**CSC 455 - RDS Project**

**“Broker-Lite”**

Bobby Palmer, Samnang Pann

**I. Motivation**

Provide fast, effective, and streamlined asset management to users (stock-brokers) for their customers. Display real-time market information to inform asset management decisions and provide up to date customer portfolio data. Provide tools for users to effectively manage assets through the buying and selling of stocks. Give users portfolio performance tracking tools which track and display customer portfolio performance over time.

**II. Functional Requirements**

i. Tools Utilized

1. SQLite v. 3.22.0 *Data storage, organization, retrieval, data processing.*

2. Java SE Development Kit 8 *Application front-end.*

3. JDBC-API *Back-end interface between application front-end “client” and database.*

4. Jersey-API *Back-end interface for retrieval of live market information.*

ii. Supported Operations

1. User authentication, registration, and deletion.

a. Prevent unauthorized access of user/customer data.

b. Provide tools for new users to register for application use.

c. Allow for safe deletion of users who deregister.

2. Customer registration and deletion.

a. Provide users ability to register new customers.

b. Protect customer data from unauthorized access by only allowing users

to view, and modify customers registered to them.

c. Allow for safe deletion of customers by authorized users.

3. Customer portfolio tracking.

a. Allow users to view portfolio data for all their registered customers.

b. Allow users to view customer portfolio performance over time.

4. Customer Asset Management.

a. Allow users to buy/sell stocks for their individual customers.

b. Prevent unauthorized transactions, that is, only allow users to purchase

assets for customers with enough cash.

c. Ensure customer data accurately reflects updated transaction data.

5. Live Market Tracking

a. Update current stock prices with internet retrieved live market data.

b. Ensure customer portfolios reflect current market data.

c. Ensure transactions are made using current market data.

d. Provide users tools for viewing current market performance, and track

overall market performance over time.

**III. User Interface Specification**

i. Broker-Lite will utilize Model-View-Controller design paradigm for front end.

1. Models:

a. User Model: Handles transactions between database and users.

b. Customer Model: Handles customer based transactions with database.

c. Stock Model: Handles all stock data transactions, with the database and

with live stock information API.

2. Views:

a. Views present information retrieved from models to the User.

3. Controller:

a. Controller accepts all user input, validates it, and updates models.

**IV. Database Scheme**