MoneyBMineWP Web Application:

1. **Three different users:**

- Admin:

  In the (NYSEDailyPriceAsController), I set [Authorize(Users ="katrina0411@live.cn")] to be the Admin role who can access the database. We can change it to the real admin when we deploy this to the company server.
  User name : katrina0411@live.cn
  password: Katrina@0

  All the action against the SQL database table (Create, Read, Update and Delete) will be under the control of this Admin user.

- Registered user:

  They can download the zip file to install the MoneyBMineWPF app to their computer to manipulate the data. The zip file also includes a User Guide shows the user how to use the application with snapshots.

- Anonymous user:

  They are able to see the data on the Home page.

2. **Different pages**

   All pages follow the same layout which contains NAVBAR can jump to other pages (except the Whole Data Page only can be redirected from the Pages relate to CRUD the database).

- **Home Page:**

  - news from global market and link to the https://www.cnbc.com/world-markets/
  - short introduction about the WPF app and link to the Download Page
   - iframe which contains a ViewData.aspx file under the folder Pages that shows the data

- **Company Page:**
  more information about the company

- **Admin Pages:**

- Create Page: create new data to database
- Details Page: read the particular record from database
- Edit Page: update the particular record to database
- Delete Page: delete the particular record from database
- Whole Data Page: create, read, update and delete data from database

- **Register Page:**

  register email and set the password

3. **Custom log:**

   I am using Log4Net to log the message in the flat file (name RollingFileLog.txt). In the Web.Config, I set the appender as RollingFileAppender which can prevent the file which storing the message from growing unlimited.
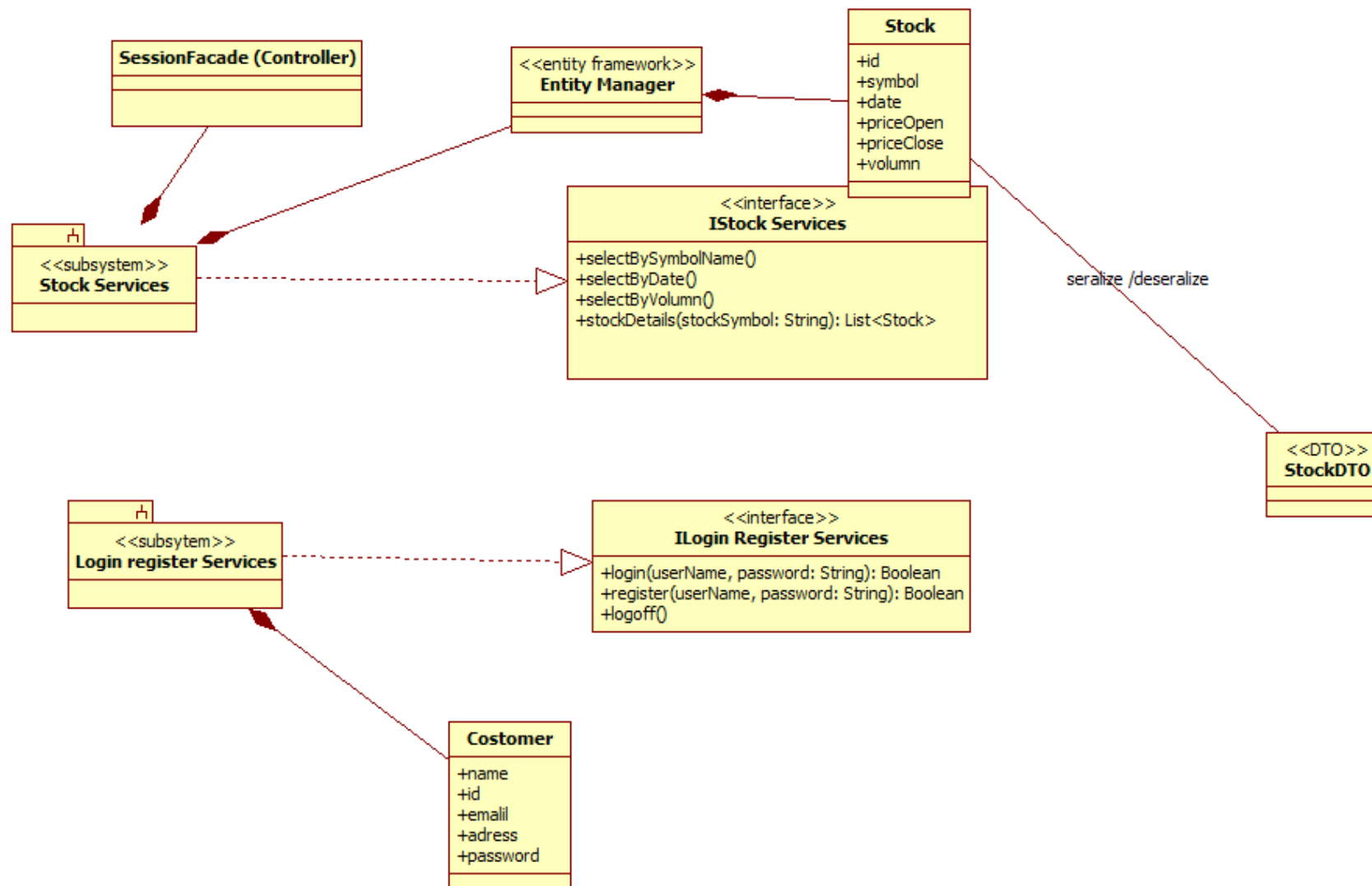   I also created a class called LogHelper which can be used in any code block if we need to add the log message function.
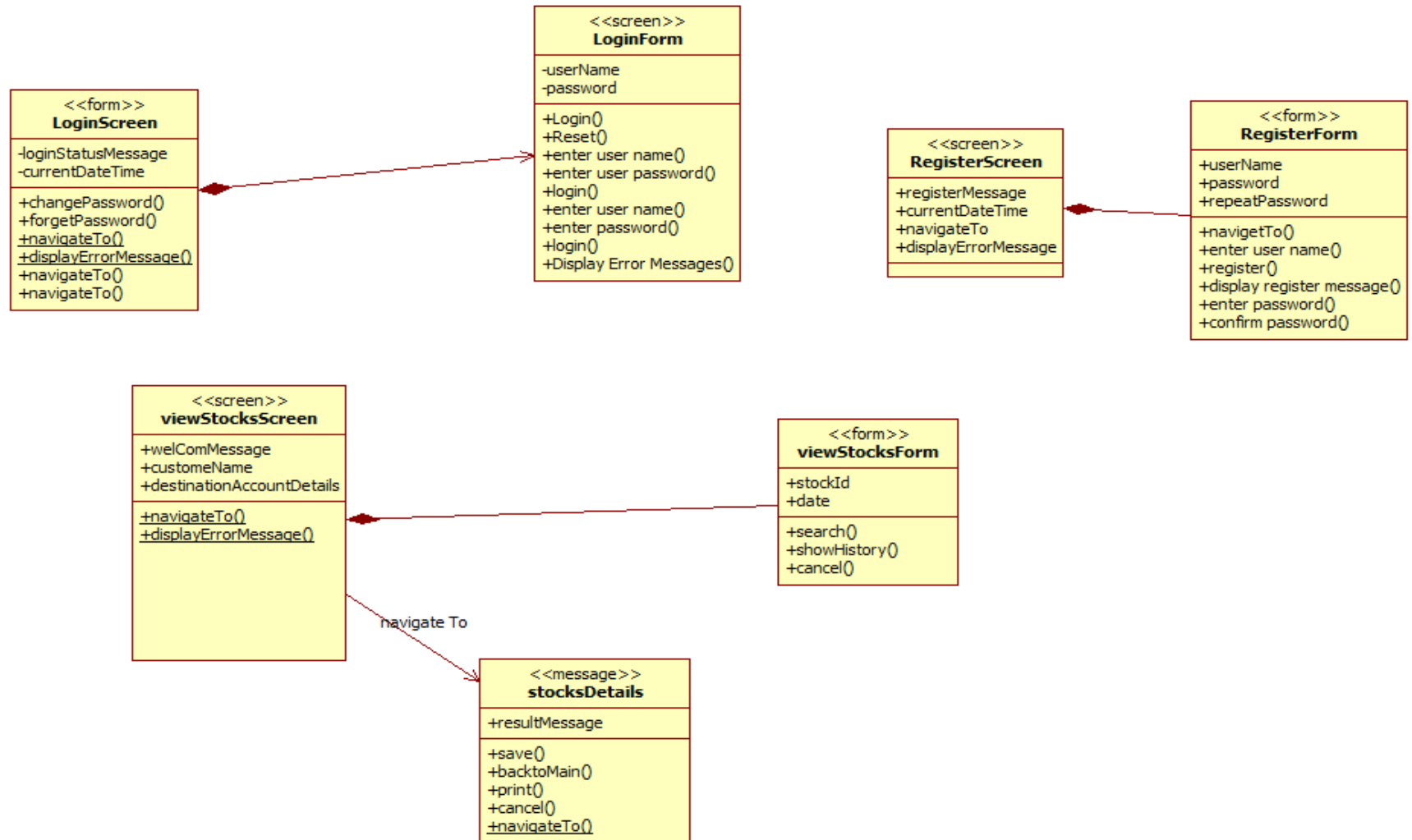   And now the Web app will log a message like the example to the file every time we visit the Home Page.
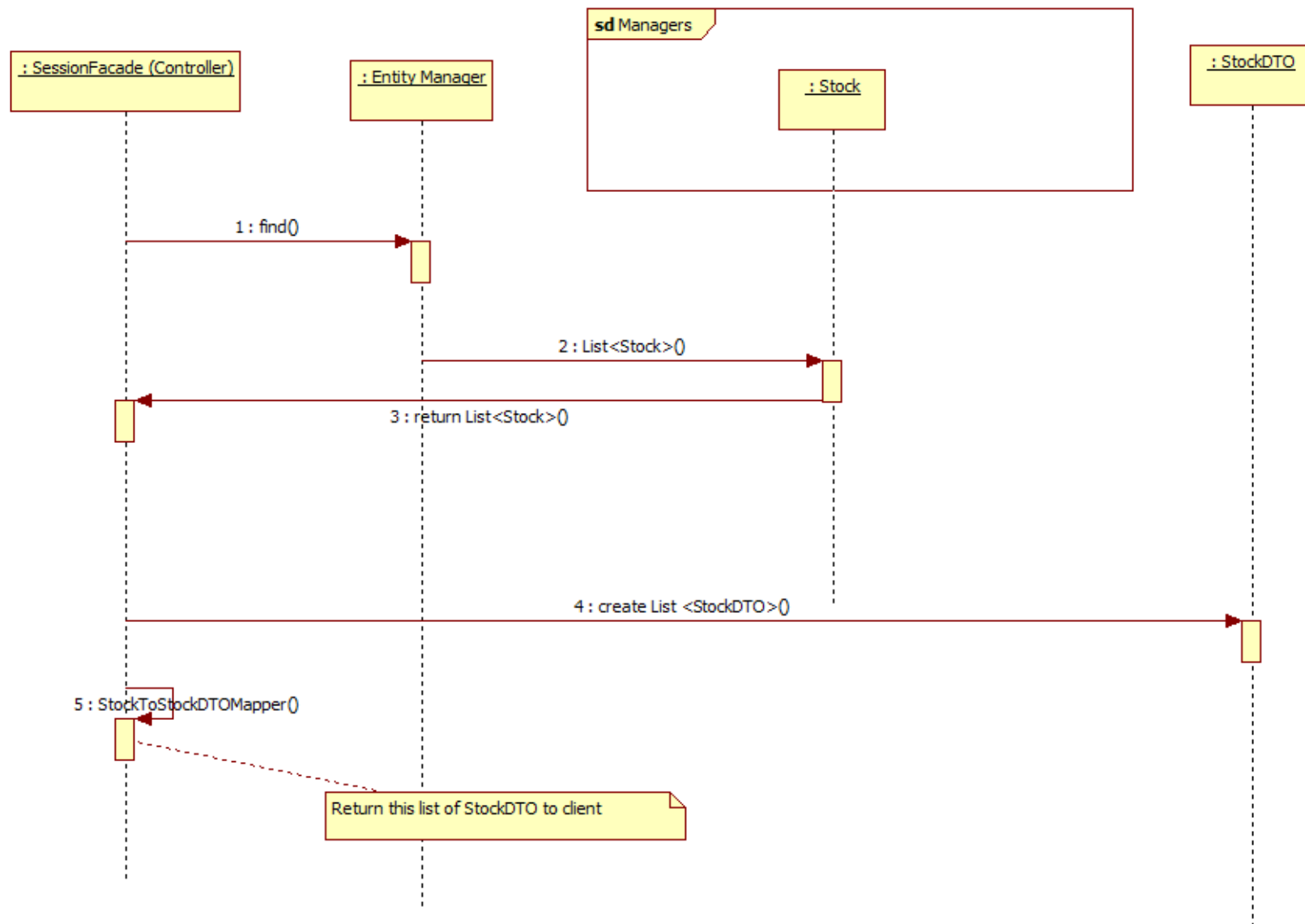   **Example:**
   21:50:13,004 [C:\Users\Upskilled\MoneyBMine-JULY\MoneyBMineWP\Controllers\HomeController.cs] INFO - This is the Home Page log message
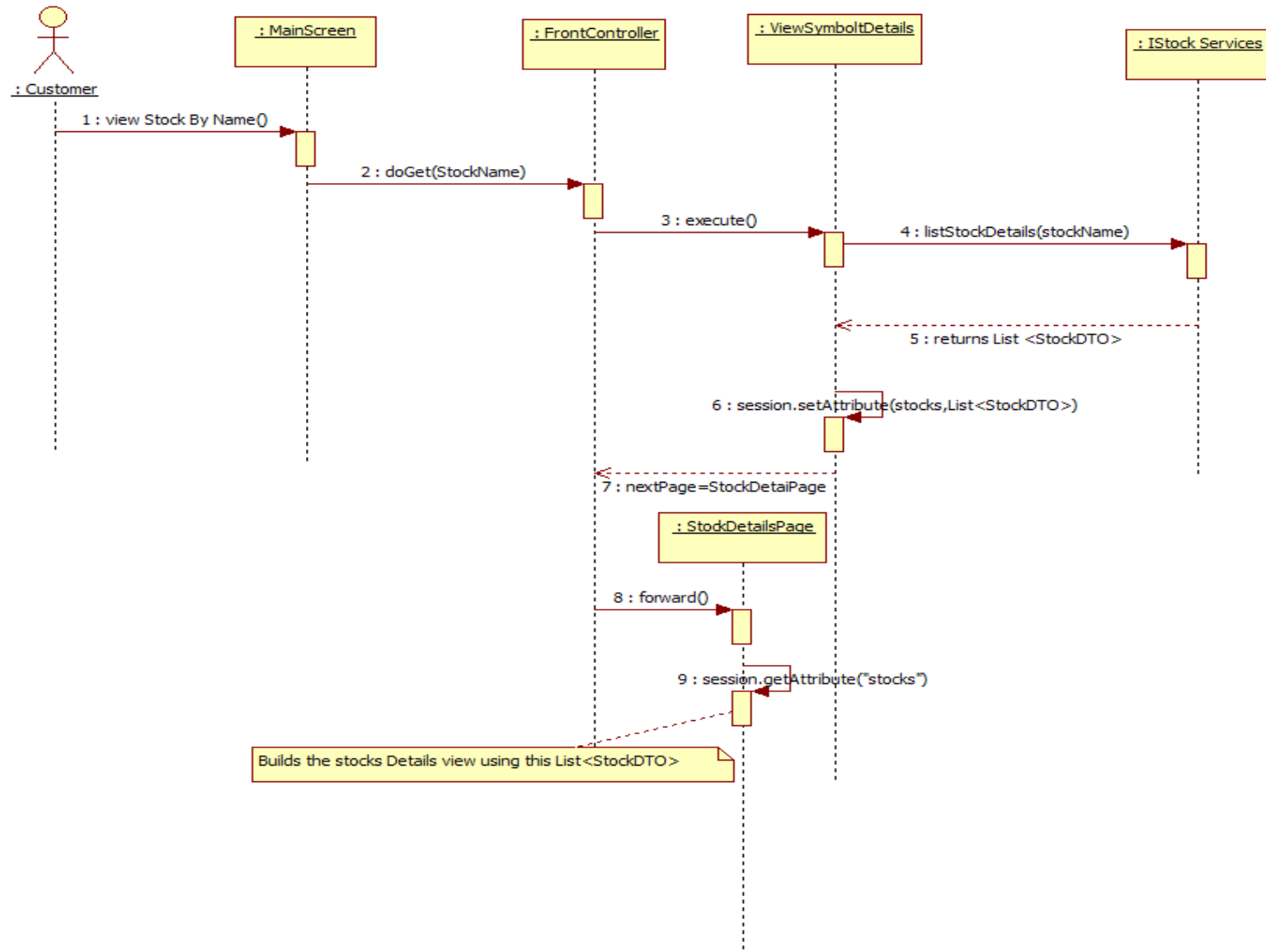
4. **Object model diagram shows different classes in the application.**

<<screen>>
LoginForm

-userName
-password

+Login()
+Reset()
+enter user name()
+enter user password()
+login()
+enter user name()
+enter password()
+login()
+Display Error Messages()

<<form>>
LoginScreen

-loginStatusMessage
-currentDateTime

+changePassword()
+forgetPassword()
+navigateTo()
+displayErrorMessage()
+navigateTo()
+navigateTo()

<<screen>>
RegisterScreen

+registerMessage
+currentDateTime

+navigateTo
+displayErrorMessage

<<form>>
RegisterForm

+userName
+password
+repeatPassword

+navigetTo()
+enter user name()
+register()
+display register message()
+enter password()
+confirm password()

<<screen>>
viewStocksScreen

+welComMessage
+customeName
+destinationAccountDetails

+navigateTo()
+displayErrorMessage()

<<form>>
viewStocksForm

+stockId
+date

+search()
+showHistory()
+cancel()

navigate To

<<message>>
stocksDetails

+resultMessage

+save()
+backtoMain()
+print()
+cancel()
+navigateTo()

5. **UML sequence diagram shows the working routes in our web application**

6. **Source code control:**

Using Git and VisualStudioTeamService for version control and collaboration which allows me to push the changes to KatrinaTong's Github and katrinatong.visualstudio.com. It shows the changes to the parent commit which can explore the changes easily I have made.

7. **Unit test**

We have a project called MoneyBMineWP.Tests to do the unit test for our web application.

MoneyBMineWPF:

1. Register user can download this application which allows them to search and sort the data they need.
2. This application also check and download the update every time user opens it when they can reach the internet.