

CS611-<Final Project>

<Portfolio Management System>

<Douglas Moy, Chengzhi Dong, Zhiqi Jin>

<dougmoymoy@bu.edu, tsuna@bu.edu , jinzhiqi@bu.edu>

<U71000048, U96356434, U10876477>

Files

<A **brief** description of each file and what it does>

DatabaseConfig.java - Java class that configures our database with the given information in our db.properties file. Has get and set functions for DB related parameters.

DatabaseConnectionPool.java - Java class that manages a pool of database connections. File also establishes a connection to the database.

Response.java - Java class that establishes the response from the database, parses if it is a valid connection.

MessageController.java - Controller that houses methods to interact with messages in the database such as adding messages to the database or retrieving them.

StockController.java - Controller that houses methods to interact with stocks in the database such as adding stocks to the database or retrieving them.

UserController.java - Controller that houses methods to interact with users in the database such as adding users to the database or retrieving them

MessageDaoImpl.java - Java class that houses functions that directly interact with the message table in our database.

StockDaoImpl.java - Java class that houses functions that directly interact with the stocks table in our database.

TransactionDaoImpl.java - Java class that houses functions that directly interact with the transactions table in our database.

UserDaoImpl.java - Java class that houses functions that directly interact with the users table in our database.

iMessageDAO.java - Interface for messageDaoImpl.java that defines the functions MessageDaoImpl.java implements.

iStockDao.java - Interface for StockDaoImpl.java that houses the functions StockDaoImpl.java implements.

iTransactionDao.java - Interface for TransactionDaoImpl.java that houses the functions TransactionDaoImpl.java implements.

iUserDao.java - Interface for UserDaoImpl.java that houses the functions UserDaoImpl.java implements.

UserStockInfo.java - Java object that represents a user's stocks and all relevant information associated with owning stocks like amount and unrealized profit. Has get and set functions for the attributes.

Message.java - Java object that represents a message in our system. Houses relevant information like timestamp and the actual message as well as read or unread. Has get and set functions for the attributes.

Stock.java - Java object that represents a stock in our system. Houses relevant information associated with a stock like price, name or symbol. Has get and set functions for the attributes.

Transaction.java - Java object that represents a transaction in our system. Houses relevant information associated with each transaction like whether or not it is a buy or sell transaction or stocks and users involved. Has get and set functions for the attributes.

User.java - Java object that represents a user in our system. Houses relevant information associated with users in our system like name, email, ID, balance. Has get and set functions for the attributes.

CurrentUser.java - Java class that maintains who the current user is.

ManagerFirstPage.java - Houses logic for the first page the manager opens. Portal to all the other pages. This file generates the swingGUI for the page.

UserNotifyPage.java - This file generates the swingGUI for the page where the manager can view all users and interact with them by sending messages or blocking users.

UserRequestsPage.java - This file generates the swingGUI for the page where the manager views all the users who have requested to become registered traders. Manager can accept or deny requests.

addEditDeleteStocks.java - This file generates the swingGUI for the page where the manager views all stocks and can add, edit or delete stocks.

ManageAccountPage.java - This file generates the swingGUI for the page where the user can manage their account. Is the place where the user can manage their account.

StockDisplayPage.java - This file generates the swingGUI for the page where the user can view all stocks, they can then buy on this page.

UserMenuPage.java - This file generates the swingGUI for the first page the user opens when logging in. This page is a portal to all the other pages.

UserStockPage.java - This file generates the swingGUI for the page where the user can view all the stocks they own. On this page they can also sell stocks.

HomePage.java - This file generates the swingGUI for allowing the user to perform different actions, such as logging in or signing up, and uses a UserController object that interacts with data access objects.

WelcomePanel.java - This file generates the swingGUI that displays a welcome message with two buttons (for the manager and the customer).

CustomerLogin.java - This file generates the swingGUI that displays a login page for customers.

ManagerLogin.java - This file generates the swingGUI that displays a login page for manager.

SignUp.java - This file generates the swingGUI that creates new accounts.

Notes

1. We used Maven dependencies and used an AWS database. This code may not work on your machine if your environment is not set up properly.

2. <Bonus Done>

-Manager can send and write a message to any user.

-User has an inbox where they can receive a message from the manager

-Manager can block users after they have already signed up.

-Check for invalid inputs, repeat sign ups.

3. <Notes to grader>

-Language of buttons may depend on the computer the user is using because java automatically creates some of the buttons.

How to compile and run

1. Navigate to the directory containing all the java files
2. Run the following instructions:

<Example below>

```
javac -d bin src/*.java
```

```
java -cp bin HomePage
```

Input/Output Example

-Input/Output is all done on the Java SwingGUI.