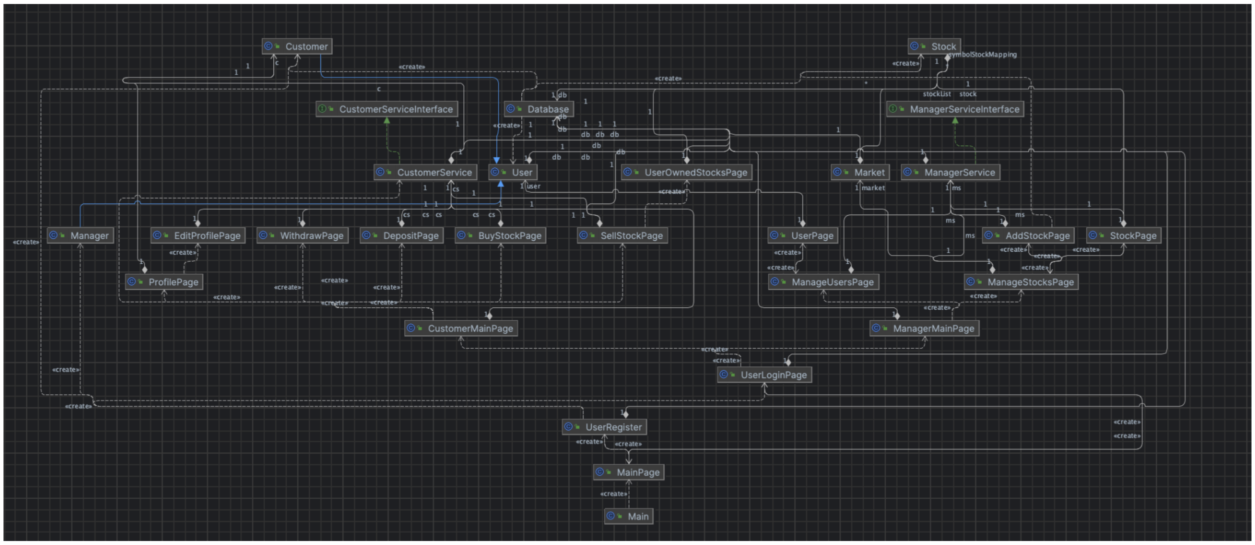
CS611-A4
Trading System

Saisriram Gunturu, Jinkun Lin, Yuanming Chai
sriramg@bu.edu,jchai23@bu.edu, jkunlin@bu.edu
U74142372, U43949722,U91747660

UML



Files

AddStockPage: Page which allows you to add stocks

BuyStockPage: Page which allows you to buy stocks

Customer: Class that maintains customers extends Users

CustomerMainPage: Main page for Customers

CustomerService: Concrete service interface for Customers and DB

CustomerServiceInterface: Abstract service interface for Customers and DB

Database: Main DB class that maintains all DB functionality

DepositPage: Page where Customers can deposit money

EditProfilePage: Users can edit their profile

Main: Entrance to application

MainPage: Main page of the application

Manager: Class that maintains Manager extends User

ManagerMainPage: Main page for Manager

ManagerService: Concrete service interface for Manager and DB

ManagerServiceInterface: Abstract service interface for Managers and DB

ManageStocksPage: Manager Page for managing stocks

ManageUsersPage: Manager page for managing users

Market: Class that maintains Market

ProfilePage: Page that shows profile for Customer

SellStockPage: Page that shows sell stock information
Stock: Class that maintains stock information
StockPage: Page that shows Stock information
User: Main User class that is extended by Customers and Managers
UserLoginPage: Login page for Users
UserOwnedStocksPage: Page that shows User owned stocks
UserPage: Page that lets Manager view User
UserRegister: Page for registering users
WithdrawPage: Page for withdrawing cash for user

Notes

Design:
Our program is designed to support customers and a unique manager to perform trading operations. We choose to use a SQL database with the pre-constructed schema to set up the persistent data storage for user data and stock information. We also create GUI interfaces to support users' interactions with our program. The GUI is connected with our Java backend classes and methods that are used to perform and execute the operations. We implemented separate service layers for the manager and customers. This design decouples the functionality of a unique stock market manager and customers, making sure each separate role can only access the operations that only belong to the role’s responsibilities. The manager will interact with the database (add/modify stocks/customers) with methods from ManagerService class and each customer will interact with methods (trade stocks/deposit/withdraw) from CustomerService class. We also implemented the singleton pattern for the manager account creation and the database so we are only using the same manager account every time the user logs in as manager and every user is accessing the same only database. We also implemented abstraction, encapsulation, and polymorphism in our structure creating interfaces for the service layer and user superclass for the manager and customers. Our program is scalable because of the separation of the user layer, service layer, and database layer which makes it easy to add additional features to each layer.

How to compile and run

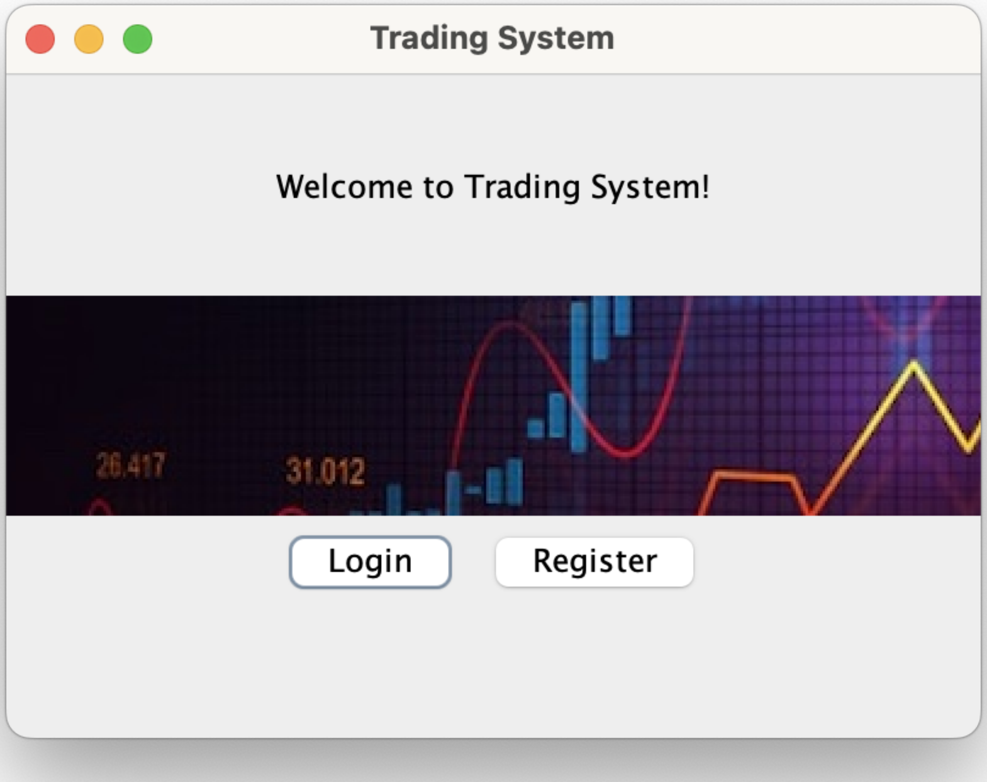
```
javac Main.java

java BeginGame
```

Input/Output Example

<Place here an example of how the program runs. Include both its outputs and correctly formatted inputs. Please clearly mark the inputs.>

1. Run main

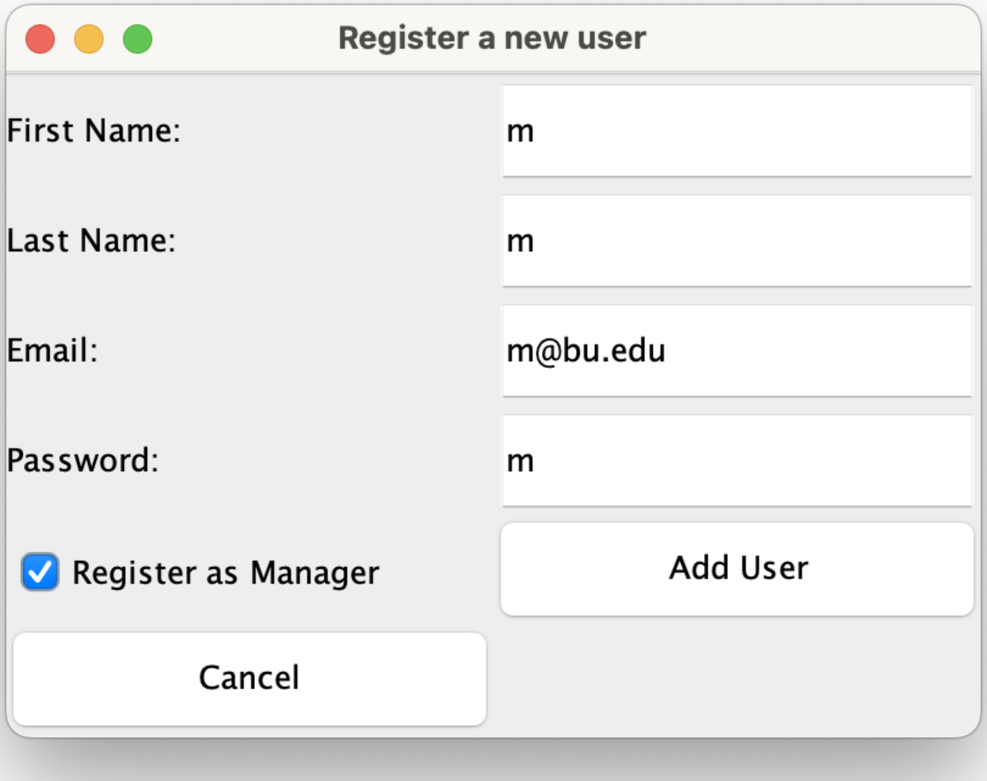
2. Main page



Trading System

Welcome to Trading System!

3. Press register. There is only one manager. So the Manager class has used Singleton pattern to ensure the uniqueness. Only when there is no manager, user can check the blank at the left of "Register as Manager". If there had been a manager in the database then this blank is deactivated. After the information has been added then press "Add User". After that, you will be automatically led to the login page.



Register a new user

First Name: m

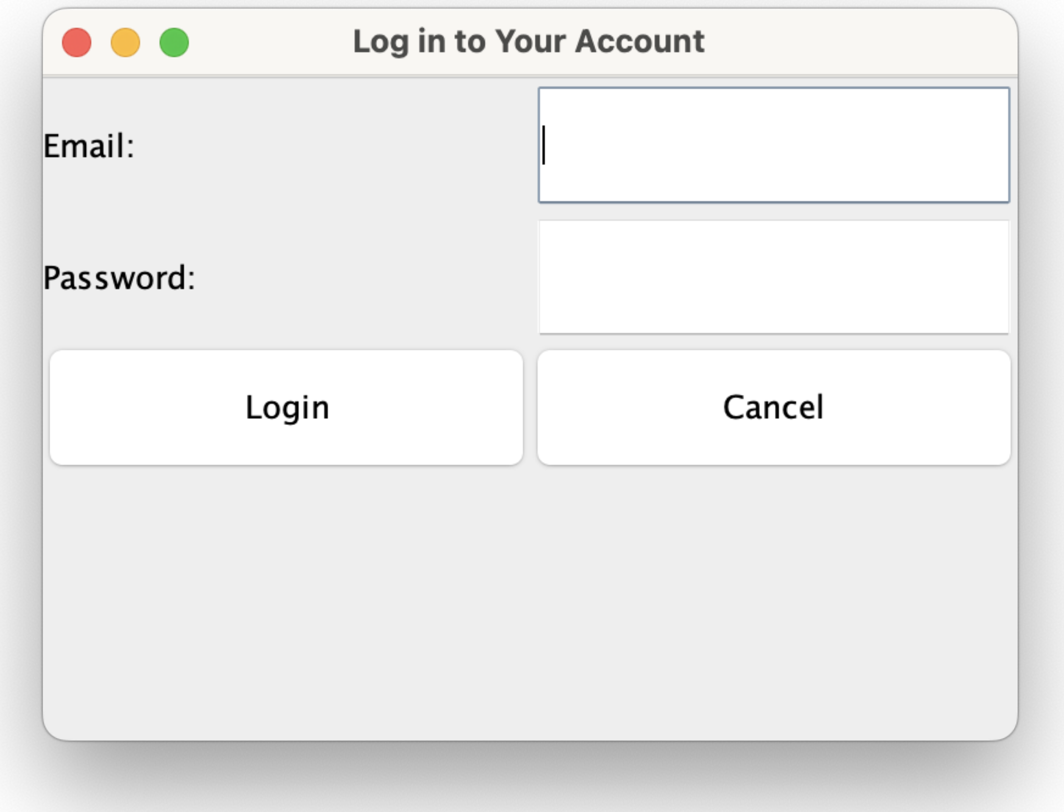
Last Name: m

Email: m@bu.edu

Password: m

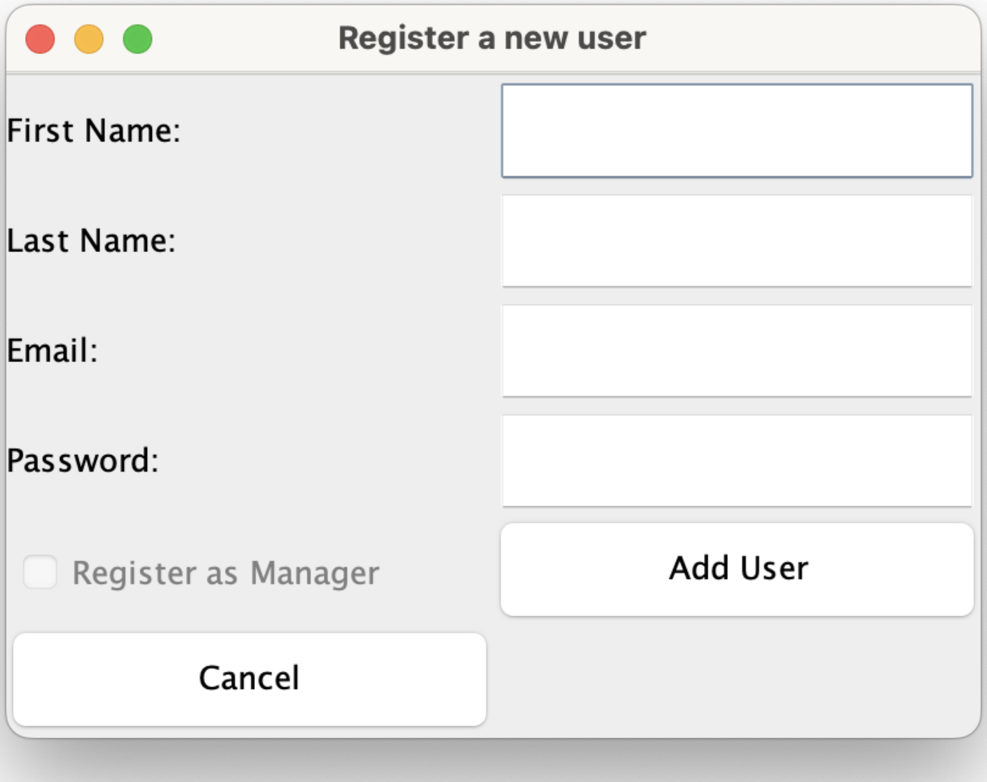
☑ Register as Manager

7. Login page. Press cancel to go back to the main page in Step 2.

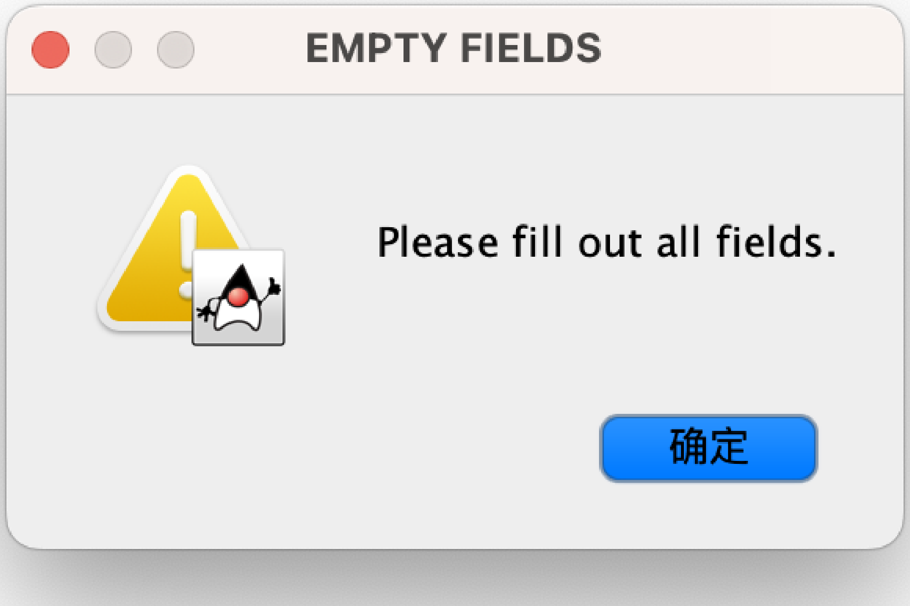


8. Next time when you click register on the main page you won't be able to register as manager anymore because there has been one in the database. The blank of "register as manager" would be deactivated. A string of "Manager is

there” will be outputted.

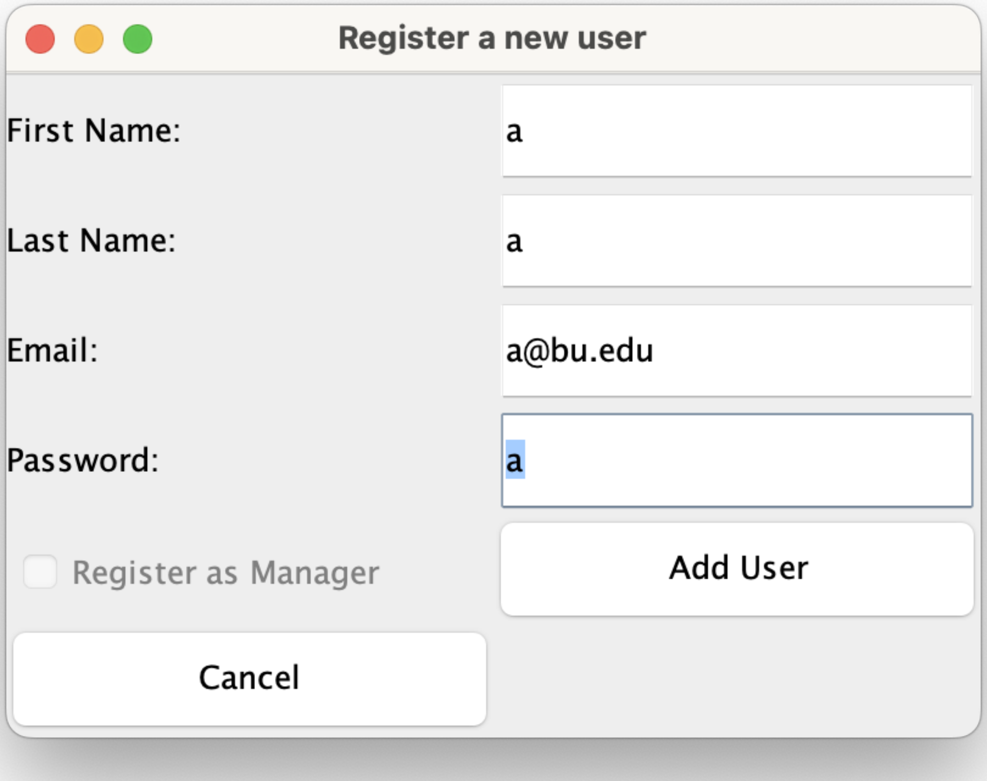


9. Write the information needed to register as a customer.If you leave some blank unfilled then a warning message will appear.



EMPTY FIELDS

Please fill out all fields.



Register a new user

First Name: a

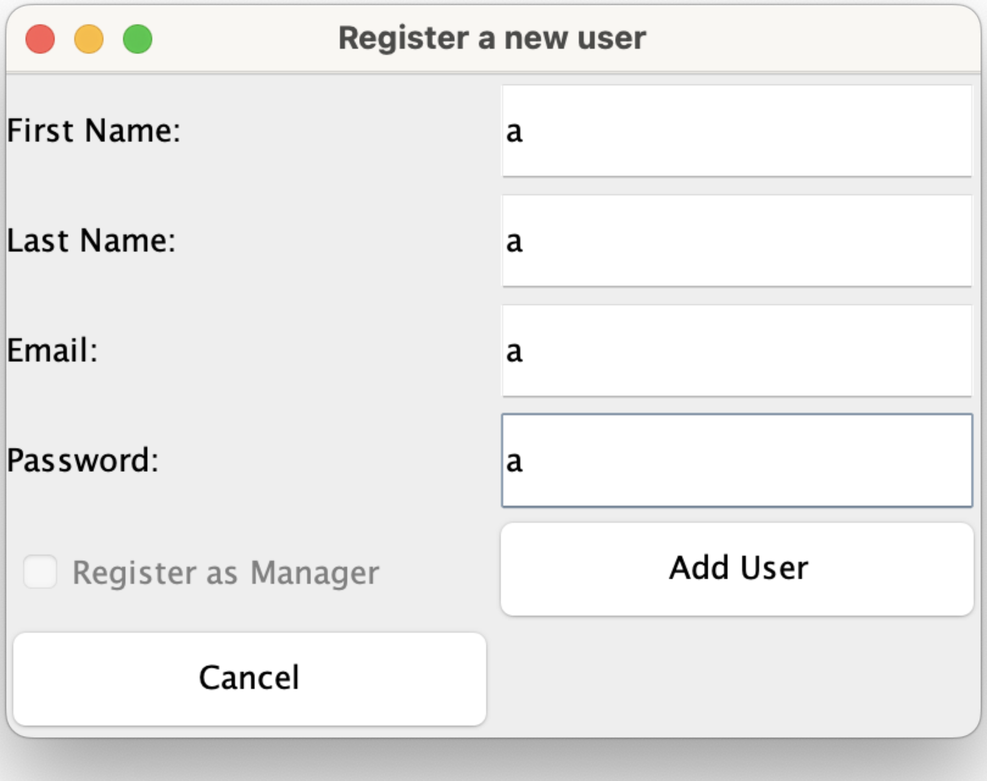
Last Name: a

Email: a@bu.edu

Password: a

□ Register as Manager

If you put in an invalid email without a "@", then when you press" Add User" an warning will be demonstrated.



Register a new user

First Name: a

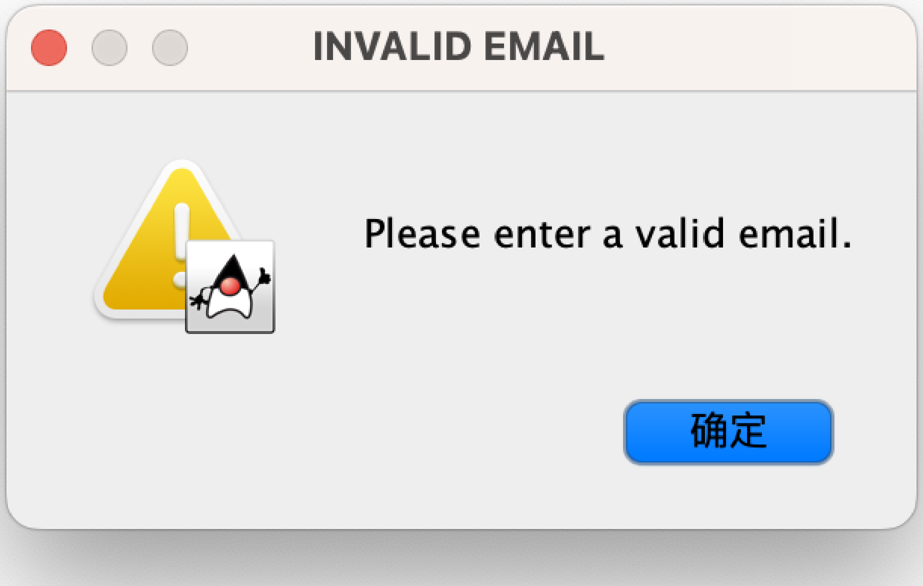
Last Name: a

Email: a

Password: a

☐ Register as Manager

10. invalid email format error report: Press "OK" to go back to the Register page.

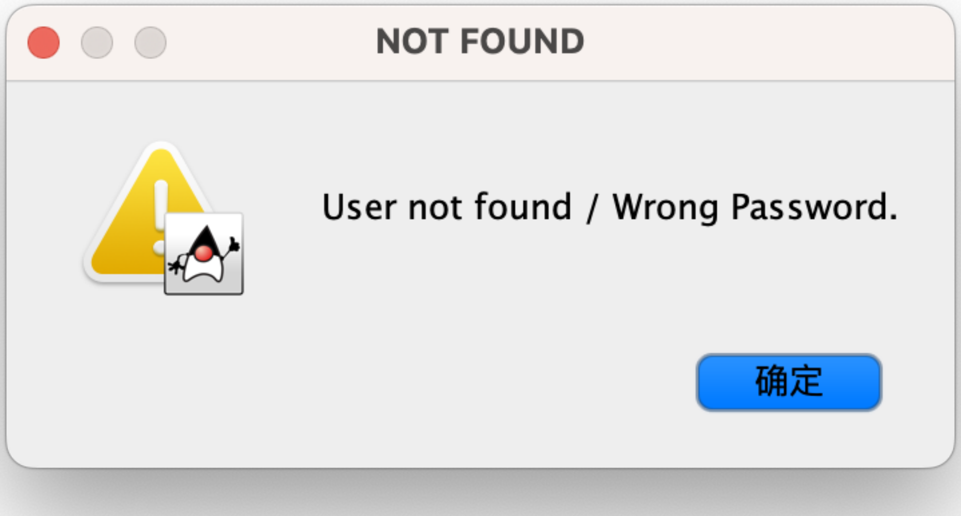

INVALID EMAIL

Please enter a valid email.

12. After a valid email is entered and the "add user" button is clicked. You automatically go to the Login page. Press cancel to go back to the main page.

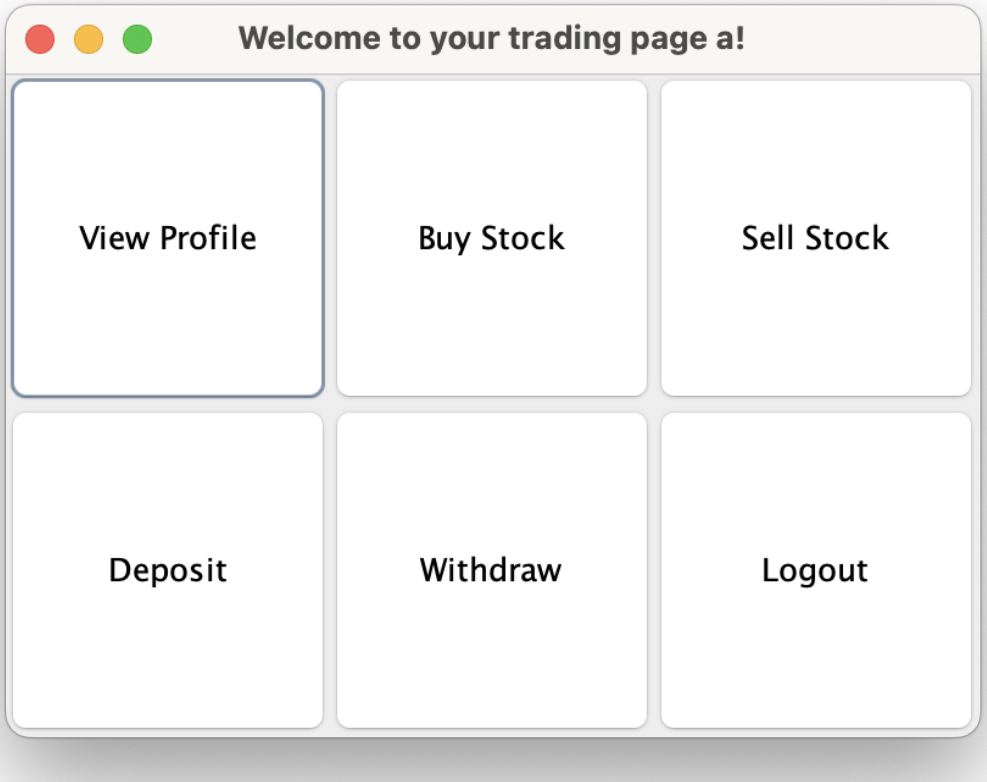
13. using the correct username and password to login.

13. if the email or password is wrong, then when press login, console outputs: "User not found. \n Error login to database." A warning message will also be shown.

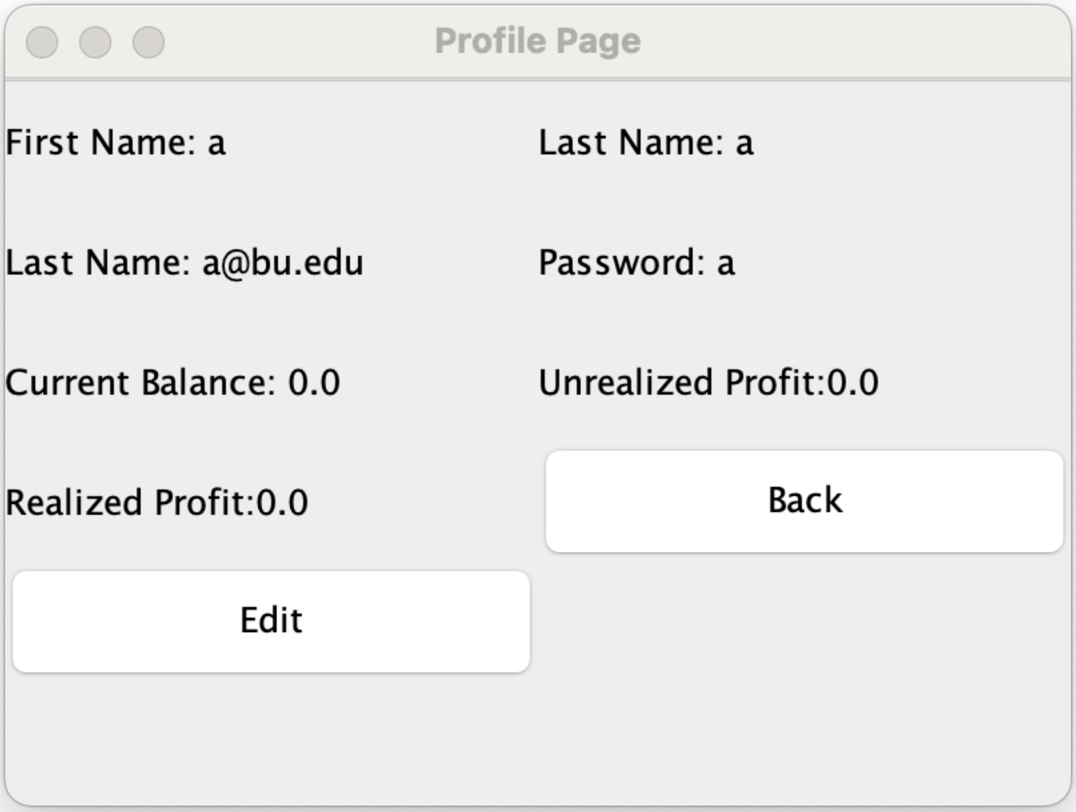

NOT FOUND

User not found / Wrong Password.

16. User's page when correct email and password was used to login. Console outputs users email and password and message "User successfully logged in!"! For the six button on the page of a customer, you can do different things. Clicking "View Profile" you'll be showed more information of this customer. Clicking "Buy stock" "Sell Stock" to buy from the market or sell from what the customer has bought. "Deposit" and "Withdraw" are used to add or withdraw money in the account.



17. press "View Profile" then we go to the profile page. It shows the customer's account information. Click "Edit" button to change the user's email or password. When a user is registered, he or she has no money in the account. He or she has to deposit money into the account. Current balance is the cash the user can immediately withdraw or buy new stocks with. Unrealized profit is the market value of all the stocks minus their costs based on the "First-in, First-out" principle. Realized profit is calculated by subtracting the total deposit from current balance. The total deposit is all the money you have put into the account using deposit and withdrawal buttons. It's also equal to the sum of all the individual balance of each stock.



Profile Page

First Name: a

Last Name: a

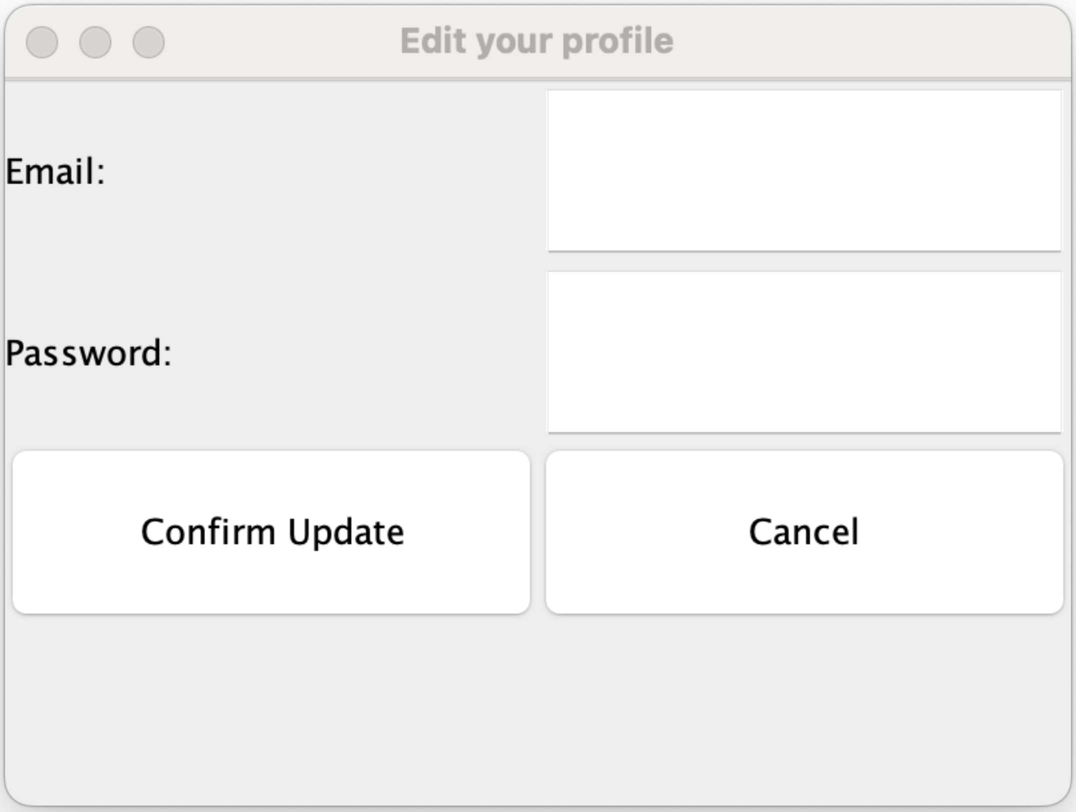
Last Name: a@bu.edu

Password: a

Current Balance: 0.0

Unrealized Profit:0.0

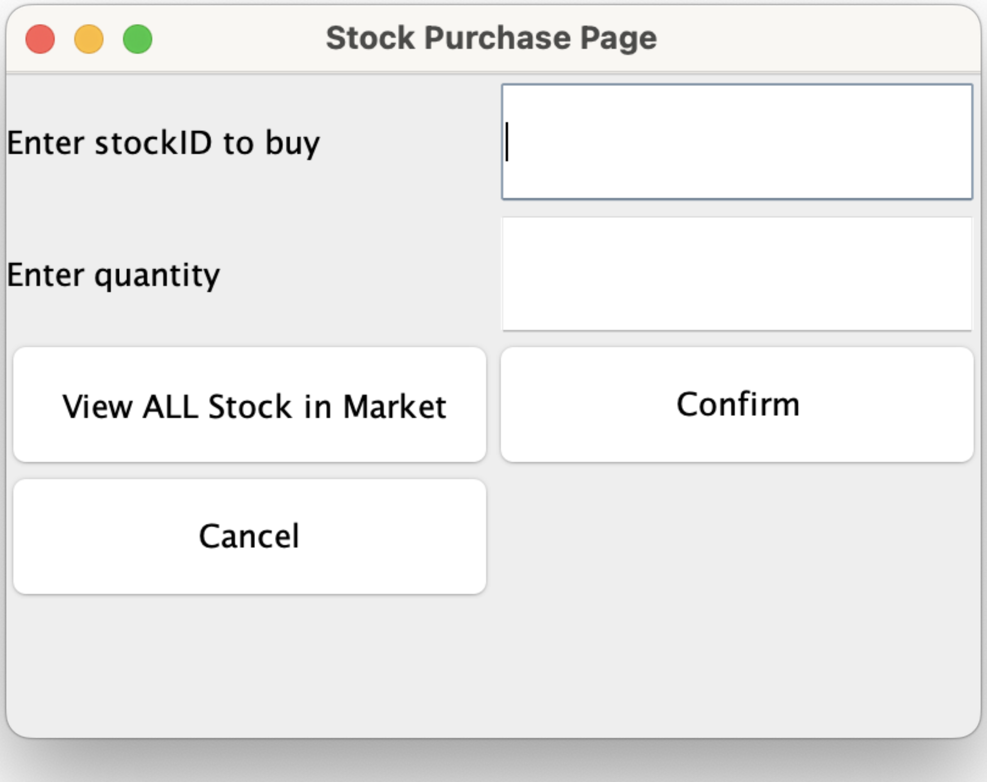
Realized Profit:0.0

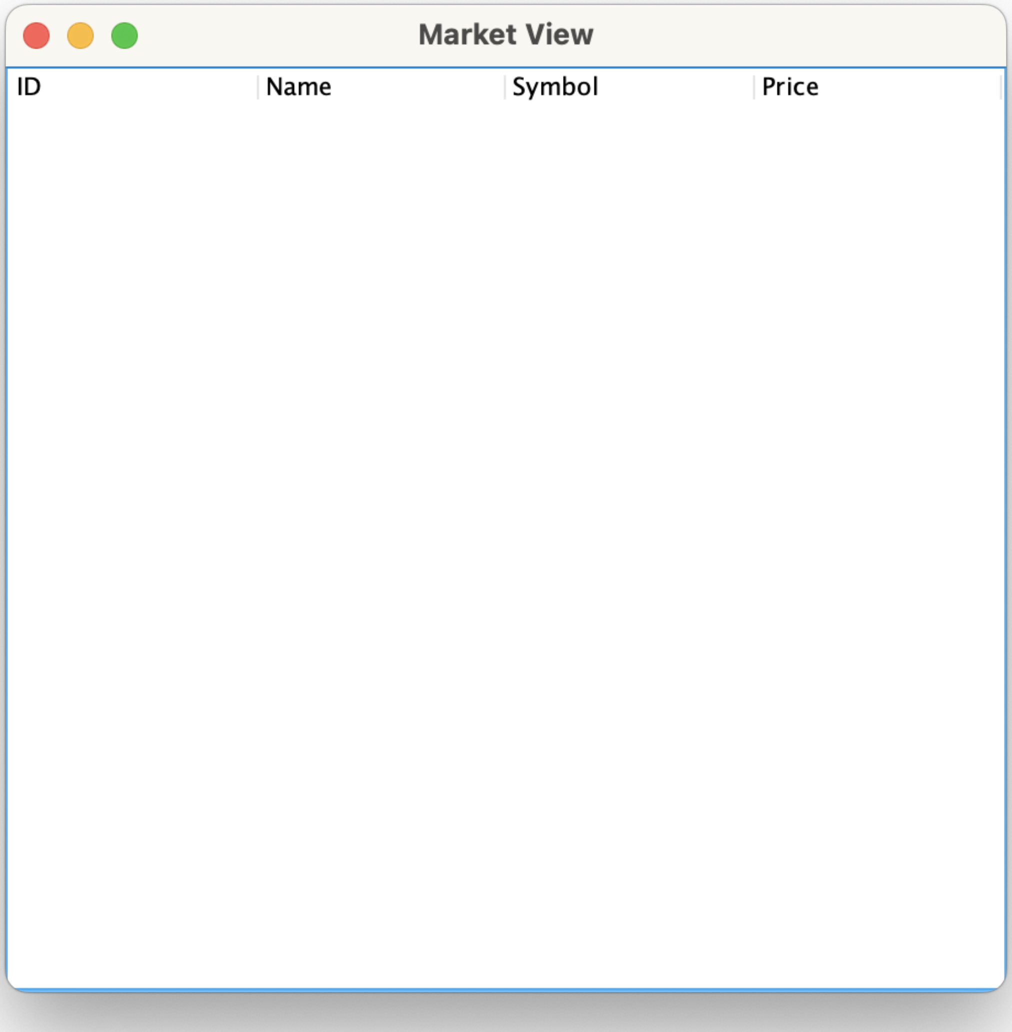


18. After inputting the updated email and password, click confirm update to confirm and go back to the user’s main page. If the new email is in invalid format the warning message of invalid email will be shown again.

19. Clicking “Buy Stock” button leads you to the buy stock page.You have a “View All Stock in the Market” button to go the a table of market see all the

stocks and their information.



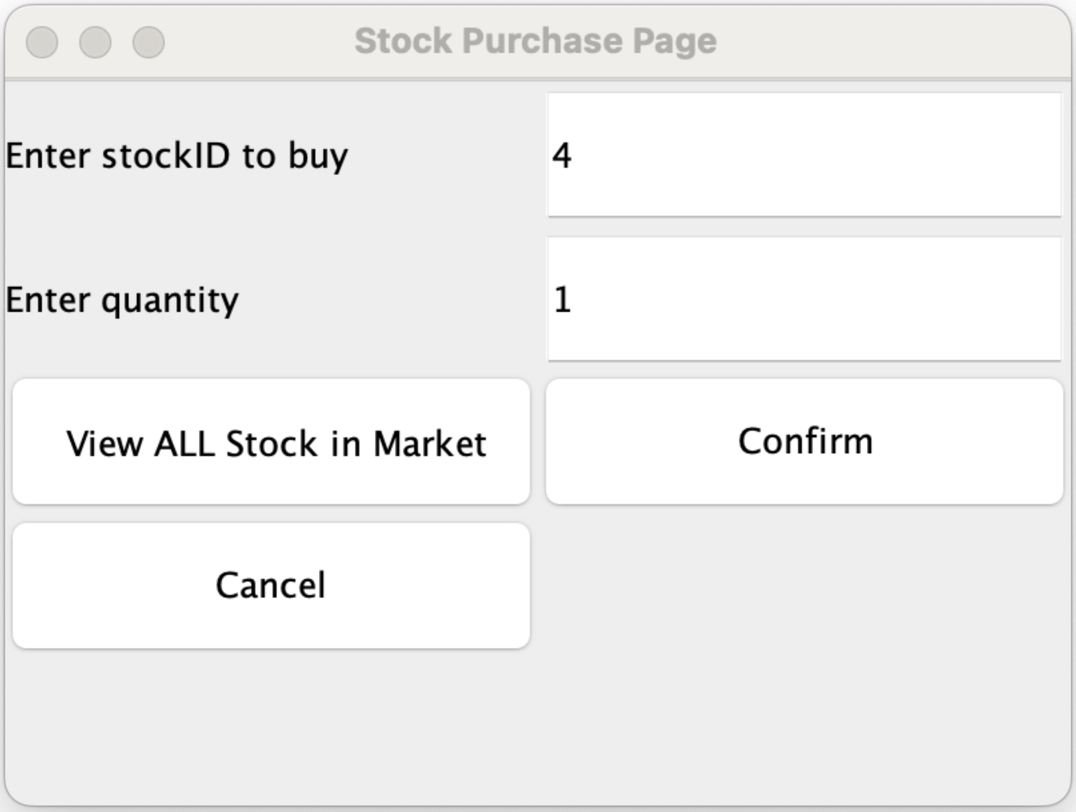


Market View

ID	Name	Symbol	Price

20. When the manager hasn’t add any stock to the market or no stock is active, the user can see no stocks from the market. If the customer enters an ID or

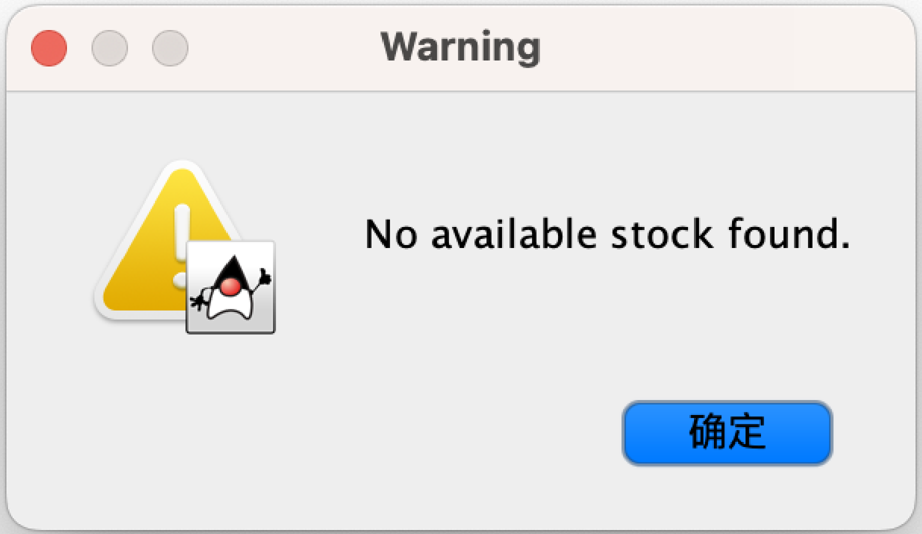
quantity that doesn’t exist in the market, a warning message will be displayed.



Stock Purchase Page

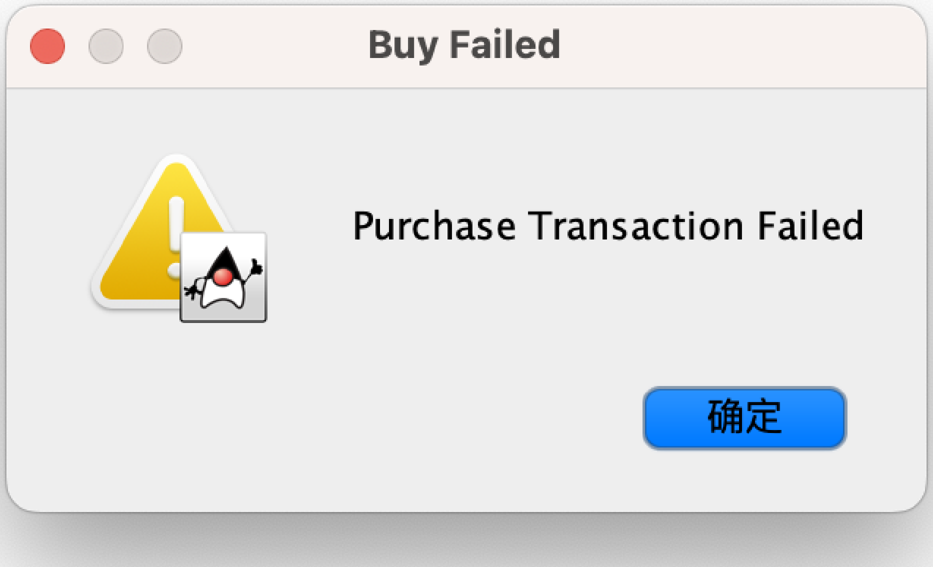
Enter stockID to buy: 4

Enter quantity: 1



No available stock found.

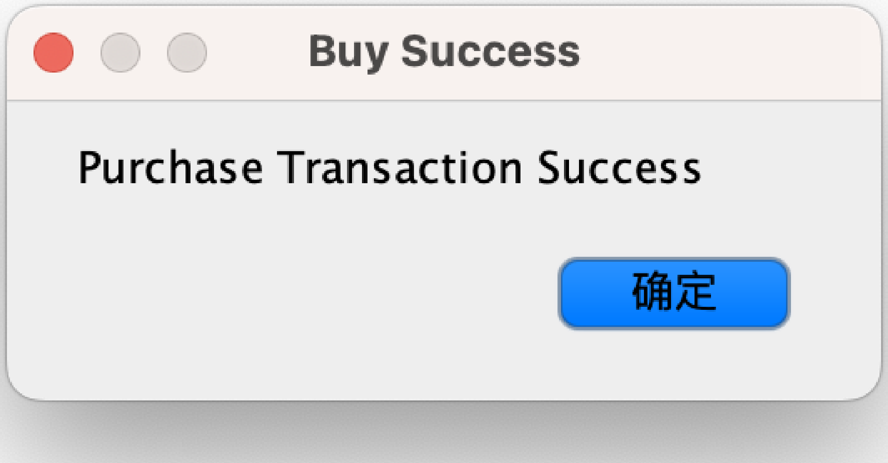
If the customer enters an ID whose corresponding stock is blocked by the manager or you don’t have enough money on the account to buy the stock, a warning message will appear. And the console outputs “You don't have enough money” to remind the customer



Buy Failed

Purchase Transaction Failed

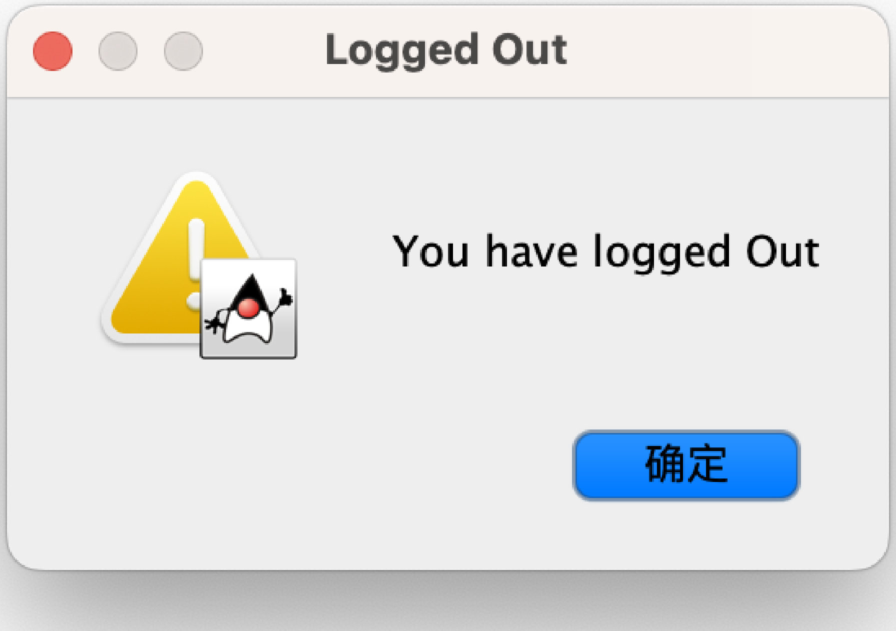
21. If the customer has enough money in his or her account then when pressing the confirmation button



Buy Success

Purchase Transaction Success

18. press log out. Press "OK" to go back to the main page.



Logged Out

You have logged Out

1. Check negative price and negative quantity of sell/buy/add/change price
2. Uniqueness of stock symbol
3. Price/quantity goes beyond the limit of Integer. 2^31
4. Approvement of users to allow/prevent them log-in
5.