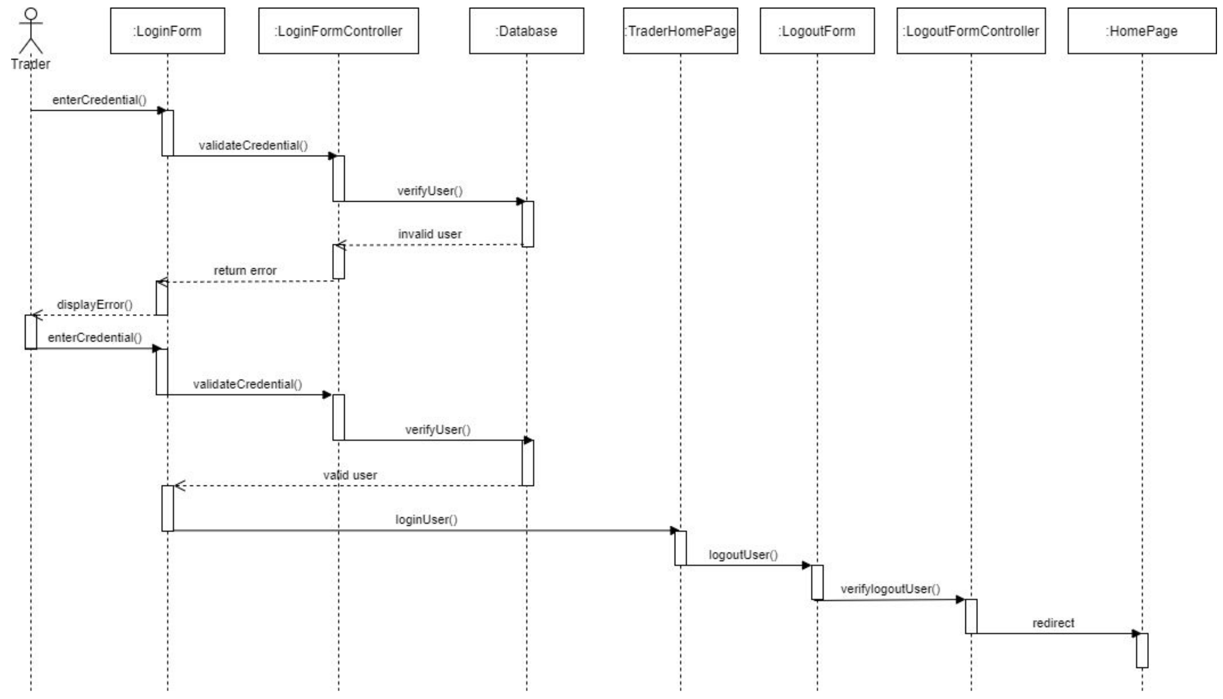
1(a) Create account



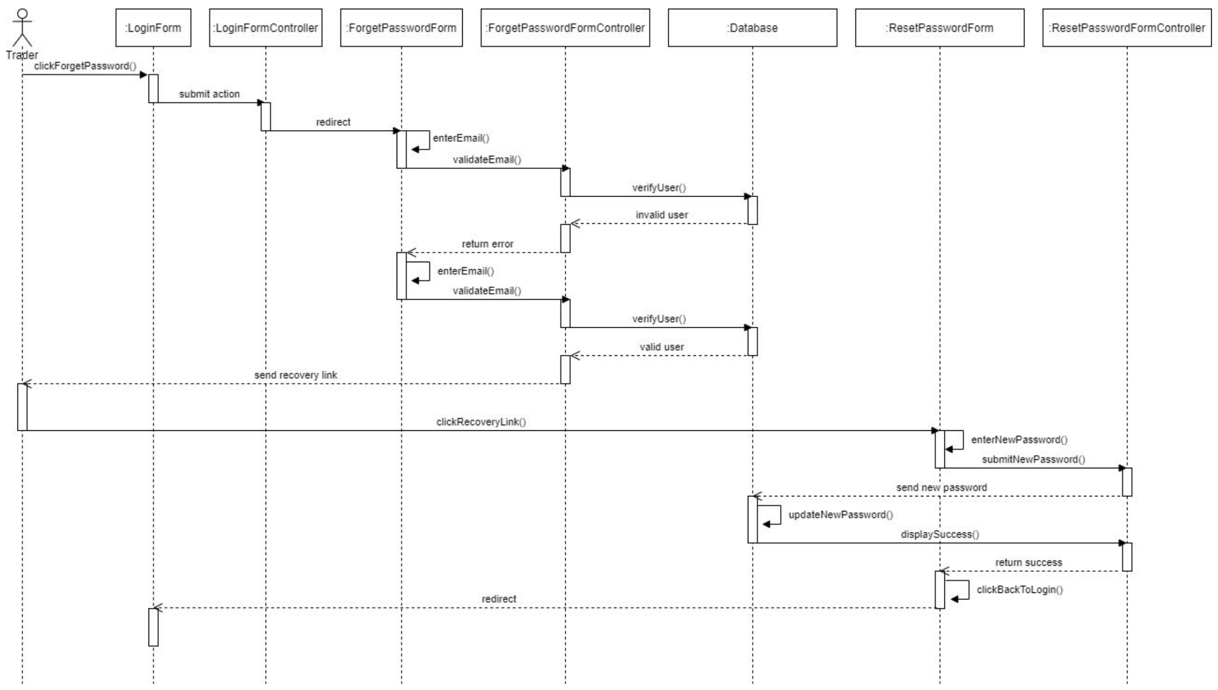
1(b) Login



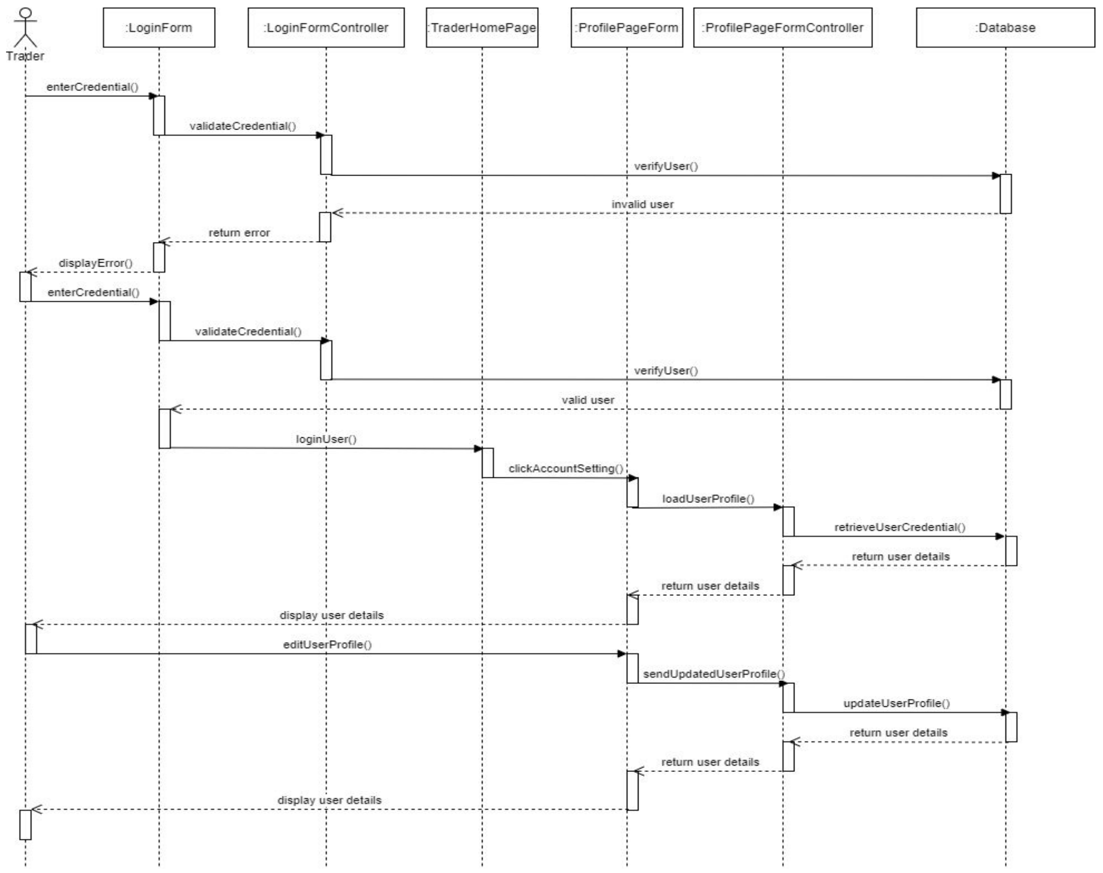
1(c) Logout



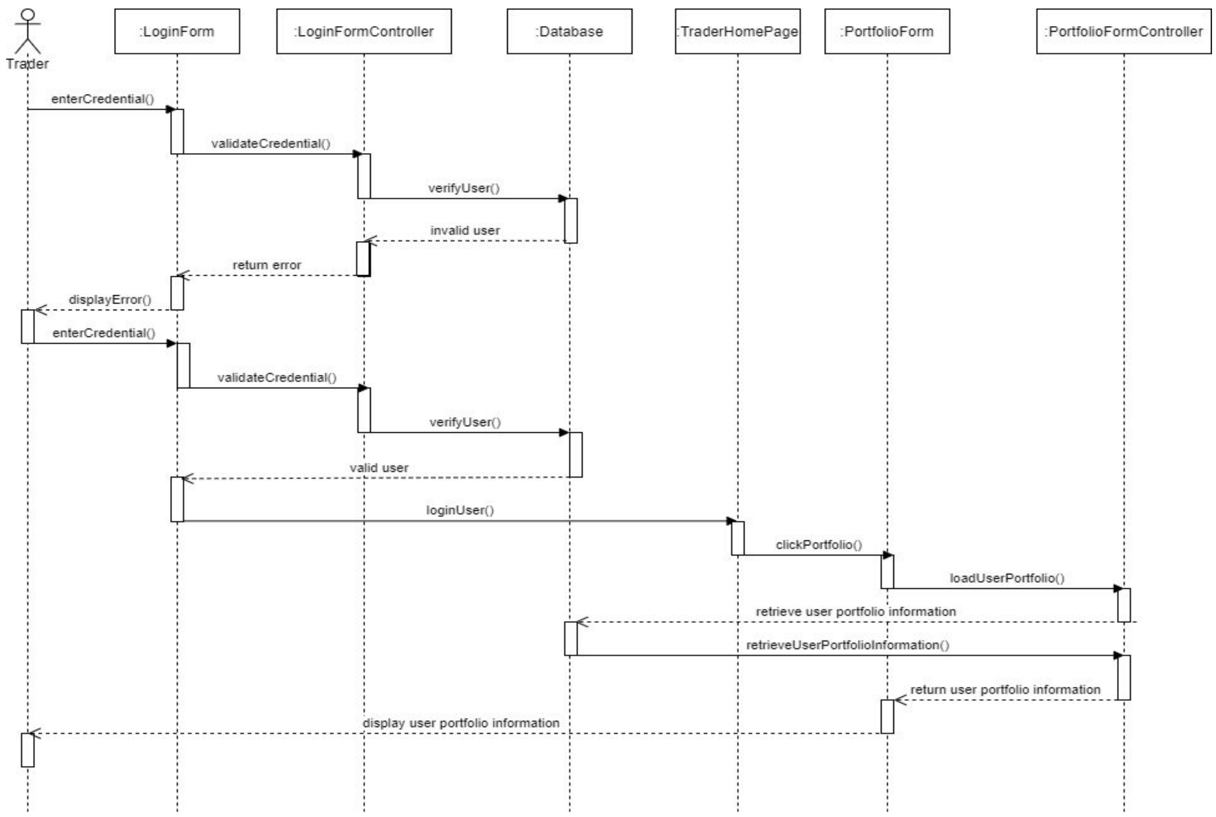
1(d) Reset Password



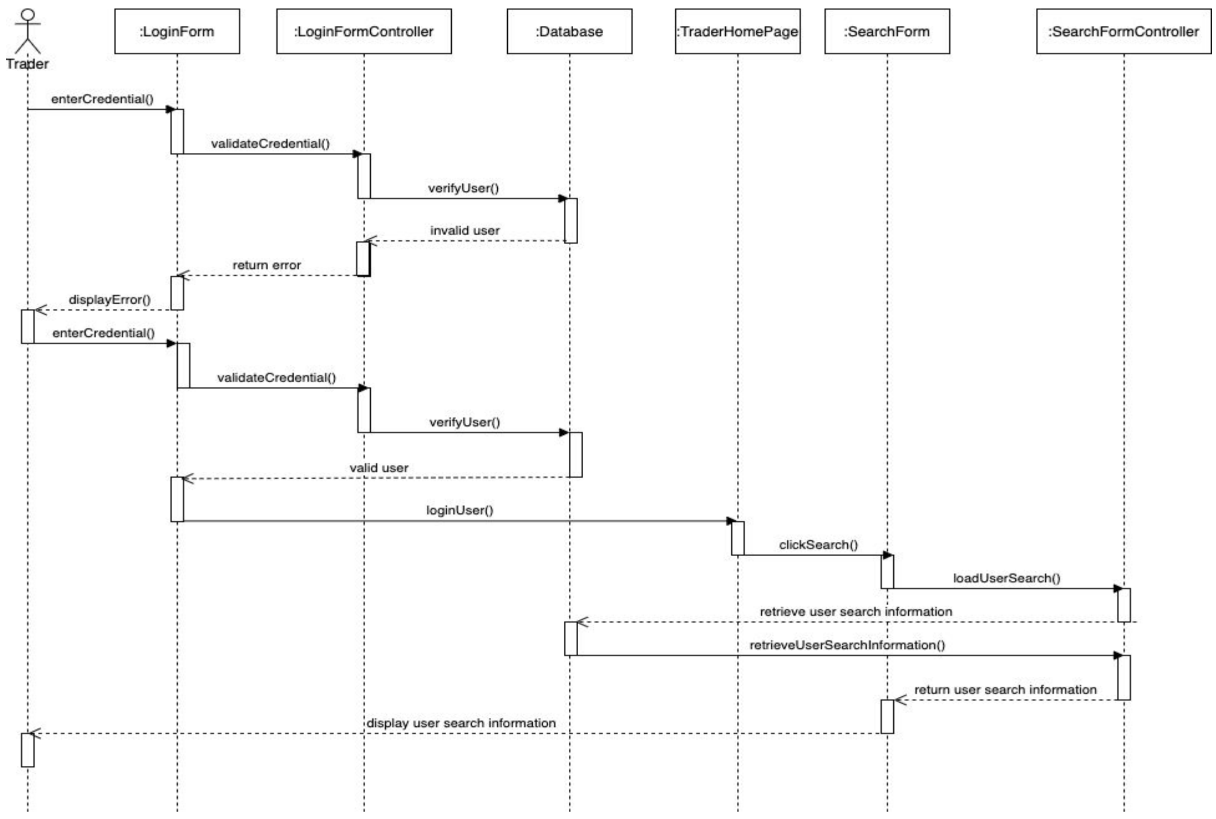
1(e) Update Account



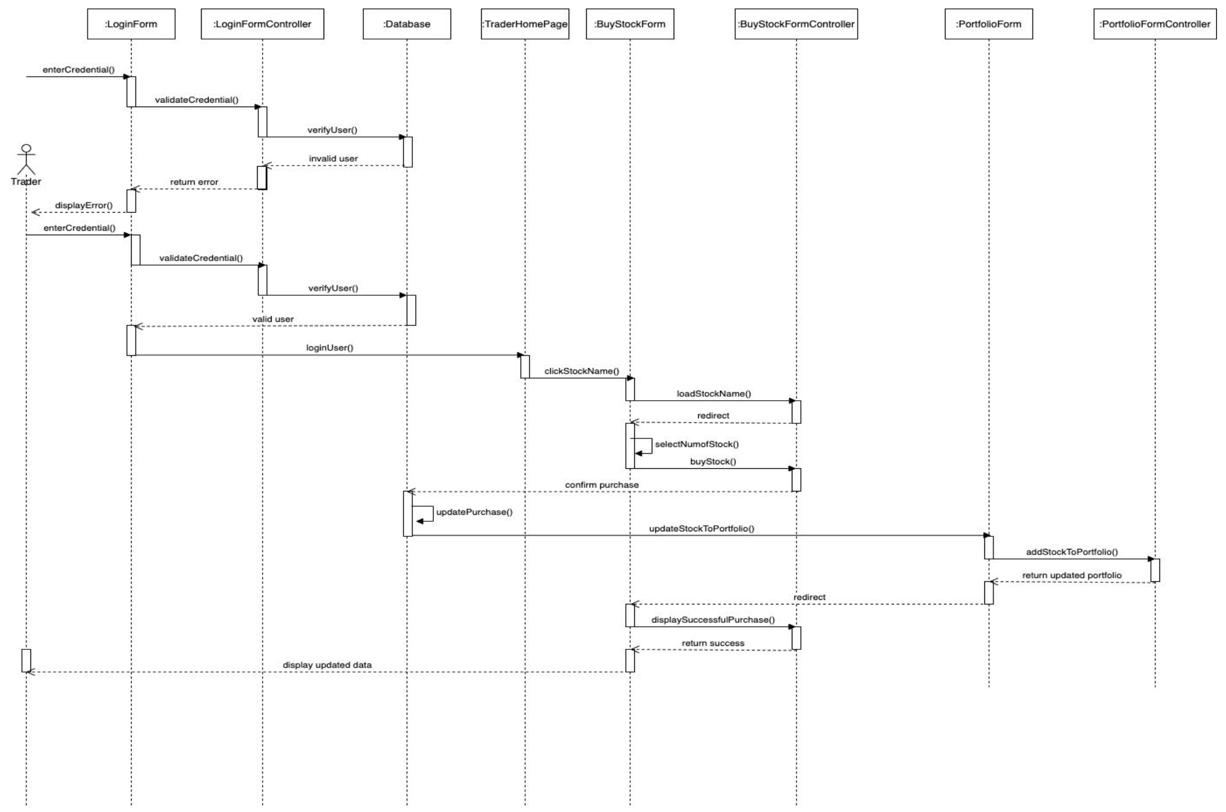
1(f) View Portfolio



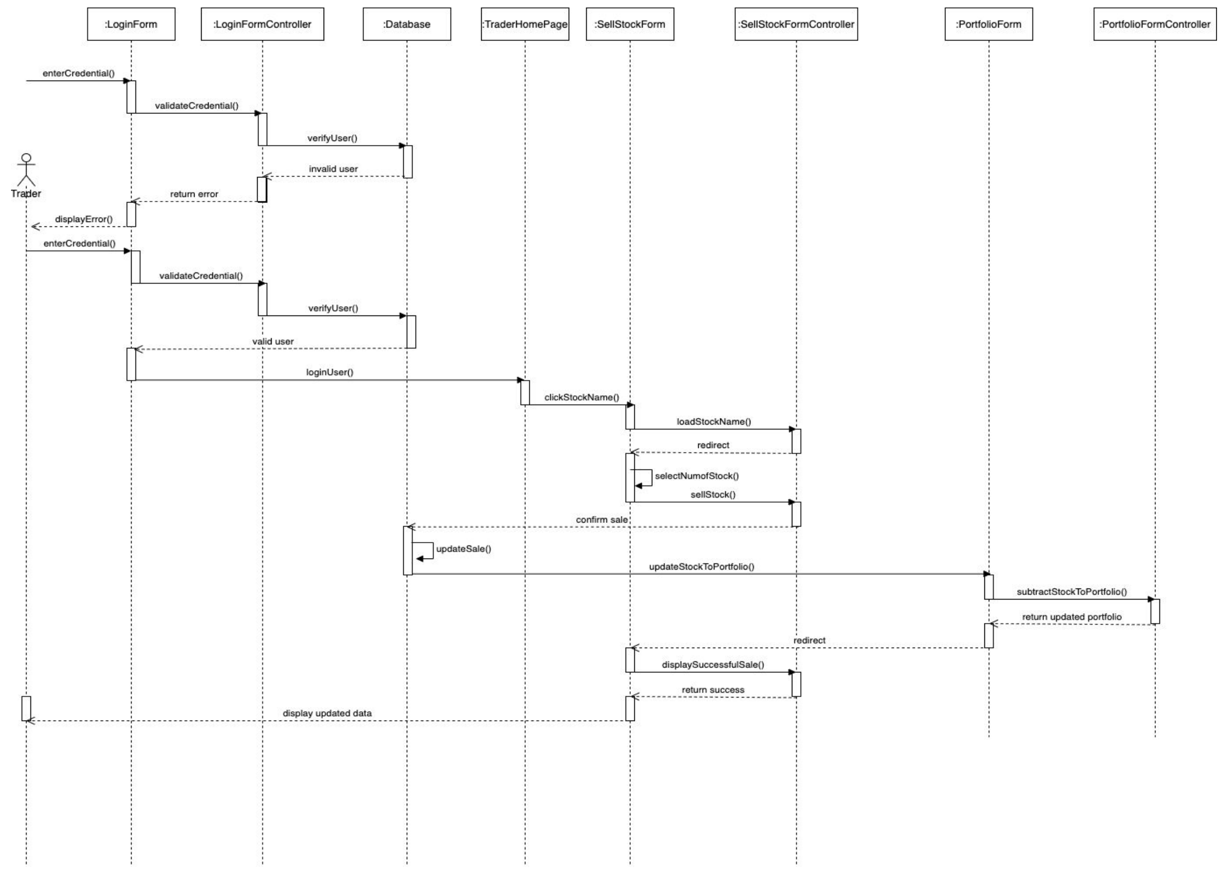
1(g) Search for Stocks



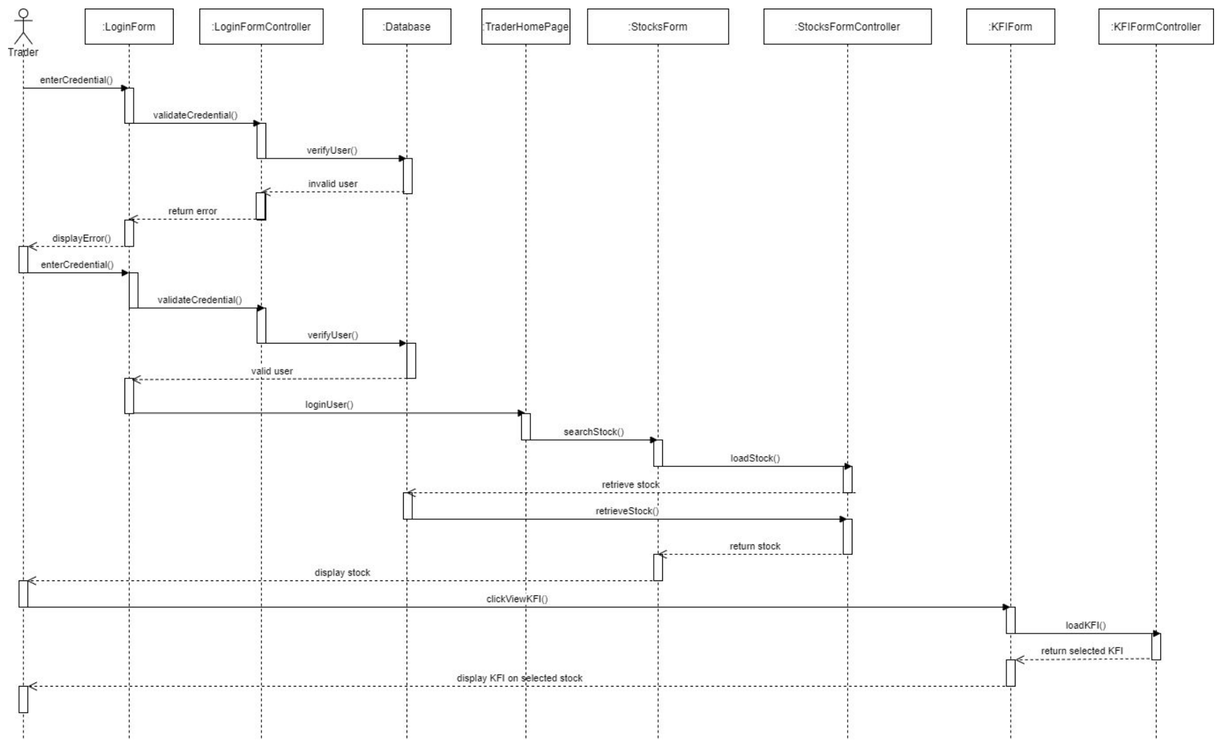
1(h) Buy Stocks



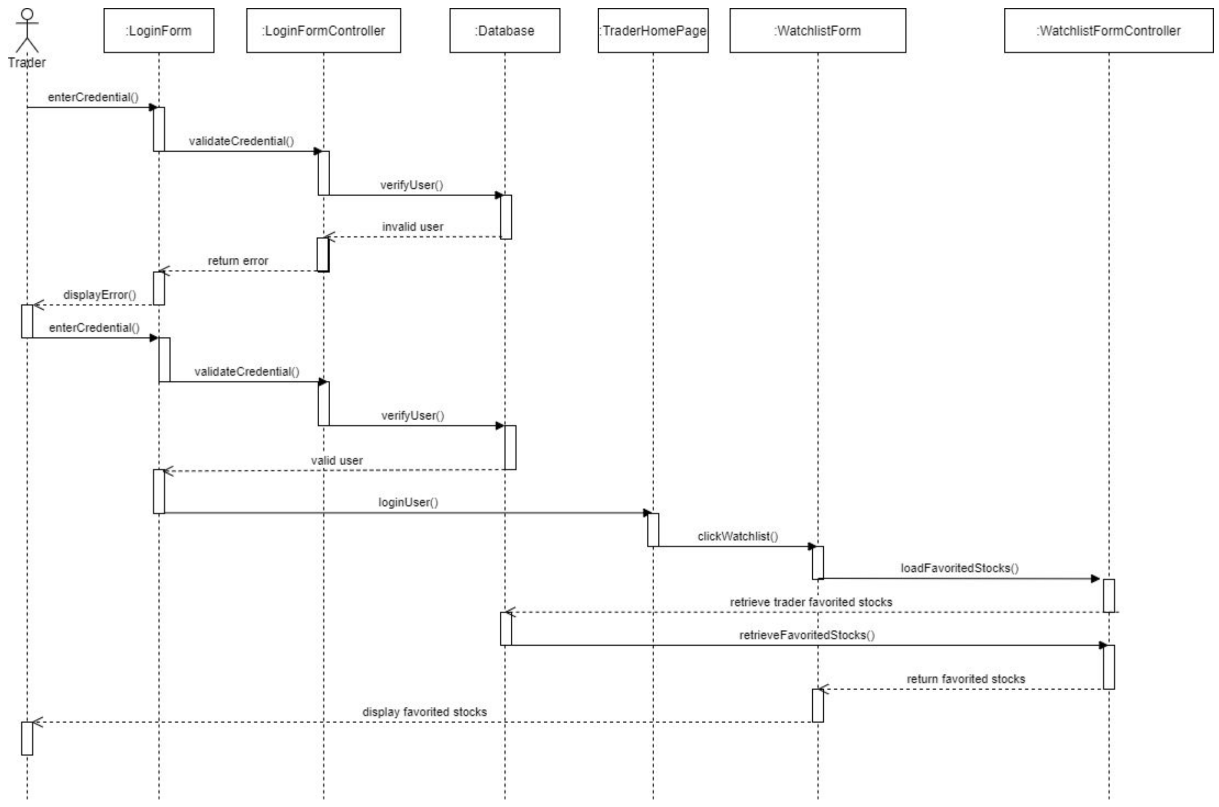
1(i) Sell Stocks



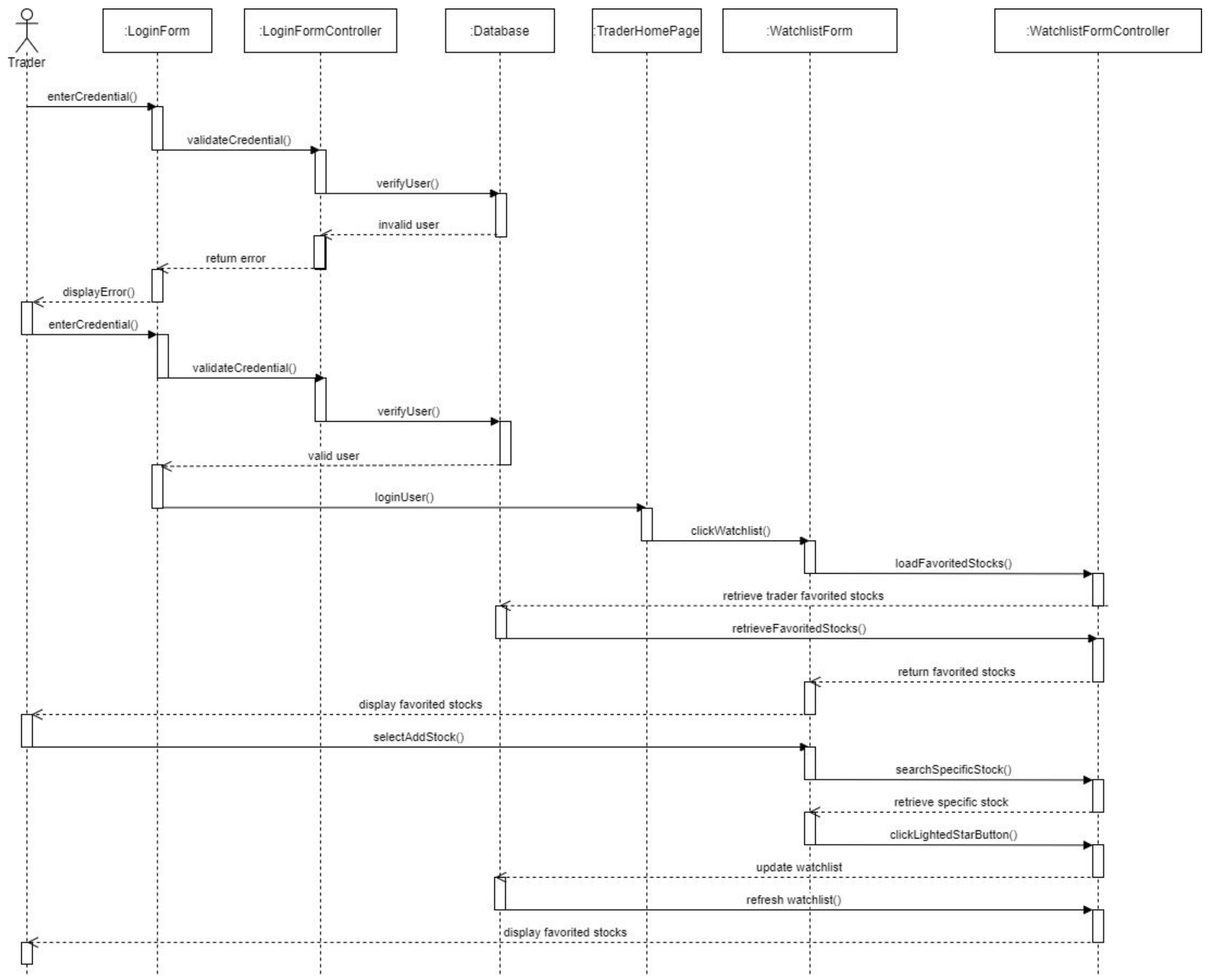
1(j) View Key Financial Information



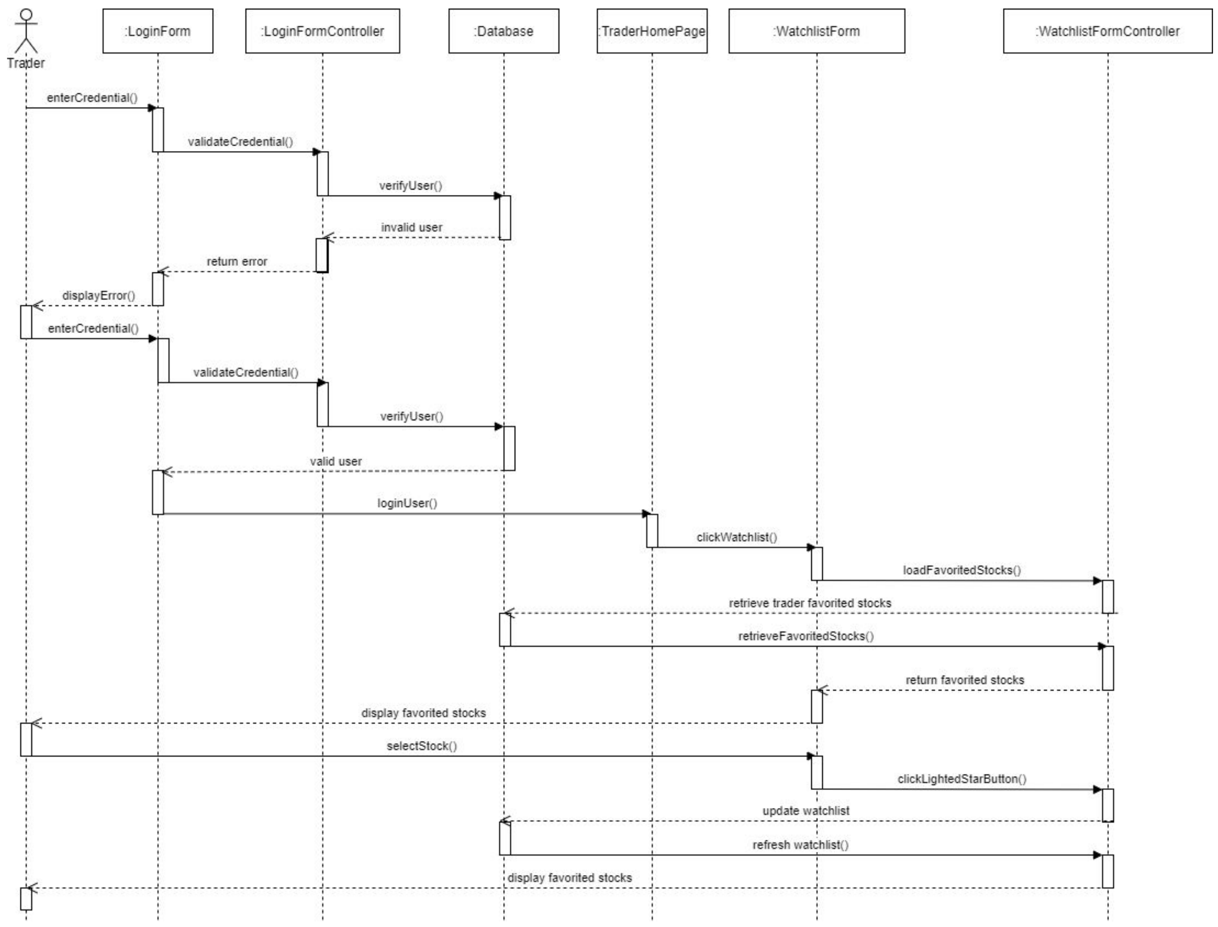
1(k) View Watchlist



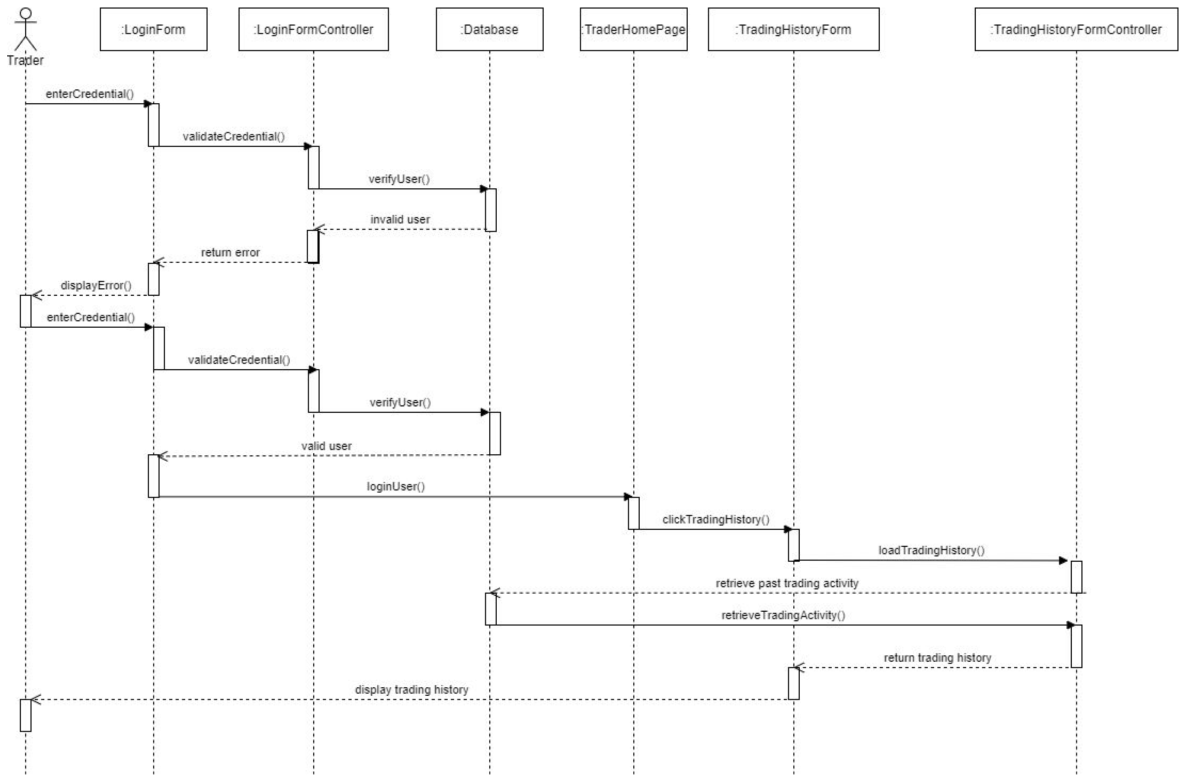
1(I) Add Stocks to Watchlist



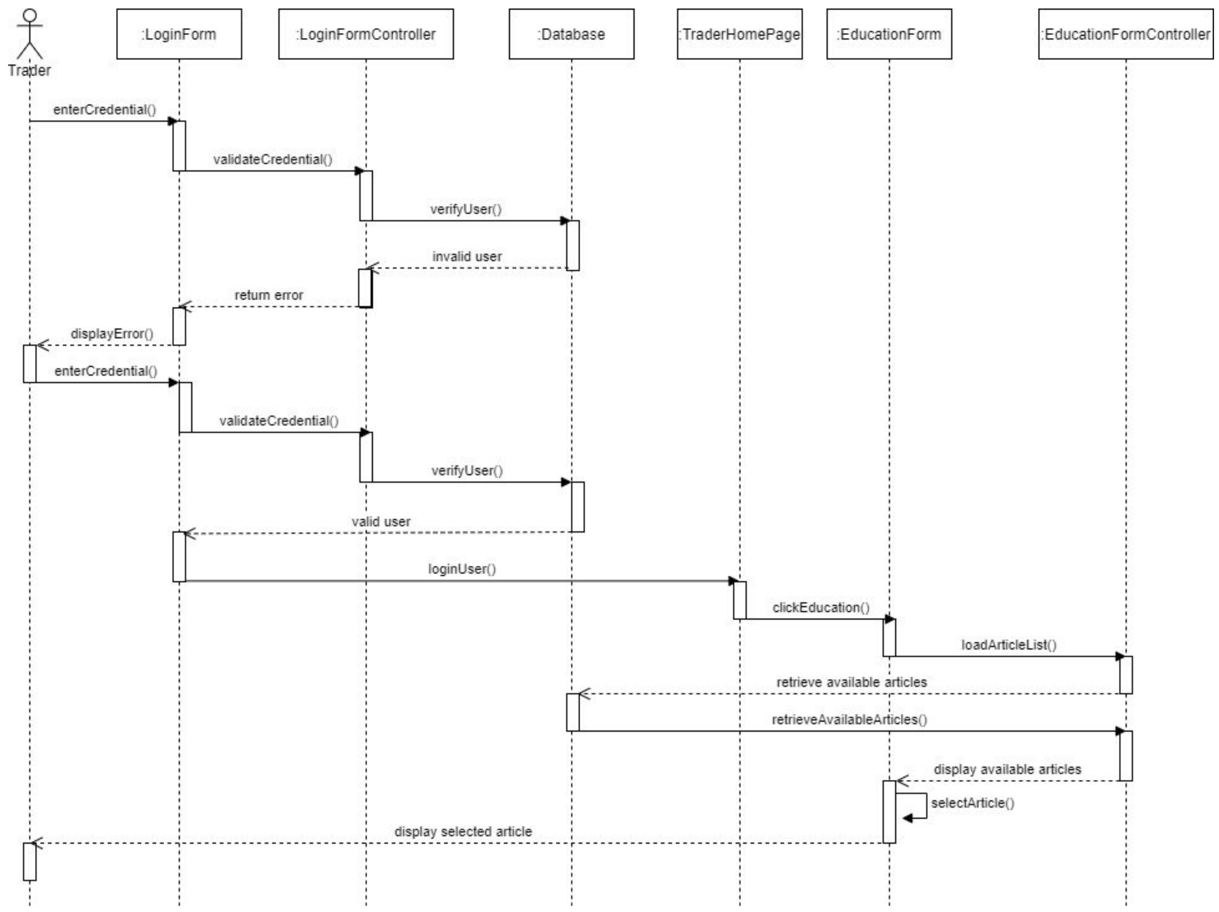
1(m) Remove Stocks from Watchlist



1(n) View Trading History

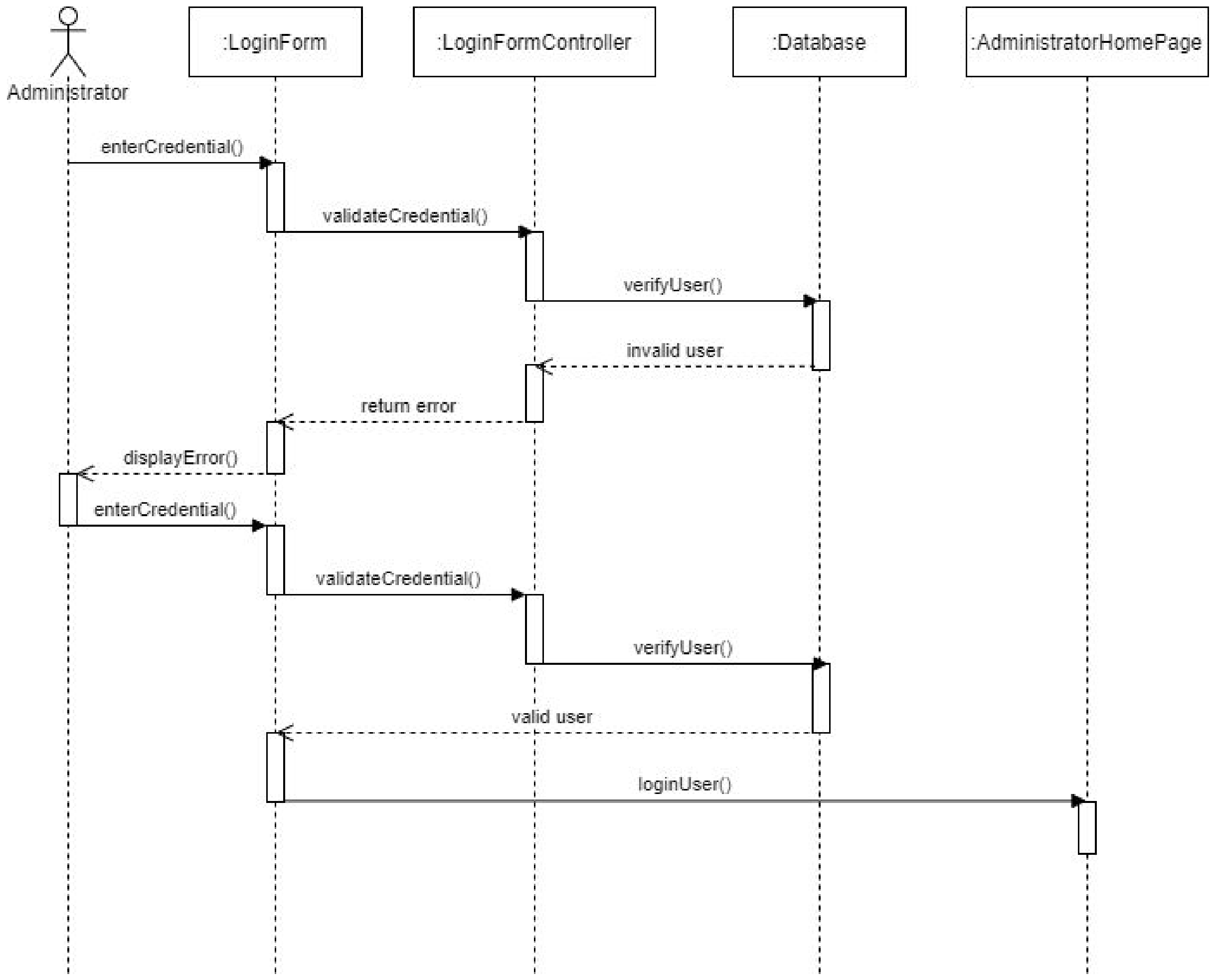


1(o) View Educational Materials

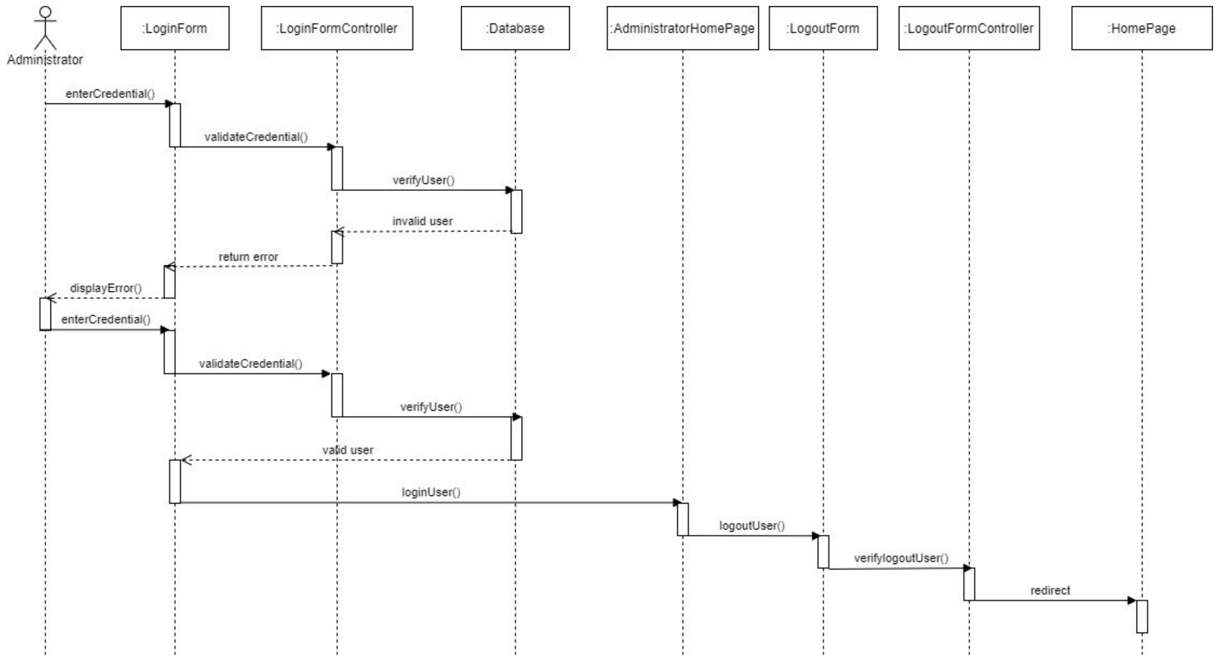


2. Administrator

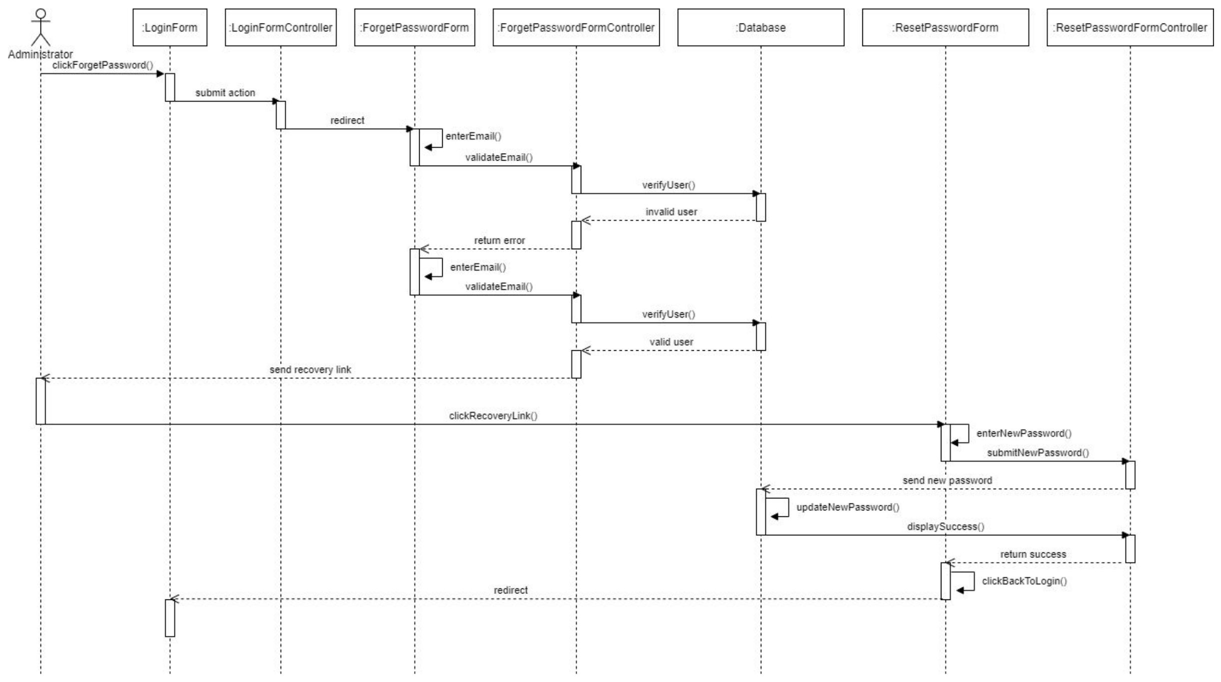
2(a) Login



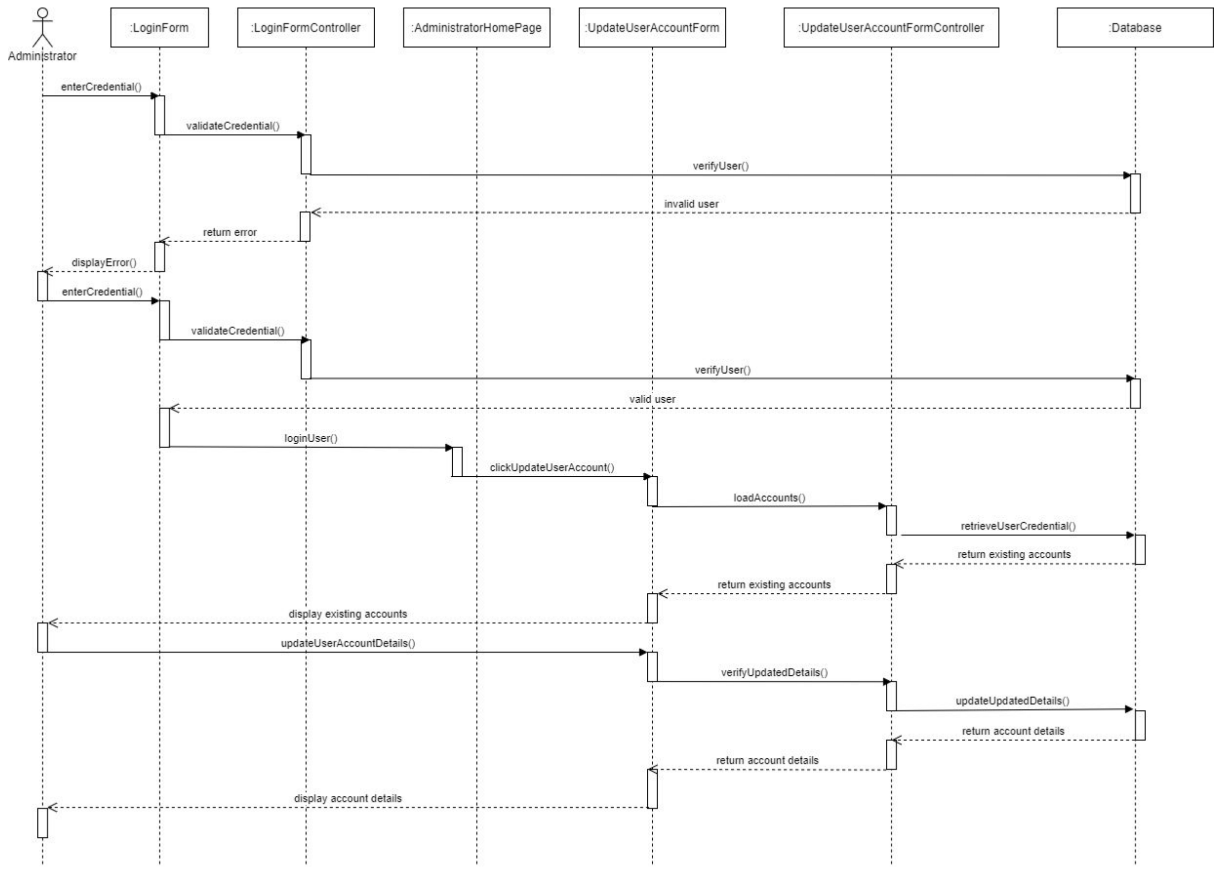
2(b) Logout



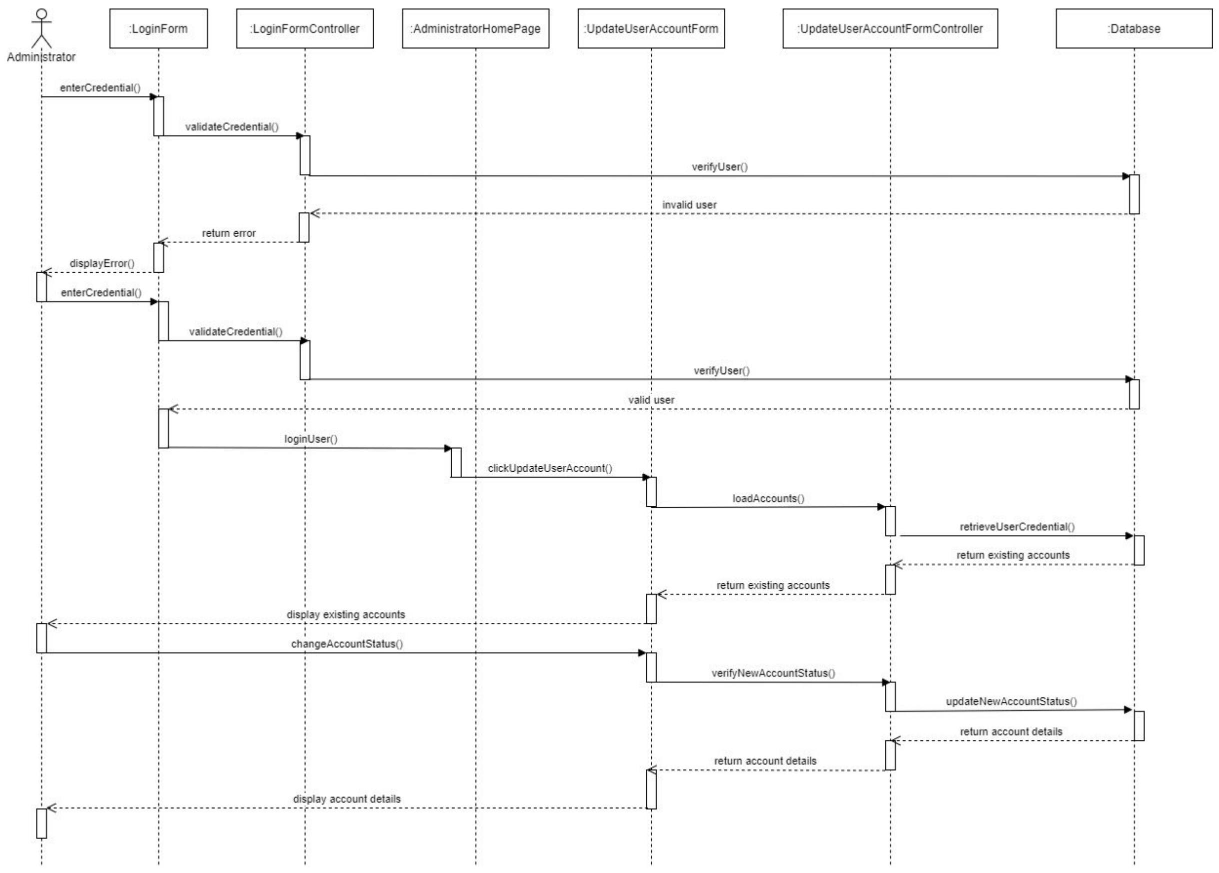
2(c) Reset Password



2(d) Update Accounts



2(e) Suspend Accounts



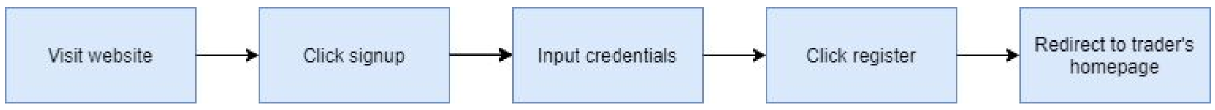
Appendix C

Process Flow Diagram

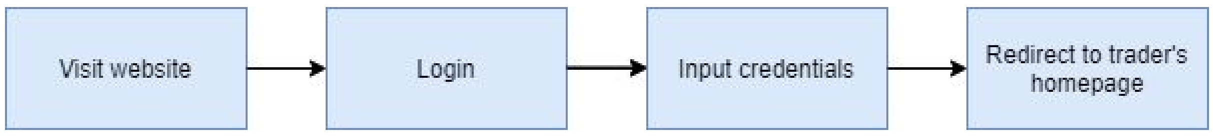
The diagrams below are a visual representation of how the flow will be for each function, with the presumption that no errors occurred. We having the following roles as the actors of our process flow diagram:

> **A.** Trader
> **B.** System Administrator

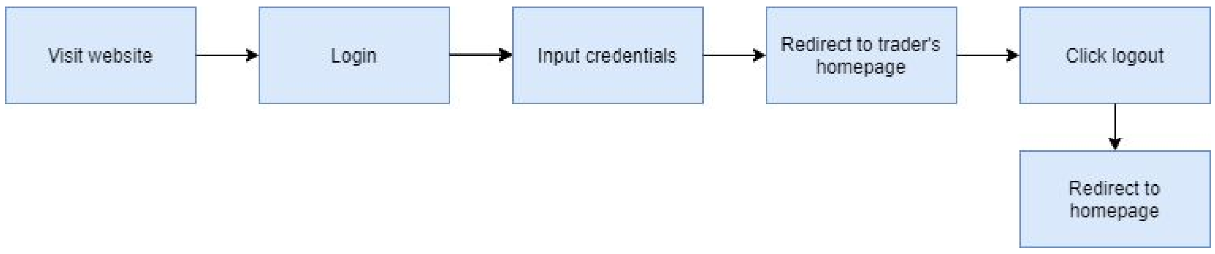
A.1 Create account



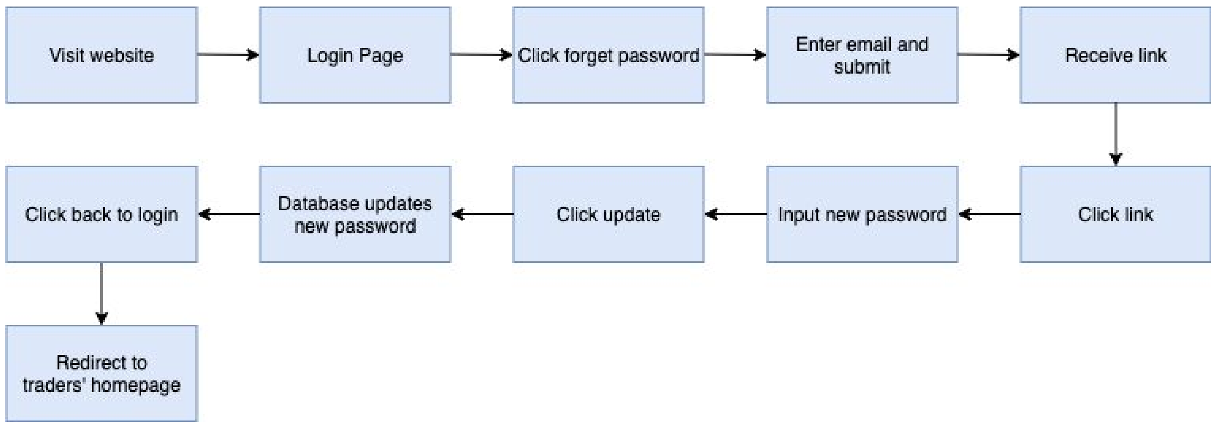
A.2 Login



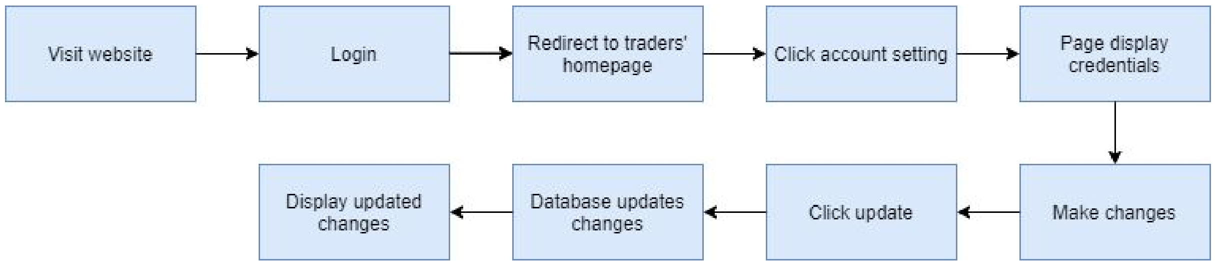
A.3 Logout



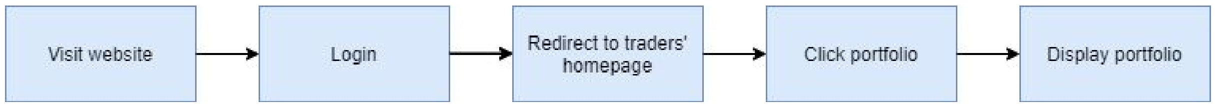
A.4 Reset Password



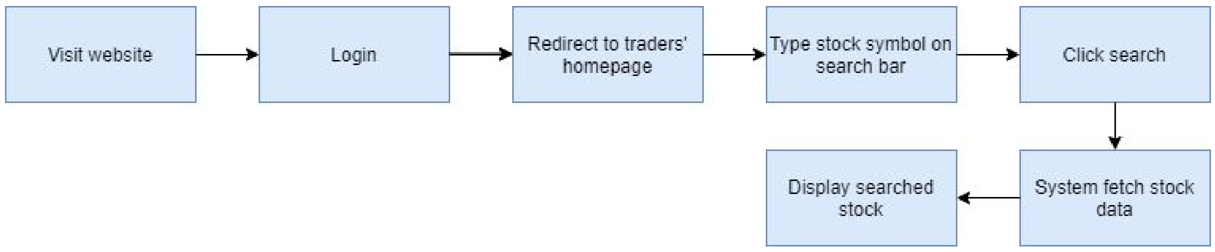
A.5 Update Account



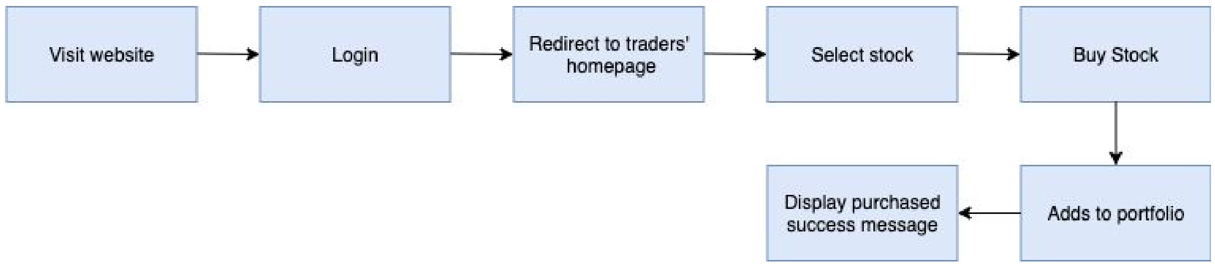
A.6 View Portfolio



A.7 Search for Stocks

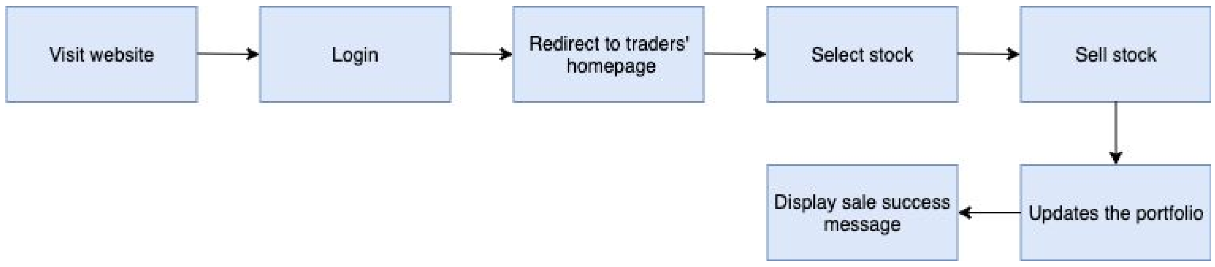


A.8 Buy Stocks

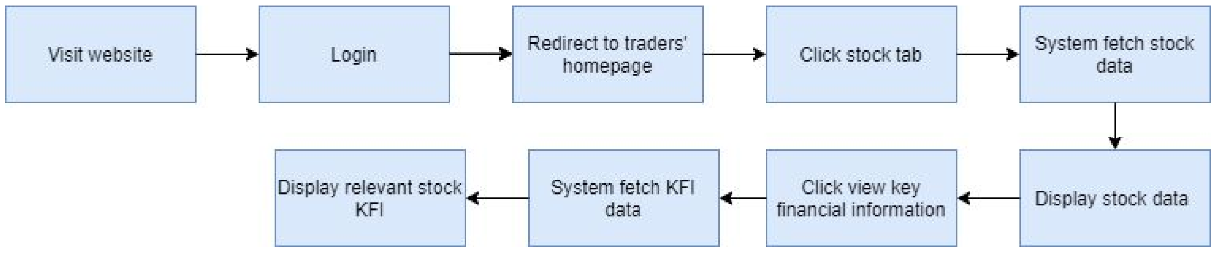


Visit website
Login
Redirect to traders' homepage
Select stock
Buy Stock
Adds to portfolio
Display purchased success message

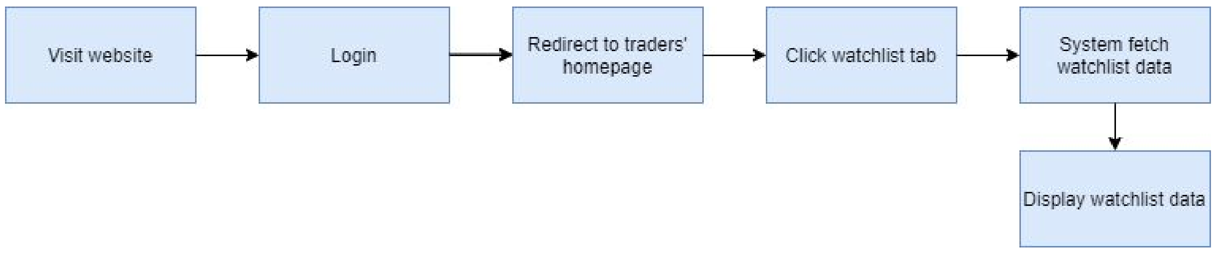
A.9 Sell Stocks



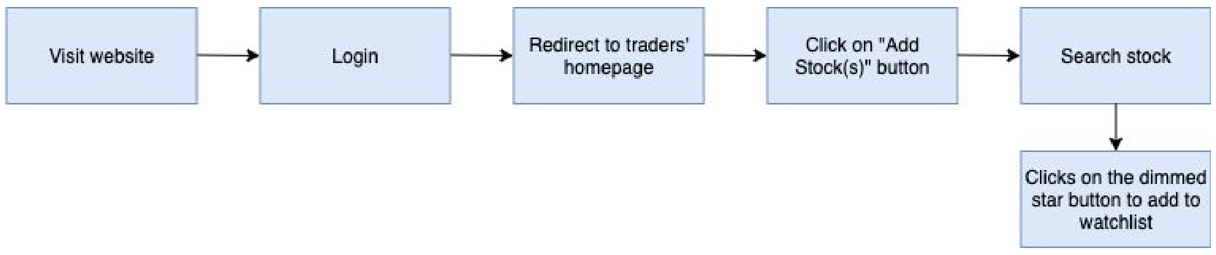
A.10 View Key Financial Information



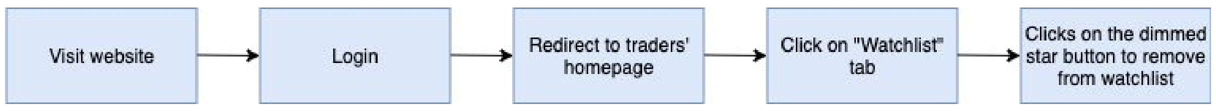
A.11 View Watchlist



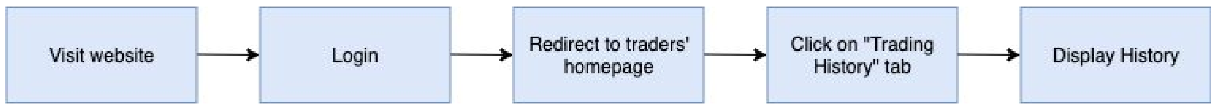
A.12 Add Stocks to Watchlist


Visit website
Login
Redirect to traders' homepage
Click on "Add Stock(s)" button
Search stock
Clicks on the dimmed star button to add to watchlist

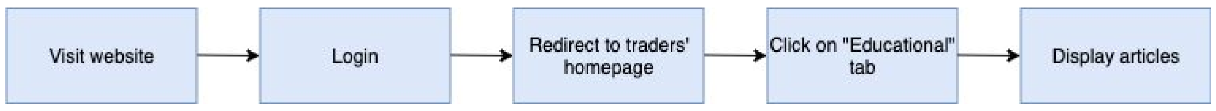
A.13 Remove Stocks from Watchlist


Visit website
Login
Redirect to traders' homepage
Click on "Watchlist" tab
Clicks on the dimmed star button to remove from watchlist

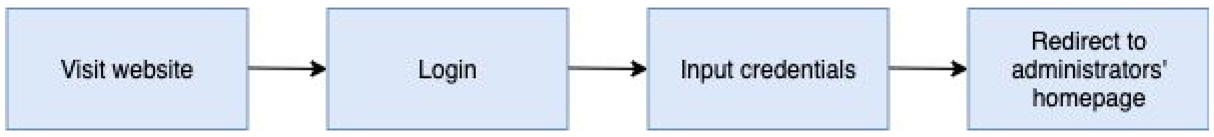
A.14 View Trading History



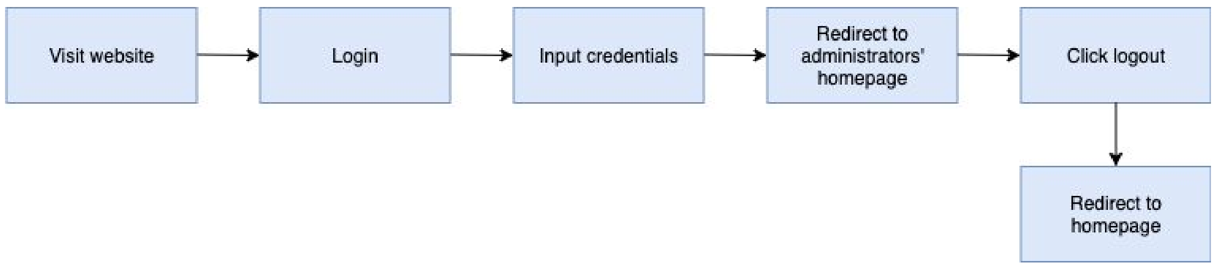
A.15 View Educational History



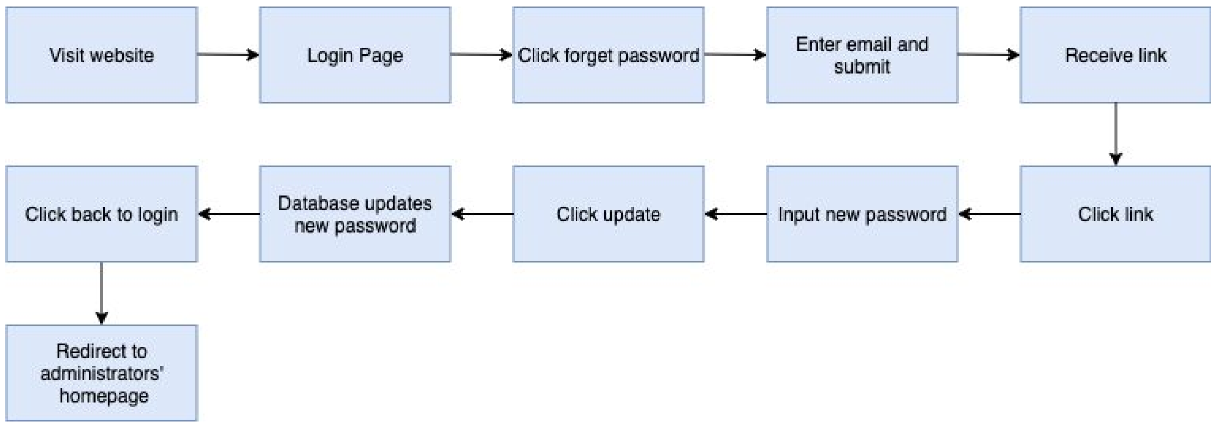
B.1 Login



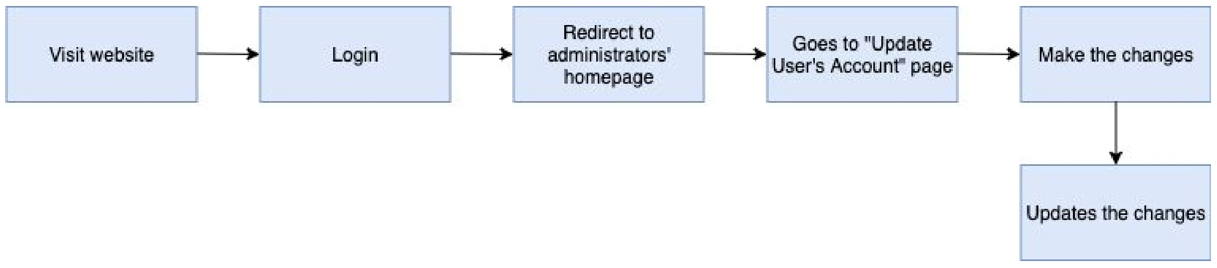
B.2 Logout



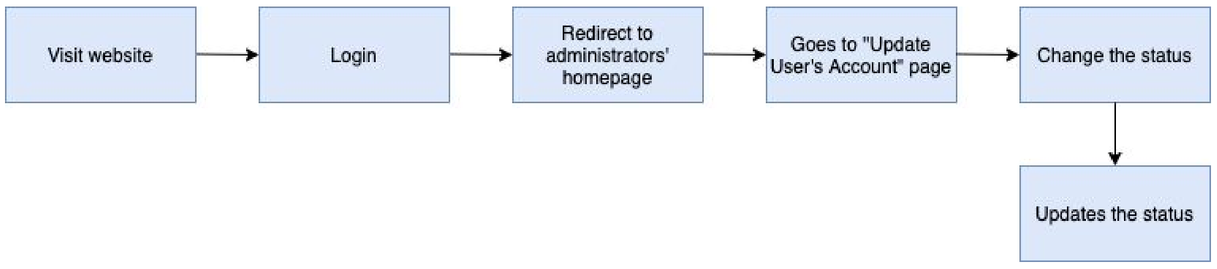
B.3 Reset Password



B.4 Update Accounts


Visit website
Login
Redirect to administrators' homepage
Goes to "Update User's Account" page
Make the changes
Updates the changes

B.5 Suspend Account


Visit website
Login
Redirect to administrators' homepage
Goes to "Update User's Account" page
Change the status
Updates the status

Glossary

- **Amazon RDS** - A distributed relational database service by Amazon Web Services
- **Cascading Style Sheets (CSS)** - A simple mechanism for adding styles to Web documents
- **HyperText Markup Language (HTML)** - The standard markup language for documents designed to be displayed in a web browser
- **JavaScript** - A client scripting language which is used for creating web pages
- **MySQL** - A relational database management system based on Structured Query Language (SQL)
- **Python** - An interpreted, objected-oriented, high-level programming language with dynamic semantics
- **Tableau** - A data visualisation tool
- **XAMPP** - Am open-source cross-platform web server solution stack package includes MySQL and PHP etc
- **SQL Injection** - A web application vulnerability that allows malicious users to interfere with SQL queries
- **Cross-site Scripting (XSS)** - A web application vulnerability on the application interactivity with the users