Final Project User Stories

By Andrew Gaw

Case Study: Brokerage Trading Platform

Basic stock trading program where user can register, login, and establish cash position then start trading. The program will keep track of positions, showing history of trades for each user and display current outstanding positions. The program also keeps track of total profit and loss and net cash available to trade. Cash balances can be updated and also impacted by the actual trades. There are a number of integrity checks to make sure user can only purchase the amount of shares that is within their cash account limits. User also cannot short sell nor sell more shares than they have of a particular position. The stock price currently is not live and are static prices updated via database from one of the major exchanges. A random number generator on stock prices is currently used for testing purposes as temporary pseudo plug for live feed onto the system. The stock symbols are all pre-validated as they are populated from a major stock exchange database.

Prerequisites

* Apache Tomcat server 9
* Eclipse IDE
* Java 8 SDK \* JRE
* Oracle 11g Database –must set on autoincrement of keys on the DB itself.
* JUnit 4

JUnit 4 tests (Unit testing for major DAO methods)

* Test were conducted on major DAO classes, further testing and development needs to be done on other DAO classes.

Known Bugs

* Registration page does not provide all the prompts to show why a user may have made a mistake in registration – although validation is fine.
* Not all figures show proper decimal places or dollar signs.
* Logout credentials appear to not work properly at the moment, paging back after logout on some pages appear possible.
* Certain pages may retain information when it should be cleared.
* Couldn’t get the check user null to prevent unauthenticated users from going to pages that require log in. System would throw exception errors so instead…..
* The navigation bar buttons are disabled depending on the state the user is in. Maybe this not the best way to design a program but the tabs displaying financial information requires proper log in.

Uncompleted

* Live stock price feeds into the system.
* Live graphs of major indices.
* Stock news section.
* User update personal info.
* Admin user – to log in and make changes.
* Allow more user functionality, view historical trades by date, symbol, etc..

Problems faced:

* Navigating and learning Spring MVC and do this project on it.
* Passing information from controllers to controllers to view
* Customizing non-model based information to be displayed in the view – eventually implemented another model to take advantage of form-backing-object capability from MVC.
* Persisting information among controllers.
* Getting around users having access to sensitive pages without logging in.
* Basic Login credentials did not anticipate the need to initialize additional user information to access other parts of the program if user choosers to skip to certain tabs after logging in.

Built with

* Java 8 – Programming Language
* Spring MVC – Spring framework 5.0
* JSP – Java Server Pages
* JDBC – interact with Oracle DB
* HTML/CSS
* JUNIT 4 - Testing
* Maven – Dependency Management

User Stories

* As a user, I can register an account to get access to the platform.
* As a user, I can confirm my registration detail before registering.
* As a user, I can create a cash position to start trading.
* As a user, I can log in with username and password.
* As a user, I can log out.
* As a user, I can access any stock symbol to potentially trade the ticker.
* As a user, I can check the price of the stock.
* As a user, I can place a buy or sell on the stock.
* As a user, I can input the number of shares to buy.
* As a user, I can confirm my trade before actual order is executed.
* As a user, I can check my trading history.
* As a user, I can check my current portfolio.
* As a user, I can check my cash balance.
* As a user, I can check my profit and loss figure.
* As a user, I can deduct cash from my account.
* As a user, I can add cash from my account.