**Use Case 1: User\_Sign\_In**

**Overview:**

This Use Case enables existing Users to log into their own account and allows new Users to create a Virtual Stock account.

**Rationale:**

User accounts personalize Virtual Stock usage so that a User can reuse many Virtual Stock features customized to the User’s needs. These User accounts are also necessary to implement the Virtual Stock protocol function. With those aspects in mind, this Use Case ensures that a User has a Virtual Stock account before using Virtual Stock’s main functions.

**Preconditions:**

1. User has opened the Virtual Stock application successfully.

2. User has not signed in.

**Scenario:**

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1. User runs the Virtual Stock application. | 1. Sign In View will display the Sign In Form and prompt the User to enter Username and Password, or create a new account. |
| 2. User fills out the Username and Password fields and submits form. | 2. Sign In View checks form info in the User Database and finds match; displays MainWindow View. |
| 3. User fills out the Username and Password fields and submits form. | 3. Sign In View checks form info in the User Database and finds no match; clears form and prompts User to resubmit. |
| 4. User clicks the Sign up button | 4. User\_Sign\_Up invoked. |

**Scenario Notes:**

2 and 3 are mutually exclusive with 4. In other words, a User cannot sign into an existing account and create a new account during User\_Signs\_In. 3-5 occur in order, but can be broken if the User signs in correctly with Username and Password.

**Post Conditions:**

1. Home View is displayed.

2. User Profile information is loaded.

**Required GUI:**

SignIn

**Use Cases Utilized:**

User\_Creates\_Account

**Timing Constraints:**

None

**Use Case 2: User\_Sign\_Up**

**Overview:**

This Use Case enables the User to create new account.

**Rationale:**

This Use Case’s rationale directly follows from the rationale given in UC1.

**Preconditions:**

1. User has opened the Virtual Stock application successfully.

2. User has not signed in.

**Scenario:**

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1. User runs the Virtual Stock application. | 1. Sign In View will display the Sign In Form and prompt the User to enter Username and Password, or create a new account. |
| 2. User clicks Sign Up | 2. Sign In View prompts User to enter a unique Username and password. |
| 3. User enters all text fields entry. | 3. Sign In View will check basic format, and also check whether this Username exists. If it exits, message that Username already existed, clear the Username field. |
| 4. User presses Sign up Button | 4. If both Username and Password fields are filled and correct, Sign In View adds account and displays Home View with default Profile. If incorrect, message that at least one field is not filled in correctly is displayed. |
| 5. User presses Cancel button. | 5. Sign In View displays Sign In Form. |

**Scenario Notes:**

Items 1-4 must be done in order. Items 5 are mutually exclusive.

**Post Conditions:**

The User has a Virtual Stock account and Home View is displayed with default profile.

**Required GUI:**

MainWindow, SignIn

**Exceptions:**

1. Username entered already exists.

2. Username does not meet format requirements.

3. Password does not meet format requirements.

**Use Cases Utilized:**

None

**Timing Constraints:**

None.

**Use Case 3: Show\_Main\_Window**

**Overview:**

This Use Case enables Users to see the main window

**Rationale:**

**Preconditions:**

1. User has signed in.

2. MainWindow is displayed.

3. User has connected to the Internet.

**Scenario:**

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1. User clicks stock name | 1. Virtual stock will show the details of the stocks including price, shares, open, volume, today’s high, average volume today’s low, market capacity, 52 week high, 52 week low, P/E ratio and Div/Yield. |
| 2. User clicks buy | 2. A window will show the price and there will and input for user to type the number of shares to be purchased. User’s remaining cash can also be shown out. |
| 3. User clicks sell | 3. A window will show and asks the user how many shared a user wants to sell. |
| 4. User clicks Account - Portfolio | 4. Portfolio window shows up |
| 5. User clicks Account - Account | 5. Account window shows up |
| 6. User clicks Account - Banking | 6. Banking window shows up |
| 7. User clicks Account - History | 7. History window shows up |
| 8. User clicks Account - Settings | 8. Settings window shows up |
| 9. User clicks Help - About | 9. About window shows up |
| 10. User clicks File - Exit | 10. Program exits |

**Scenario Notes:**

None

**Post Conditions:**

1. If User has made updates and saved, the User’s stock is saved successfully, and MainWindow displays the current shares of users’ stocks

2. If User cancels, MainWindow displays the original stocks the user has.

**Required GUI:**

MainWindow, StockPurchase, StockSell, Portfolio, Account, Banking, History, Settings, About

**Exceptions: Use Cases Utilized: Timing Constraints:**

None None None

**Use Case 4: Show\_Portfolio**

**Overview:**

This Use Case is to enable Users to see the portfolio window.

**Rationale:**

**Preconditions:**

1. User is logged into Virtual Stock.

2. MainWindow is displayed.

3. User clicks Account-Portfolio.

**Scenario:**

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1. User clicks 1D. | 1. Chart gives 1 day data of user’ stocks |
| 2. User clicks 1W. | 2. Chart gives 1 week data of user’ stocks |
| 3. User clicks 1M. | 3. Chart gives 1 month data of user’ stocks |
| 4. User clicks 3M. | 4. Chart gives 3 months data of user’ stocks |
| 5. User clicks 1Y. | 5. Chart gives 1 year data of user’ stocks |
| 6. User clicks ALL. | 6. Chart gives all data of user’ stocks |

**Scenario Notes:**

Items 1-6 are mutually exclusive.

**Post Conditions:**

The user will be returned to the MainWindow.

**Required GUI:**

Portfolio

**Exceptions:**

None

**Use Cases Utilized:**

Show\_Main\_Window

**Timing Constraints:**

None.

**Use Case 5: Show\_Account**

**Overview:**

This Use Case it to enable Users to show user’s account information including the total money, stock and cash percentage. User can also withdraw funds from this window.

**Rationale:**

**Preconditions:**

1. User is logged into Virtual Stock.

2. MainWindow is displayed.

3. User clicks Account-Account.

**Scenario:**

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1. User clicks OK. | 1. Window closes |
| 2. User clicks Withdraw Funds. | 2. A window will show up and ask the user how much funds the user wants to withdraw. |

**Scenario Notes:**

It the user types more money than he/she has, the system will not withdraw the funds from virtual stock to his/her bank and asks the user to reinput all the numbers.

**Post Conditions:**

The user will be returned to the MainWindow.

**Required GUI:**

Account

**Exceptions:**

None

**Use Cases Utilized:**

Show\_Main\_Window

**Timing Constraints:**

None.

**Use Case 6: Show\_Banking**

**Overview:**

This Use Case enables Users to transfer money to bank, transfer money from bank to virtual stock, automatic deposits and link account. The user can also find the linked bank account(s).

**Rationale:**

**Preconditions:**

1. User is logged into Virtual Stock.

2. MainWindow is displayed.

3. User clicks Account-Banking.

**Scenario:**

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1. User clicks “Transfer to virtual stock” | 1. The software will pop up a window and ask the user to select a bank account and money to be transferred. |
| 2. User clicks “Transfer to bank” | 2. The software will pop up a window and ask the user to select a bank account and money to be transferred. |
| 3. User clicks “Automatic deposit” | 3. The software will pop up a window and ask the user to select the time and money to be transferred from bank account to virtual stock |
| 4. User clicks “Link account” | 4. The software will pop up a window and ask the user to select a bank account and input the routing number and account number. |

**Scenario Notes:**

1. There must be at least one bank account in user’s banking page.

2. There must be enough funds for user to transfer.

**Post Conditions:**

The user will be returned to the MainWindow.

**Required GUI:**

Banking

**Exceptions:**

None

**Use Cases Utilized:**

Show\_Main\_Window

**Timing Constraints:**

None.

**Use Case 8: Show\_History**

**Overview:**

This Use Case enables Users to see the history records. It will also allow user to select the date from a specific time.

**Rationale:**

**Preconditions:**

1. User is logged into Virtual Stock.

2. MainWindow is displayed.

3. User clicks Account-History.

**Scenario:**

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1. User selects a date range from the comboBox | 1. The system will show the history record during this time. This includes stock name, sell or buy, price and date. |

**Scenario Notes:**

None

**Post Conditions:**

The user will be returned to the MainWindow.

**Required GUI:**

History

**Exceptions:**

None

**Use Cases Utilized:**

Show\_Main\_Window

**Timing Constraints:**

None.

**Use Case 9 Show\_Settings**

**Overview:**

This Use Case enables Users see and edit settings

**Rationale:**

**Preconditions:**

1. User is logged into Virtual Stock.

2. MainWindow is displayed.

3. User clicks Account-Portfolio.

**Scenario:**

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1. User edits account | 1. Account name changes |
| 2. User edits email | 2. Email address changes |
| 3. User edits password | 3. User password changes |
| 4. User edits phone number | 4. user phone number changes |
| 5. User edits address | 5. User address changes |
| 6. User edits Update button | 6. All changes saved |
| 7. User edits Cancel button | 7. All changes discarded |

**Scenario Notes:**

None

**Post Conditions:**

The user will be returned to the MainWindow.

**Required GUI:**

Settings

**Exceptions:**

None

**Use Cases Utilized:**

Show\_Main\_Window

**Timing Constraints:**

None.