Written by Samual Allan

Assessment 1 Report on the Trading Application

Overview of the Trading Application

The trading application models a simple market and exchange system, designed around the buying and selling of stocks using RabbitMQ as the middleware. This project focuses on enabling real-time ordering and trade execution.

Middleware Utilization with RabbitMQ

RabbitMQ is employed as the core middleware for message brokering in this trading application. Its role is to ensure efficient and reliable communication between different components of the system. Specifically, RabbitMQ handles the following tasks:

- **Message Routing and Exchange Management**: RabbitMQ is in charge of the routing of messages between the trader and the exchange by setting up exchanges for orders and trades. This ensures that all relevant messages are broadcasted to the appropriate queues.
- **Asynchronous Communication**: RabbitMQ enables asynchronous communication by allowing the application to publish and consume messages independently. This is imperative for handling multiple orders and trades in real-time without delays.

Key RabbitMQ functionalities in the application include:

- **Connection Management**: Establishing and maintaining a connection to the RabbitMQ server.
- **Exchange and Queue**: Setting up exchanges and binding them to the appropriate queues for orders and trades.
- **Message Publishing and Consumption**: Methods to publish orders and trades and to consume messages from the respective queues.

Implementation

The trading application is developed using C# and Windows Forms, providing a user-friendly interface for interaction. The primary components of the application include:

- **User Interface**: The GUI is designed to allow users to enter their username, select a stock, and submit buy or sell orders. The interface also displays current buy and sell orders and lists the prices of various stocks.
- **Order Management**: Users can submit buy and sell orders, which are broadcast to the RabbitMQ exchanges. The orders are displayed in respective lists.
- **Trade Execution**: The application processes trades by matching buy and sell orders based on the specified prices.

Team Collaboration

Written by Samual Allan

After our first meeting in week 2, Dalton created the GitHub repository for the project, which served as the central platform for version control and collaboration. The repository allowed team members to work on different parts of the project simultaneously, allowing for efficient code management and integration. This also allowed for members to peer review code to maintain quality and consistency.

User Interface

The trading app features a GUI to provide users with more pleasurable trading experience. The main elements of the interface include:

- **Username Entry**: A text box for users to input their username, which is used to track their orders and trades.
- **Stock Selection**: A combo box for selecting stocks from a predefined list.
- **Order Submission**: Separate tabs for submitting buy and sell orders.
- **Order Lists**: List boxes displaying current buy and sell orders.
- **Stock Prices**: A list view showing the latest prices of various stocks.