STOCK MANAGEMENT SYSTEM

Team Members:
Jagrit Singh (17BCE0030)
Ifham Ali (17BCE0638)
Navya RG (17BCE2416)

Report submitted for the
Final Project Review of

Course Code: CSE3001 – Software Engineering

Slot: F1
Lab: L39 + L40

Professor: Swathi J.N

Contents:

1.INTRODUCTION

1.1 Theoretical Background

The project is about management of a stock system and portfolio tool that tracks all of your investment accounts, all in one place. A solution that not only caters to the need of a Stock Broking firm but is also scalable without compromising on performance. This system can be widely used by retailers, shopkeepers, manufacturing units and other merchants across different businesses. This is mainly designed for the manufacturing companies. It improves its business by facility of maintaining the record. Its password setting of the administrator helps to improve the security. It helps the user to view the details of the stock of various categories and the sales of their requirement. It will provide you with various sorts of information like- The name and number of the products that are available in the stock or when to reorder a particular product. A particular product's stock will be updated. The products whose sales have been high or low will be recorded. There are four main modules- PRODUCT MANAGEMENT, PURCHASE MANAGEMENT, SALES MANAGEMENT and USER MANAGEMENT.

STAKEHOLDERS

i. Stock Agent/Broker

1) Execute the buying and selling of shares.
2) Must view the current price of the stock, bid and accept for the next price of the stock available.

ii. Customers (who buys stocks)

1)Requests share purchases and selling.
2)Views the shares.

iii. Management

1) Supervises the price of the stock.
2) Generates the complete report.
3) Calculates the overall performance.

1.2 Motivation

Currently there are thousands of company shares that can be bought or sold. With the present population of the world, there are millions of transactions happening per second all over the globe. To facilitate these transactions, stock management firms are available. In these firms, it can be tedious to manually track all transactions and requests that are happening. It would be beneficial if there exists a system to reduce the complexity by letting the computer perform all financial calculations.

1.3 Aim of the proposed work

Our project aims to develop a web application that can used by the brokers, customers and admin in stock broking firms. It would help the brokers to easily track all requests made to them by their allotted clients and perform transactions to execute their orders. It would enable customers to place requests without actually contacting the broker and also enable the firm's administrators to view all owned shares, transactions and requests that have been and are being processed in the firm.

1.4 Objectives of the proposed work

The objectives of the work are as follows:-

- The objective is to create a stock management system with the perspective of a broker in which stocks can be easily managed , maintained and request to order conversion will take place
- The stock management system is required to monitor the day to day workings of the stock and perform calculations / exchanges with the stocks that are in place already.
- The efficient management of stock orders , requests and other entities will allow proper implementation of the software in question.We will use the software to mminimize human capital of the software firm and efficient

1.5 Report organization

Section 2 explains the existing models of stock broking systems.
Section 3 explains the functional, non-functional and system requirements of our project.
Section 4 explains the high level and detailed design.
Section 5 explains the implementation and testing of our project.

2.LITERATURE SURVEY

2.1 Survey of the existing models/work

There are a lot of stock broking firms that currently exist in India. Some of them are 5paisa, Zerodha, Angel broking, Sharekhan, Prostocks, Samco, Upstock, Fyers, Beeline, Aliceblue, Finvasia, Tradeplus, Edelweiss, ICICIdirect, HDFC securities etc. Customers prefer different stock broking firms according to the amount they charge for each executed order. These firms offer customized support and interaction in facilitation trades, managing portfolios, financial planning and wealth management services for clients. Clients are assigned to individual licensed stockbrokers and or financial advisors. Commissions at full-service brokerages are much higher than those at discount brokers as full-service brokers can provide expertise for people who don’t have the time to stay up-to-date on complicated issues such as tax or estate planning. However, those who want to just execute trades without extra services, prefer discount brokers.

3.PROPOSED SYSTEM REQUIREMENTS ANALYSIS AND DESIGN

3.1 Introduction

System Requirement Analysis

The purpose of this segment of the report is to describe the specific requirements of the Stock Management System project that are to be met by the implementation effort of JAGGU software. Included with the description of the requirements is a description of any constraints or assumptions that the project is working within.

Lastly, the purpose of this segment is to communicate the system attributes of the Stock Management software. These system attributes include reliability, availability, scalability, maintainability, and portability.

System design analysis

The purpose of this segment is to analyze the design of the Stock Management System project that are to be met by the implementation effort of JAGGU software. It would establish the overall structure of the software system, identify the the subsystems. The detailed description of how these subsystems are going to communicate with each other is also mentioned.

While writing this report, project dependencies were taken into account and are expressed clearly. The purpose of this project is dual-fold: to give detailed descriptions of the requirements of the customer and also state the performance requirements of the consumer. The standards while developing the software are also mentioned.

3.2 Requirement Analysis

3.2.1 Functional Requirements

3.2.1.1 Product Perspective

The stock management system that is to be developed by us is a web application that would facilitate stock exchange between traders. This system would be a solution that not only caters to the need of a stock broking firm but is also scalable without compromising on performance.

The stock management system would cater to the need of three stakeholders, namely- Stock agent/broker, customers (who buys stocks) and management. The customer will be able to request the broker to sell/purchase shares when the prices reach the quoted amount (quoting a stock). The broker is a person who would execute the stock exchange and the management will be able to view generated reports.

By implementing such a system, it would be easier to exchange shares as all the transactions are recorded in the database and potential errors that may occur can be avoided due to computerization of tasks.

3.2.1.2 Product Features

The follow is a table of the requirements that the system shall meet.

ID	Origin	Shall Requirement
1	*Customer*	The stock management system SHALL calculate performance and compare it with BSE and Nifty.
2	*Customer*	The stock management system SHALL facilitate stock search using any

		search engine.
3	*Customer*	The customers SHALL have remote access to the system.
4	*Customer*	The customer SHALL be able to create an account which will be used for performing transactions.
5	*Customer*	The customer SHALL be able to login to his/her account using the credentials derived during account creation.
6	*Customer*	The customer SHALL be allotted a broker upon request.
7	*Customer*	The customer SHALL be able to quote stock i.e., specify a price which if reached the broker shall sell/purchase shares.
8	*Customer*	The customer SHALL be automatically notified on the execution of his orders.
9	*Customer*	The customer SHALL be able to view all the shares owned by him.
10	*Customer*	The customer SHALL be able to report to the management, any grievances experienced by him.
11	*Broker*	The broker SHALL be able to create an account which will be used for executing the orders given by customers.
12	*Broker*	The broker SHALL be able to login to his account using the credentials derived during account creation.
13	*Broker*	The broker SHALL be able to view the available shares.
14	*Broker*	The broker SHALL be able to view the orders placed by his clients.
15	*Broker*	The broker SHALL be able to execute the selling/purchase of shares.
16	*Broker*	The broker SHALL be able to view the portfolios of his clients.
17	*Broker*	The broker SHALL be able to advice his clients.
18	*Administration*	The management SHALL be able to view reports containing shares owed by every customer in the stock broking firm.

3.2.1.3 User Characteristics

The following table identifies and describes the different users of the JAGGU software. The information gathered about the different users of the system helped define what the software needs to do.

User	**Description**
Customer	The customer is anyone who can quote stock. This is a very large group of users from all different backgrounds. Because of this, the system should be easy to use and confirm to commonly understood user interface styles for wide acceptance. The system should be able to be used on a wide range of popular system platforms to be able to meet the wide range of potential users.
Stock agent	The stock agent is anyone who executes transactions upon orders placed by customers. This is not a very large group of users from different backgrounds. These agents work in a stock broking firm and follow the rules created by the firm. They can access the system only from the office premises.
Management	The management will be able to view reports generated on the entire system and also to view complaints placed by customers.

3.2.1.4 Assumption and Dependencies

ASSUMPTIONS:

The following table lists the assumptions made by the requirements that define the software.

Assumption	**Description**

Consistency of database	The defined requirements assume that the transactions performed are real time.
Successful execution of requests.	The defined requirements assume that the broker will not ignore any of the requests.

DEPENDENCIES:

Inter-module Dependencies

Independent Modules

The following modules are independent and do not rely on any other modules to initiate them or to provide data. The modules pertaining to:

- Share Information Module.
- Ledger Balance Module.

Dependent Modules

The following modules are dependent on one another for their functioning.

Order Module: The Stock Management software accepts all the requests from the user, collects and gathers it to execute them against the market reality. Hence, this module is dependent on the market reality as well as on other modules such as- The Request module and the Customer module. Until a request ID is generated an order cannot be placed. Similarly, until a customer ID is validated an order cannot be generated.

Portfolio Module: This feature allows the user to maintain and view their portfolio by evaluating the current net value of the stock as well the growth of the stock value. Thus, this module is dependent on the Stock Information module and the Ledger Balance module. As the price and value of stocks keep changing and as the customer purchases or sells the shares, the portfolio is modified.

Request Module: This module collects the requests of the user regarding buying or selling of the shares and conveys it to the management system and it finally reaches the broker. Hence, this module is also dependent on the Stock Information module, because as the share value keeps varying, the buying and selling of shares which depends upon the rate is also affected and varied.

Inter-process Dependencies

As described earlier the main processes are the Customer process, the Broker process and the Request process. The Customer process depends on the broker process for accepting the request of purchase or selling of the shares on the basis of the Request ID for further placing the order. This is the most important inter-process dependency as it connects three important processes which form the basis of other processes.

Data Dependencies

The following Data Flow Diagram shows the data dependencies between the various entities and modules.

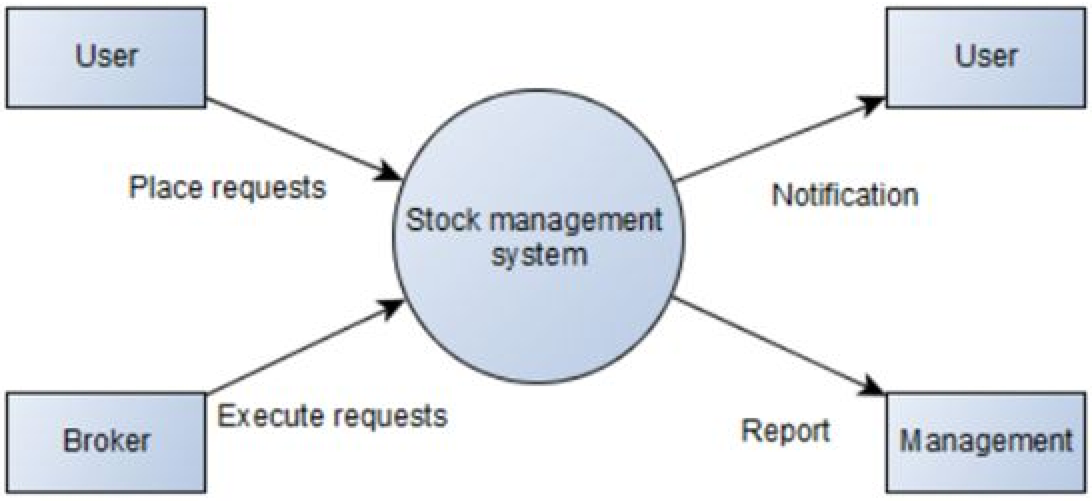


Figure 1, Level-0 data flow diagram

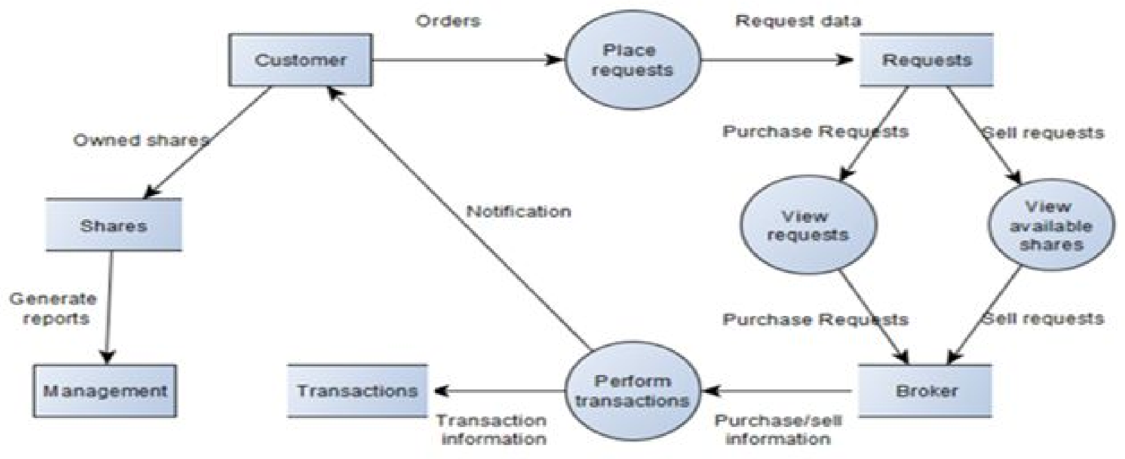


Figure 2, Level-1 data flow diagram

3.2.1.5 Domain requirements

The customers are required to know how a stock broking firm and the stock market functions. They are expected to know how share trading is performed. Brokers have to know the inner workings of the stock market so that he can advice the customer. The admin has to have good managing capabilities to manage the brokers and customers.

3.2.1.6 User Requirements and Product Specific System Requirements

USER REQUIREMENTS

ID	Origin	*User Requirement*
1	*Project Description Document*	*The broker SHALL be able to access the system only from the stock broking firm.*
2	*Customer*	*The customer SHALL be able to purchase only the shares that are for sale.*
3	*Customer*	*The customer SHALL sell shares only if there is a buyer for those shares.*
4	*Administration*	*Management SHALL be able to only view the status of the stock broking firm, it shall not make any modifications to the shares held by various customers.*
5	*Customer*	*The customer SHALL be intimated about the exact profit that goes to the stock broking firm and to himself.*

6	*Broker*	*The transactions that are carried out SHALL be real time in order to maintain consistency of the database.*
7	*Broker*	*The transactions SHALL remain consistent so as to not create new shares.*
8	*Customer*	*The system SHALL not restrict the content of the advice provided by stock brokers to clients.*
9	*Administration*	*The system SHALL automate adding/removing of shares to the portfolio upon performing of transactions.*
10	*Broker*	*The broker SHALL not perform transactions without the client giving orders.*

PRODUCT SPECIFIC SYSTEM REQUIREMENTS

LOGGING A REQUEST

Introduction

The Stock Management software shall allow a user to send a request regarding buying or selling of the shares.

Functional Requirements

Purpose: Logging a request for buying or selling of the shares.

Input: Request ID, Customer ID, Stock ID, quantity (number of shares and their rates), Date of expiry and Buy/Sell.

Processing: The system validates the stock ID, customer ID, quantity, date of expiry and then proceeds to log the request.

Output: The request is logged and the customer receives the confirmation.

Stimulus Response

A) User logs the request that is valid.

User Actions	**System Actions**
(1) User clicks the button to generate request.	
	(2) Drop down table with all input fields is displayed.

(3) User fills in the details (Customer ID, Stock ID, quantity, rate and date of expiry)	
	(4) The system validates customer ID.
	(5) The system validates stock ID
	(6) The system validates date of expiry.
	(7) If in case of sell, validates the quantity.
	(8) If in case of buy, validates whether the customer has the apt balance.
	(9) Request ID is then generated.
	(10) The system seeks for confirmation.
(11) The user approves the confirmation.	
	(12) The request gets logged.
	(13) Display a text message stating" Your request is logged".

B) User logs the request that is invalid.

User Actions	System Actions
(1) User clicks the button to generate request.	
	2) Drop down table with all input fields is displayed.
(3) User fills in the invalid details (Customer ID, Stock ID, quantity, rate and date of expiry).	
	(4) The system attempts to validate customer ID and the validation fails.

	(5) The system attempts to validate stock ID and the validation fails.
	(6) The system attempts to validate the date of expiry and the validation fails.
	(7) If in case of sell, the system attempts to validate the quantity and the validation fails.
	(8) If in case of buy, the system attempts to validate whether the customer has the apt balance and the validation fails.
	(9) Display a text message stating “Invalid input fields”.
(10) User goes to step (1).	

EXECUTING AN ORDER

Introduction

The Stock Management software accepts all the requests from the user, collects and gathers it to execute them against the market reality. The order is hence executed by the system.

Functional Requirements

Purpose: To gather all the requests at the certain moment and to execute them against market reality.

Input: Request ID, rate, quantity, ledger balance, market availability of stock and market rate.

Processing: The system validates the request ID, rate, quantity, ledger balance, market availability of stock and market rate.

Output: Order is confirmed, ledger balance is updated, portfolio details are updated and transaction log is updated.

Stimulus Response

A) Input fields are valid.

User Actions	System Actions

(1) Enter the stock name.	
(2) Enter the stock quantity.	
(3) Enter the stock price.	
	(4) The system validates whether the given price is acceptable or not along with the ledger balance, quantity of shares and date of expiry.
	(5) The order is executed by the system.
	(6) The system updates the ledger balance, portfolio and transaction log.
	(7) The system updates the portfolio details.
	(8) The system updates the transaction log.
	(9) Display a message stating "The ledger balance, portfolio details and transaction log is updated."

B) Input fields are invalid.

User Actions	**System Actions**
(1) Enter the stock name.	
(2) Enter the invalid stock quantity.	
(3) Enter the invalid stock price.	
	(4) The system attempts to validate whether the given price is acceptable or not along with the ledger balance, quantity of shares and date of expiry and fails to validate.

	(5) The order is not executed by the system.
	(6) Display a message stating "The ledger balance, portfolio details and transaction log are invalid."
(7) User goes to step (1)	

VIEWING YOUR PORTFOLIO

Introduction

This feature allows the user to maintain and view their portfolio by evaluating the current net value of the stock as well the growth of the stock value.

Functional Requirements

Purpose: Compute the net value and growth automatically.

Input: Customer ID and password.

Processing: The system updates the prices of the various stocks of a definite quantity for the calculation of the growth of the stock.

Output: Net value is calculated along with the growth and the portfolio is updated.

Stimulus Response

A) User inputs valid credentials.

User Actions	**System Actions**
(1) Enter the customer ID.	
(2) Enter the password.	
	(3) System checks if the inputted credentials are valid and the user is genuine.
	(4) System then calculates the net value of the stock.

	(5) System also calculates the growth of a particular stock as the stock value varies.
	(6) Display a message stating the net value with its growth.

B) User inputs invalid credentials.

User Actions	**System Actions**
(1) Enter the invalid customer ID.	
(2) Enter the invalid password.	
	(3) System attempts to check the inputted credentials and fails to validate.
	(4) The system does not compute the net value and the growth.
(5) User goes to step (1).	

LOGGING A NEW PRODUCT

Introduction

A new share product can be logged on the system with the broker who has to login a new share.

Functional Requirements

Purpose: Log in a new share / new security in the system

Input: Share name and Share ID.

Processing: Validate the details and update the database

Output: Current price of the share, quantity of share available as well as the ceiling and floor value.

Stimulus Response

A) The user inputs valid share details.

User Actions	**System Actions**
(1) Enter the share name.	
(2) Enter the share ID.	
	(3) The system validates the share details.
	(4) The system displays the current price of the share.
	(5) The system computes the quantity of such shares available.
	(6) The system displays the floor value and the ceiling value of the particular share in last 52 weeks (1 year).

B) The user inputs invalid share details.

User Actions	**System Actions**
(1) Enter the share name.	
(2) Enter the invalid share ID.	
	(3) The system attempts to validate the share details and fails.
	(4) The system does not display the current price of the share
	(5) The system does not compute the quantity of such shares available.
	(6) The system does not display the floor value and the ceiling value of the particular share in last 52 weeks (1 year).
	(7) The system displays a message stating “Input valid details of the shares”.

(8) User goes to step (1).	

LOGGING A NEW CUSTOMER

Introduction

A new customer has to be logged in the software database which will guarantee a fresh customer being entered and the system expanding.

Functional Requirements

Purpose: Logging in a new customer in the database

Input: Customer name, Email ID, Phone number and Date of Birth.

Processing: Computes the age of the customer using his Date of Birth and if the age is less than 18 then he is not permitted to invest.

Output: Customer ID, age of the customer and the ID of the broker.

Stimulus Response

A) Valid Customer age (If age>=18)

User Actions	System Actions
(1) Enter the customer name.	
(2) Enter the customer Email ID.	
(3) Enter the phone number of the customer.	
	(4) Check if the inputted details are valid and new.
	(5) Compute the age of the customer and if above 18, generate various details.
	(6) Generate customer ID.

	(7) Generate broker ID.

B) Invalid Customer age (If age<18)

User Actions	System Actions
(1) Enter the customer name.	
(2) Enter the customer Email ID.	
(3) Enter the phone number of the customer.	
	(4) Check if the inputted details are valid and new.
	(5) Compute the age of the customer and if less than 18, display appropriate message.
	(6) Display a message stating "The customer is under-aged for investments."
(7) The customer goes to step (1).	

3.2.2 NON-FUNCTIONAL REQUIREMENTS

Design Constraint	**Description**

Reliability	JAGGU software guarantees a failure rate of less than 1% barring which a fixed fee will be paid to the customer as damages. JAGGU software will take care of any failure and downtime of the system will not exceed 24 hours. Failure of which will result in additional penalties as mentioned in the document.
Security	SHALL only accept connections from the Spine Secure Proxy (SSP) SHALL authenticate the SSP prior to responding to any requests using its client certificate SHALL only permit approved supported ciphers to be utilised
Portability	JAGGU software guarantees that 70% of the code will be portable and can be used in any other system with no changes whatsoever Further support will be provided to extend the functionality of the software to a maximum up to 10 systems. The charges will be levied separately as discussed in contract.

3.2.3 SYSTEM REQUIREMENTS

3.2.3.1 Hardware Requirements

The system is presumed to have a working hardware for the working of our software. The exact specifications of the system hardware are as follows:-

- Processor (CPU) with 2 gigahertz (GHz) frequency or above
- A minimum of 2 GB of RAM
- Monitor Resolution 1024 X 768 or higher
- A minimum of 20 GB of available space on the hard disk
- Internet Connection Broadband (high-speed) Internet connection with a speed of 4 Mbps or higher
- Keyboard and a Microsoft Mouse or some other compatible pointing device
- Sound card
- Speakers or headphones
- **Strongly Recommended** -High Resolution Screen with interactive capabilities

3.2.3.2 Software Requirements

The exact software requirements pertain to the browsing needs of the consumer while he uses the software.The browsing needs of the customer are as follows:-

- Chrome* 36+
- Edge* 20+
- Mozilla Firefox 31+
- Internet Explorer 11+ (Windows only)
- Safari 6+ (MacOS only)

4. DESIGN OF THE PROPOSED SYSTEM

4.1 INTRODUCTION

This segment extends the design discussion and describes the design for each system module in more detail. A UML Class diagram is included for each module design discussion. This is followed by a description of the data requirements for each module and the design of those data elements.

4.2 HIGH LEVEL DESIGN

STATIC STRUCTURAL MODEL

The static structural model that would be most suitable for the stock management system is the client-server architecture. The client-server model is used for systems which are widely distributed. Here, the clients request services and the servers provide the service. Since the system is distributed, a network has to be created to establish communication between the clients and the server. In our case, The customers are located in remote places whereas the brokers are present in the stock broking firm from where they process transactions and other services. So a wide area network such as the internet has to be created to enable communication between the customer and his broker.

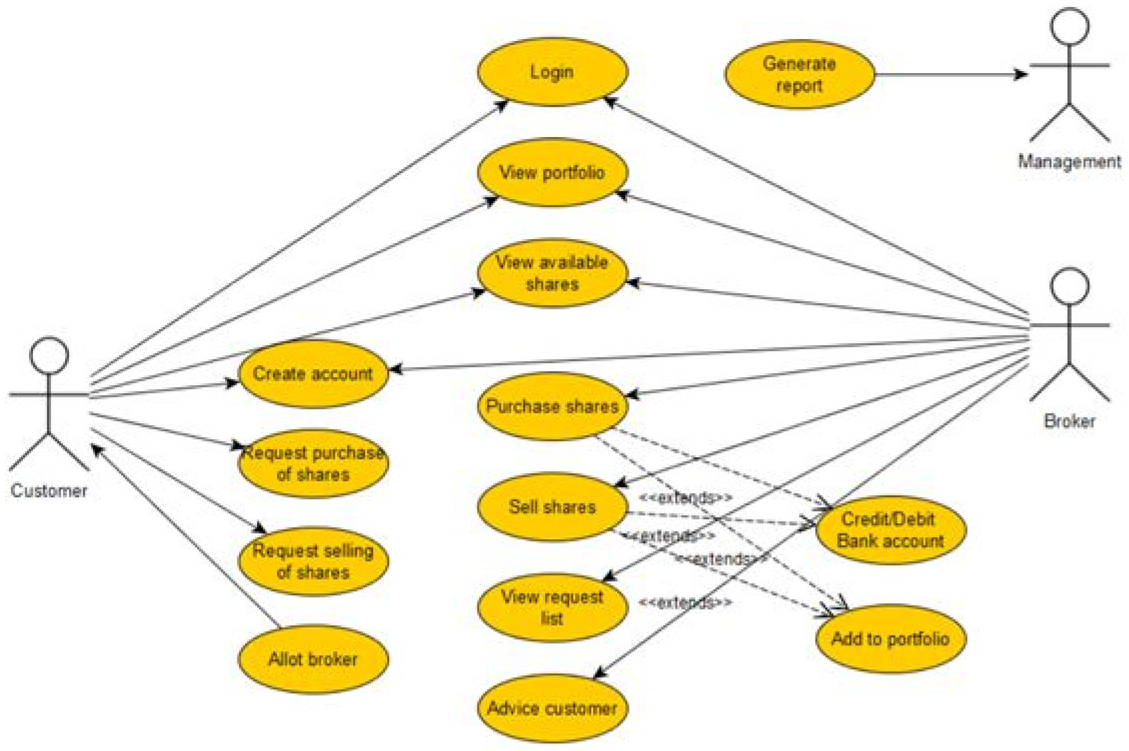
In the stock management system, distribution of data is uniform, the network can be used effectively, hardware is available for a cheap price and is easily scalable.

CONTROL OF SUBSYSTEMS

The manager model would be most suitable for the control of subsystems in our stock management system because it is a system where transactions have to be processed concurrently in real time to prevent ledger imbalance. Here, one system component (the server in the stock management firm) controls the stopping, starting and the coordination of other system processes. The server here is the central controller that manages the execution of the processes.

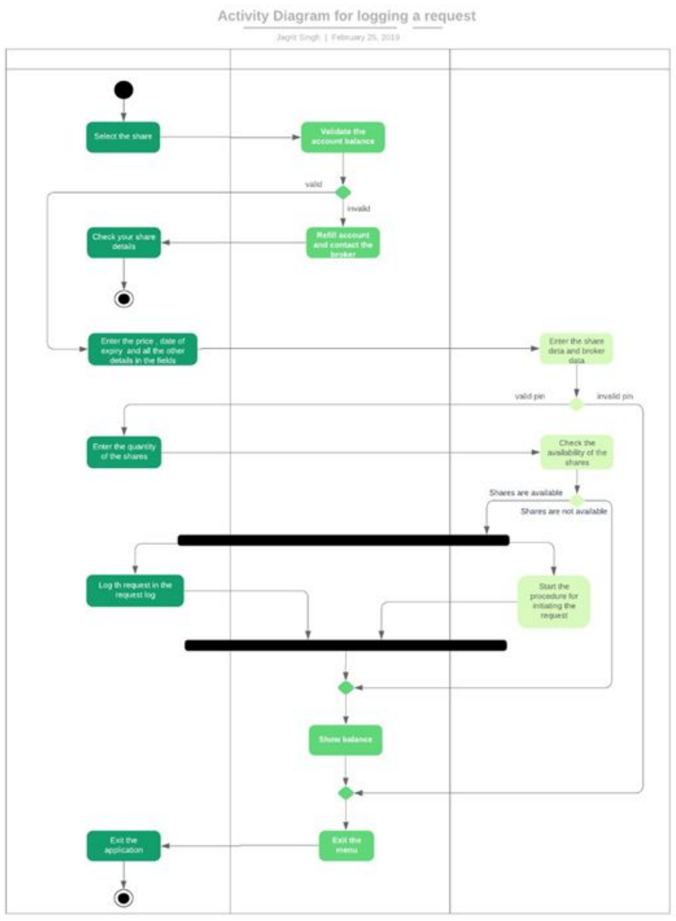
4.3 DETAILED DESIGN

<u>Usecase Diagram</u>



Activity Diagram

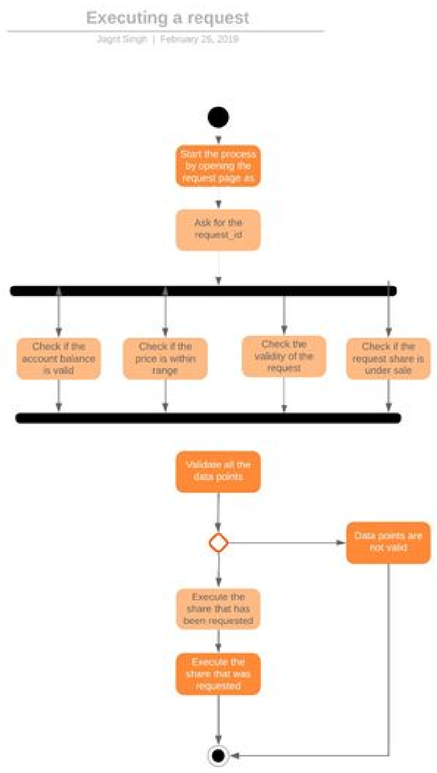
Logging a request


Activity Diagram for logging a request
Jagrit Singh | February 25, 2019
Select the share
Validate the account balance
valid
invalid
Check your share details
Refill account and contact the broker
Enter the price , date of expiry and all the other details in the fields
Enter the share deta and broker data
valid pin
invalid pin
Enter the quantity of the shares
Check the availability of the shares
Shares are available
Shares are not available
Log th request in the request log
Start the procedure for initiating the request
Show balance
Exit the application
Exit the menu

Activity diagram for logging a request

The beginning of the procedure starts with the system validating the account balance and the number of shares .Lack of either can imply that the user does not have the sufficient funds or the sufficient amount of shares to execute the order that follows.Then the order is followed to the request table.The request table validates the legitimacy of the request and then logs it.

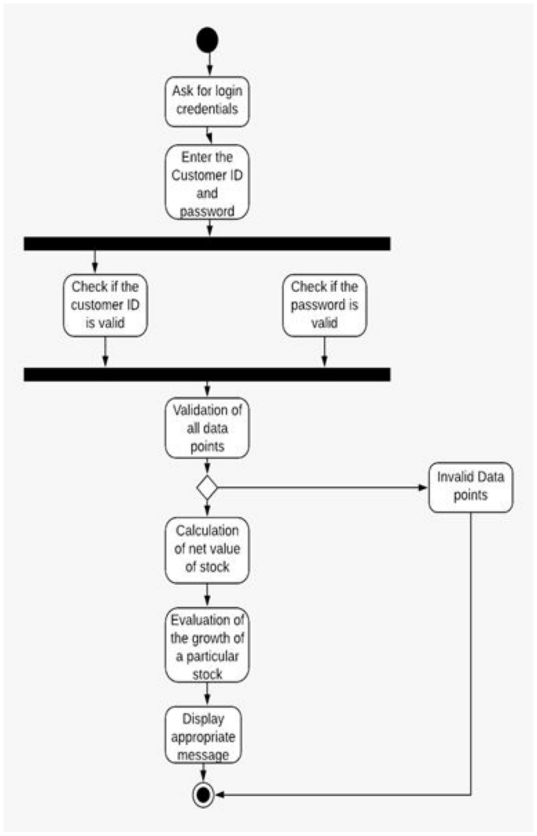
Executing a request


Executing a request
Jagrit Singh | February 26, 2019
Start the process by opening the request page as
Ask for the request_id
Check if the account balance is valid
Check if the price is within range
Check the validity of the request
Check if the request share is under sale
Validate all the data points
Data points are not valid
Execute the share that has been requested
Execute the share that was requested

Activity diagram for executing a request

The execution of the request is done by the broker himself.One the requests are locked , no changes can be made to the request , hence the request is permanently locked in the table.We can see that the request is permanently locked within the system.The requests locked are then processed immediately.The broker sees the viability of the requests and then validates them according to the current market specifications.

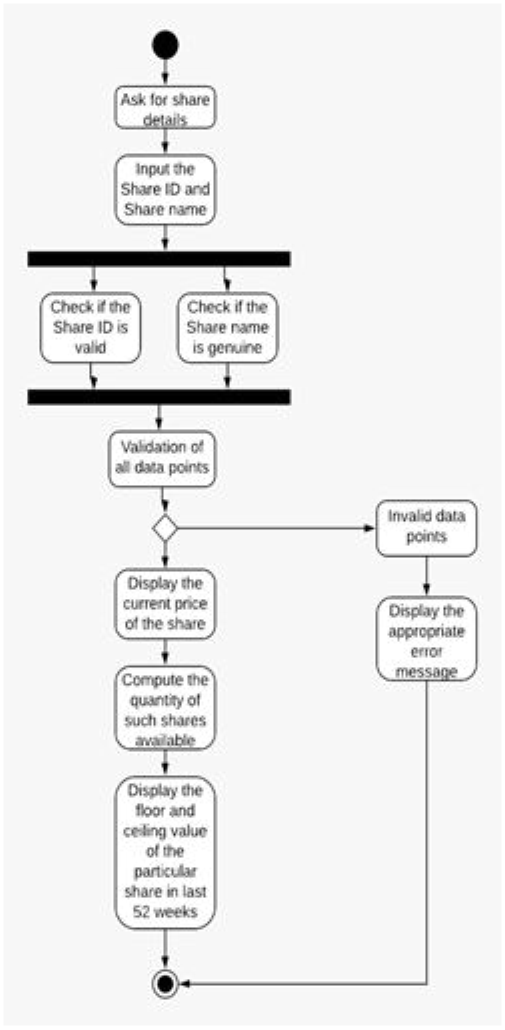
Viewing your portfolio


Ask for login credentials
Enter the Customer ID and password
Check if the customer ID is valid
Check if the password is valid
Validation of all data points
Invalid Data points
Calculation of net value of stock
Evaluation of the growth of a particular stock
Display appropriate message

Activity diagram for viewing your portfolio

To view the portfolio we have to first enter the data of the login credentials.After the credentials are verified and the project is given a detail , we move to the next step.The net quantity of the stock is taken from the orders table and the value is retrieved from the share table.Thus the total current value of the portfolio is calculated.

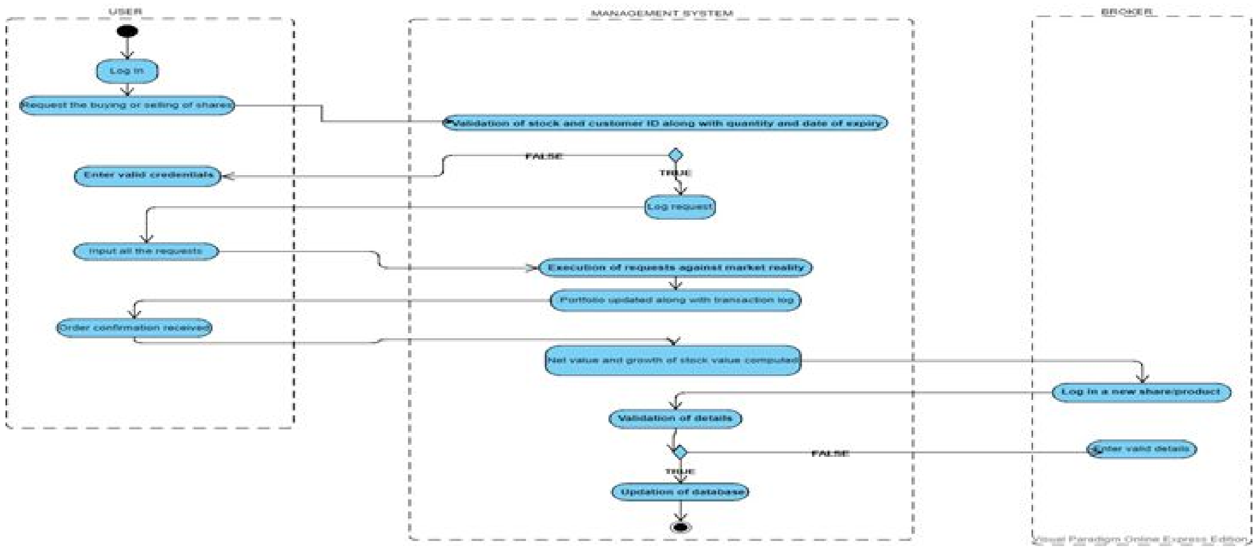
Logging a new product


Ask for share details
Input the Share ID and Share name
Check if the Share ID is valid
Check if the Share name is genuine
Validation of all data points
Invalid data points
Display the current price of the share
Display the appropriate error message
Compute the quantity of such shares available
Display the floor and ceiling value of the particular share in last 52 weeks

Activity diagram for logging a new product

The shares are updated from the administrators side only , the administrators are responsible for personally updating the value of the shares against the value of the market.The available quantity of the shares are also updated regularly by the administrators.This locking away of data is necessary because we need to prevent malpractice from the users side.

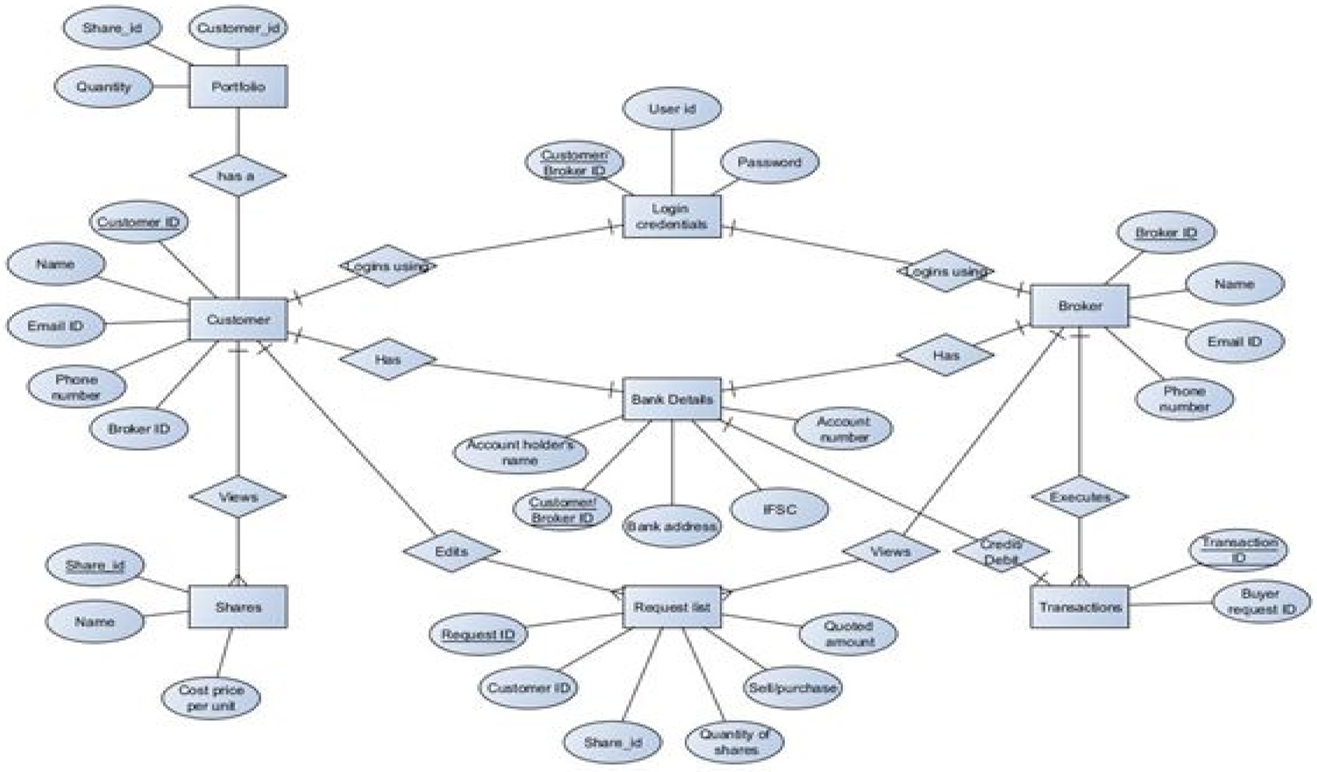
Updation of database


USER
MANAGEMENT SYSTEM
BROKER
Log in
Request the buying or selling of shares
Validation of stock and customer ID along with quantity and date of expiry
FALSE
TRUE
Enter valid credentials
Log request
Input all the requests
Execution of requests against market reality
Portfolio updated along with transaction log
Order confirmation received
Net value and growth of stock value computed
Log in a new share/product
Validation of details
FALSE
Enter valid details
TRUE
Updation of database
Visual Paradigm Online Express Edition

Activity diagram for updation of database

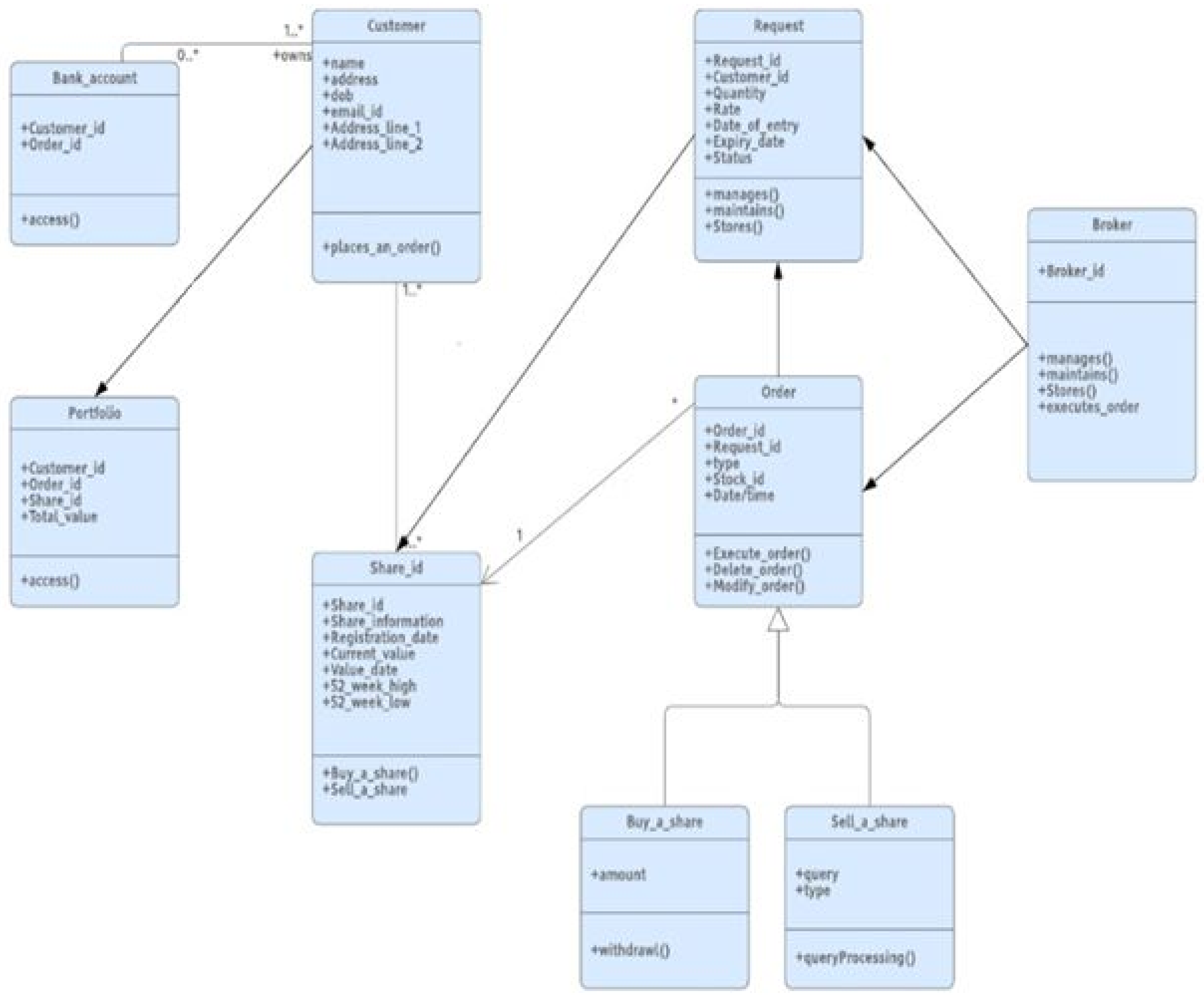
The updated database is a very complex process.Each of the actors on the scene are involved are then updating the database.The updated database is involved in the case of each actor registering their own database.The customer creates a new customer_id , the administrator creates new shares and the brokers adds and deletes the data.

ER Diagram



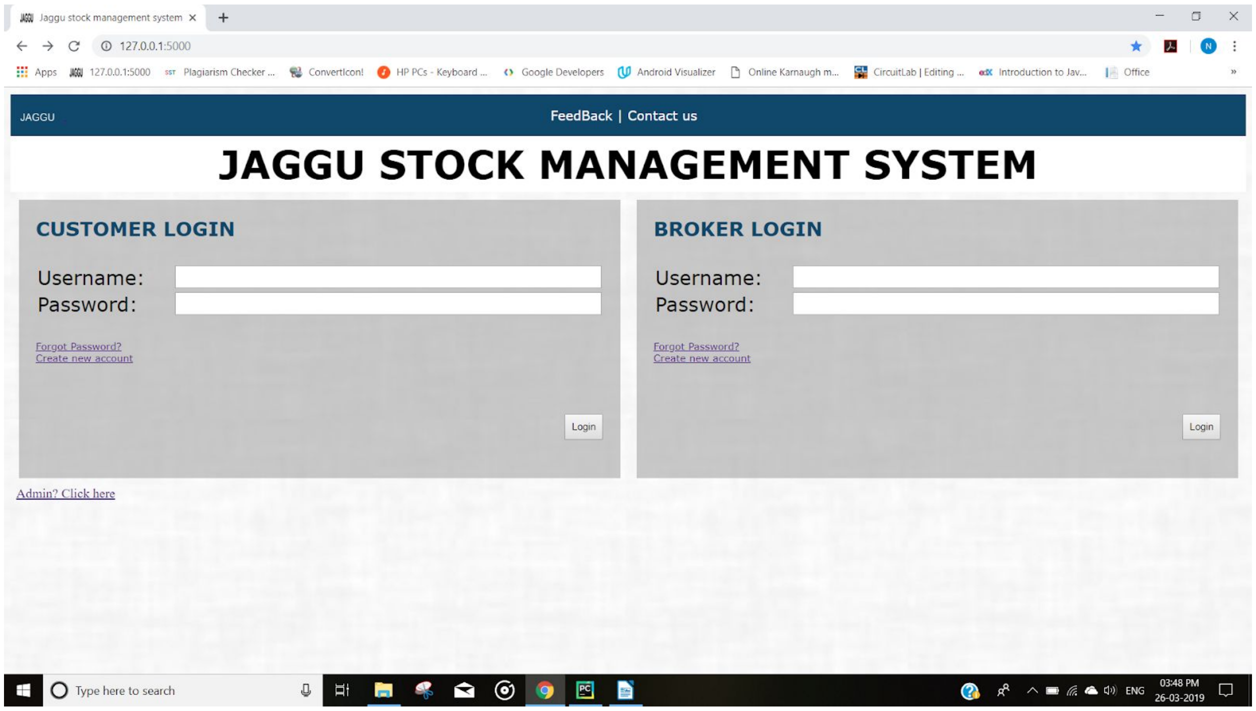
Share_id
Customer_id
Quantity
Portfolio
has a
User id
Customer/ Broker ID
Password
Login credentials
Customer ID
Name
Email ID
Phone number
Broker ID
Customer
Logins using
Logins using
Has
Has
Broker ID
Name
Email ID
Phone number
Broker
Bank Details
Account holder's name
Customer/ Broker ID
Bank address
IFSC
Account number
Views
Edits
Views
Credit/ Debit
Executes
Share_id
Name
Shares
Cost price per unit
Request list
Request ID
Customer ID
Share_id
Quantity of shares
Sell/purchase
Quoted amount
Transactions
Transaction ID
Buyer request ID

Class Diagram



5. IMPLEMENTATION AND TESTING

Implementation



JAGGU STOCK MANAGEMENT SYSTEM

CUSTOMER LOGIN

Username:
Password:

Forgot Password?
Create new account

Login

BROKER LOGIN

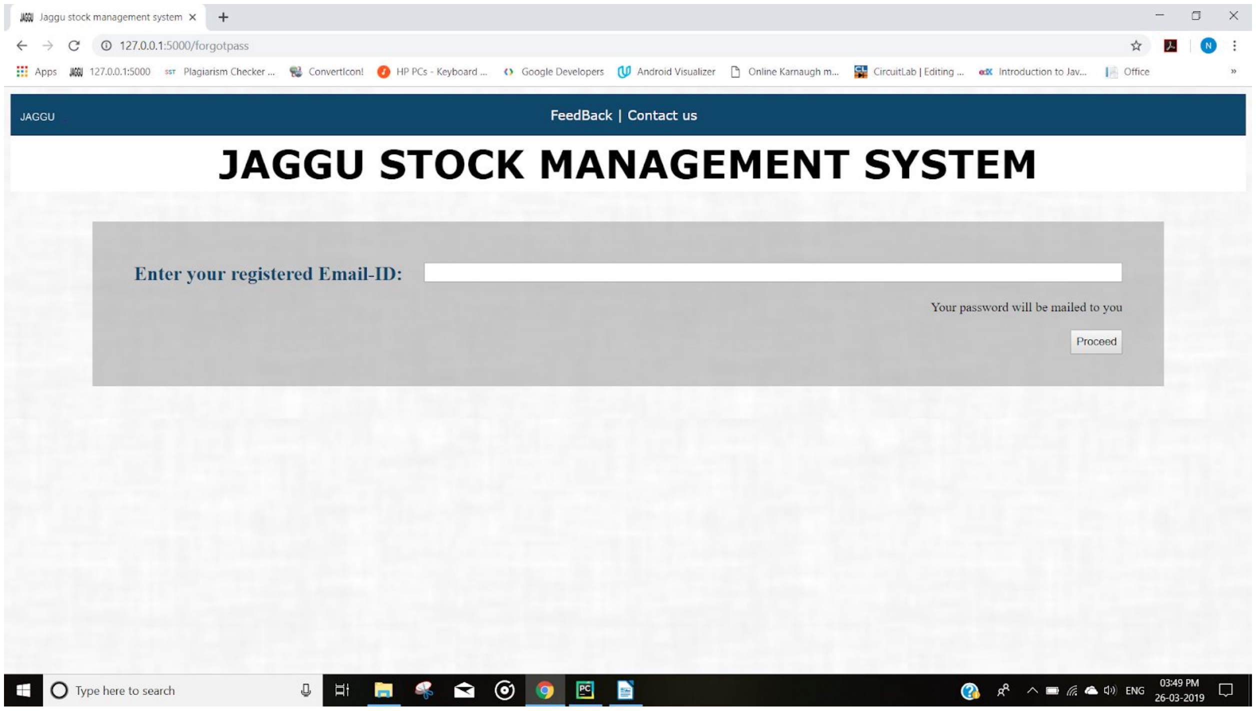
Username:
Password:

Forgot Password?
Create new account

Login

Admin? Click here

Homepage of the stock management system where the customer and broker can login. Admin can view all the ongoing requests, transactions and owned shares.



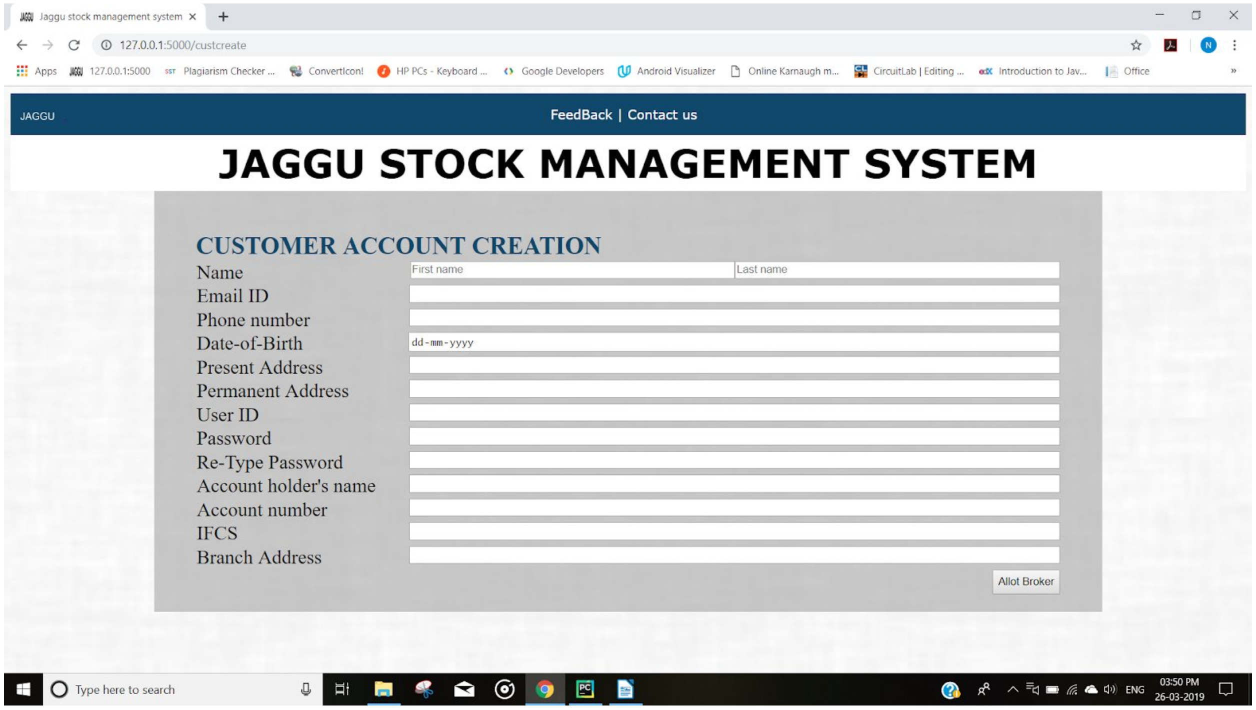
JAGGU STOCK MANAGEMENT SYSTEM

Enter your registered Email-ID:

Your password will be mailed to you

Proceed

If the user or broker has forgotten his password, he can enter his mail ID, and his password will be mailed to him.

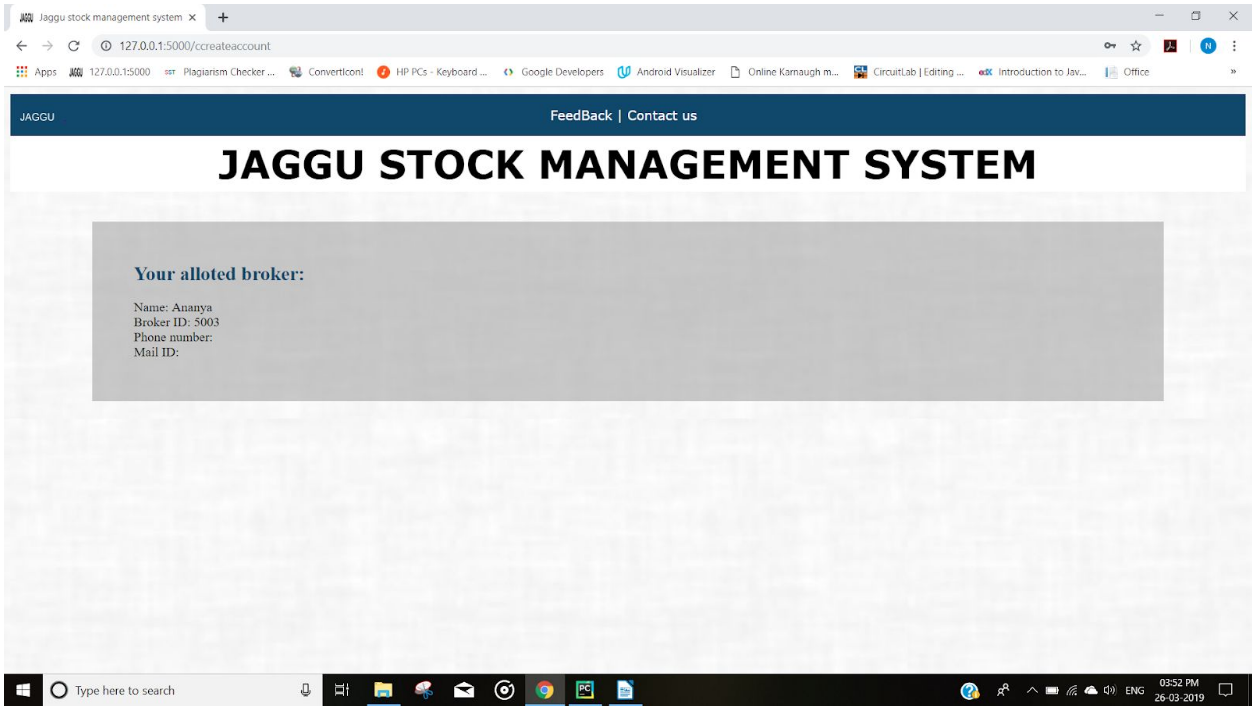


JAGGU STOCK MANAGEMENT SYSTEM

CUSTOMER ACCOUNT CREATION

Name	
Email ID	
Phone number	
Date-of-Birth	
Present Address	
Permanent Address	
User ID	
Password	
Re-Type Password	
Account holder's name	
Account number	
IFCS	
Branch Address	

The customer can enter his detailed create an account and then click on allot broker to get a broker allotted to him randomly.

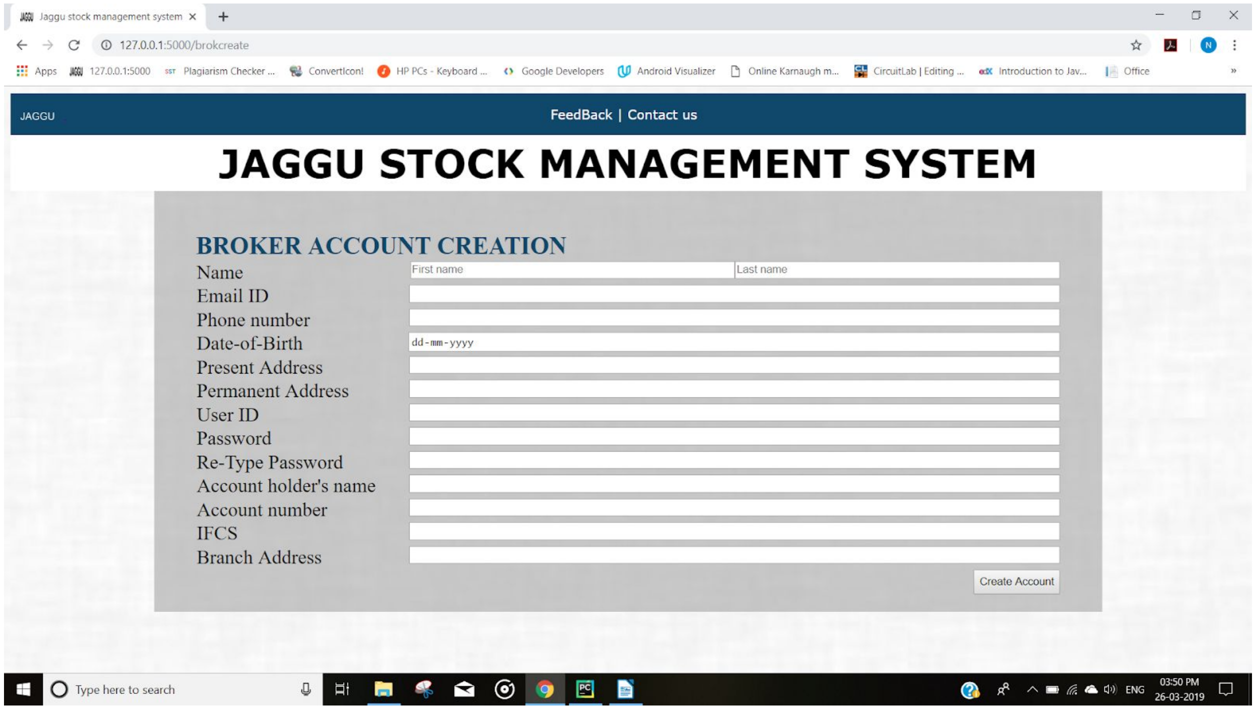


JAGGU STOCK MANAGEMENT SYSTEM

Your alloted broker:

Name: Ananya
Broker ID: 5003
Phone number:
Mail ID:

After the customer clicks on allot broker button, he will be redirected to the above page where his broker details will be displayed.

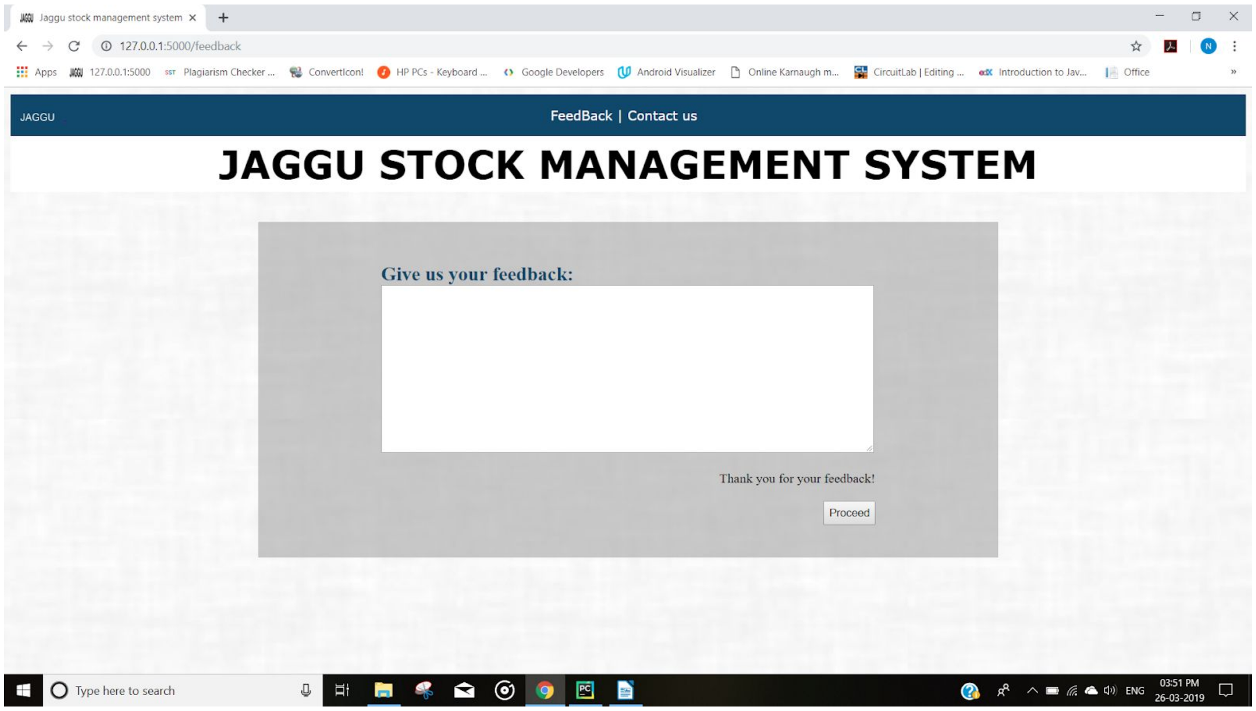


JAGGU STOCK MANAGEMENT SYSTEM

BROKER ACCOUNT CREATION

Name	First name / Last name
Email ID	
Phone number	
Date-of-Birth	dd-mm-yyyy
Present Address	
Permanent Address	
User ID	
Password	
Re-Type Password	
Account holder's name	
Account number	
IFCS	
Branch Address	

The broker can create an account by entering his details.

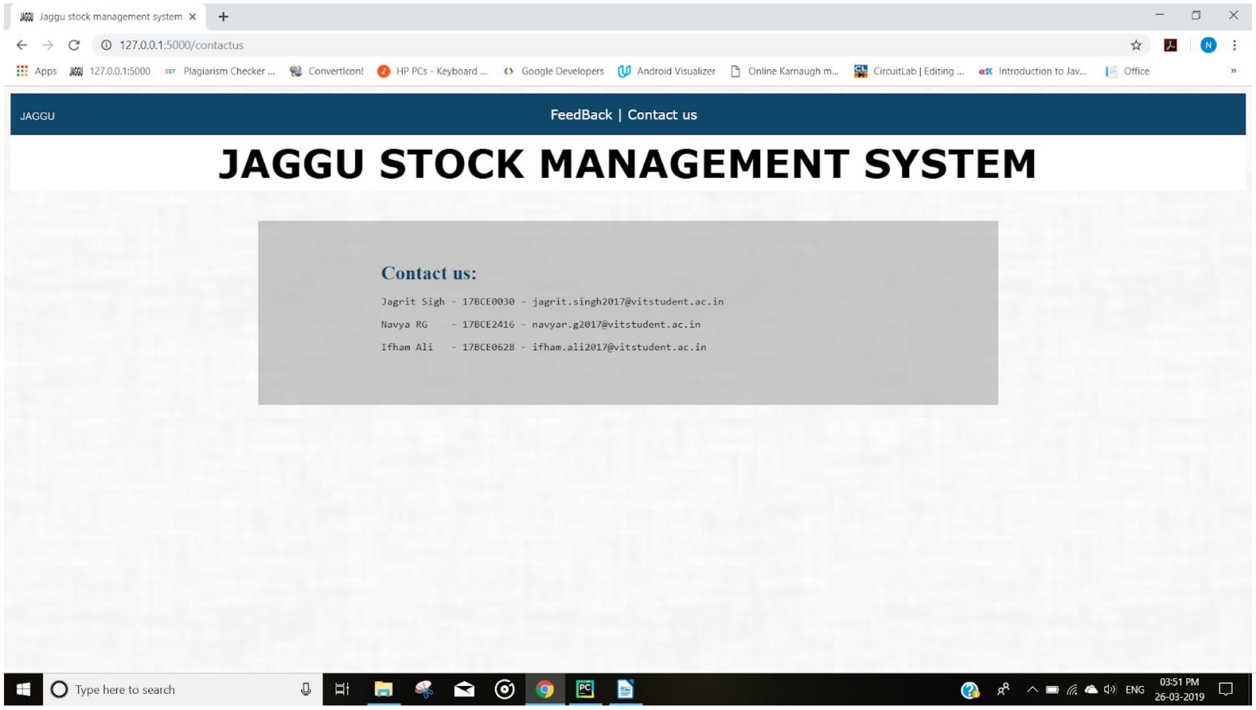


JAGGU STOCK MANAGEMENT SYSTEM

Give us your feedback:

Thank you for your feedback!

The user of this web application can give their feedback in the above page.


JAGGU

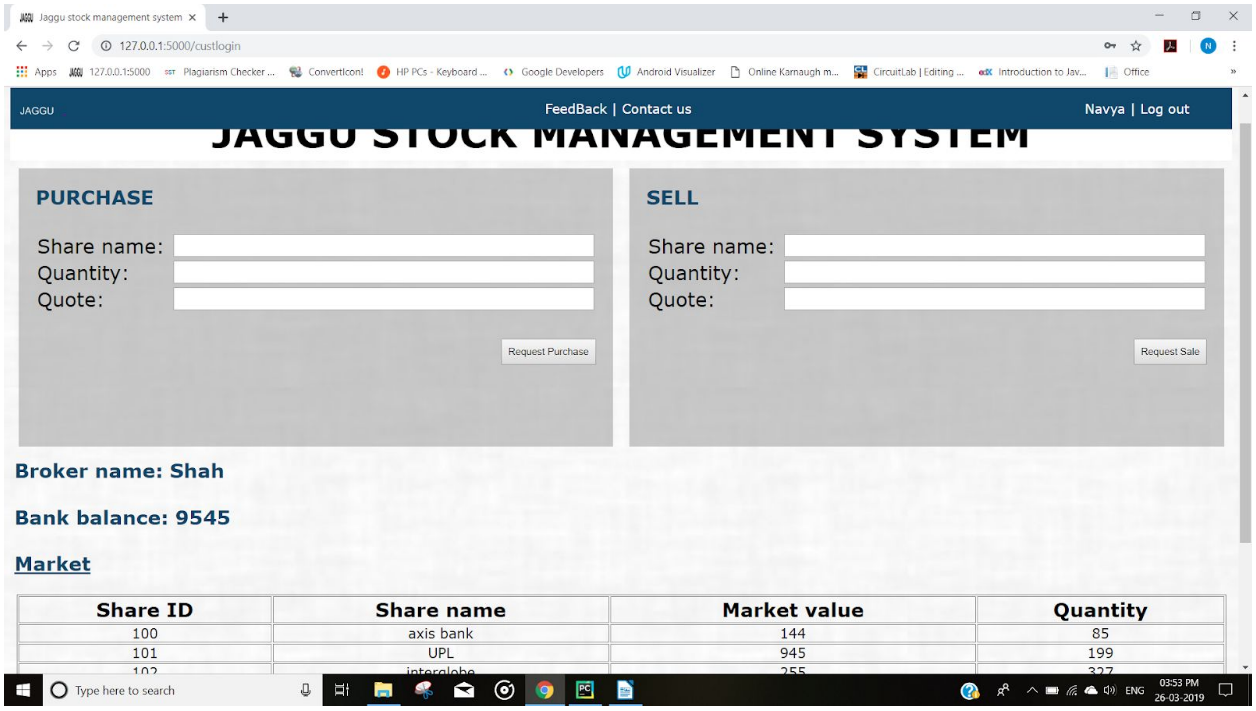
FeedBack | Contact us

JAGGU STOCK MANAGEMENT SYSTEM

Contact us:

```
Jagrit Sigh - 17BCE0030 - jagrit.singh2017@vitstudent.ac.in
Navya RG    - 17BCE2416 - navyar.g2017@vitstudent.ac.in
Ifham Ali   - 17BCE0628 - ifham.ali2017@vitstudent.ac.in
```

Through the 'contact us' link, the users are redirected to the above page which contains the details of the creators of the stock management system.


JAGGU

FeedBack | Contact us

Navya | Log out

JAGGU STOCK MANAGEMENT SYSTEM

PURCHASE

Share name:
Quantity:
Quote:

Request Purchase

SELL

Share name:
Quantity:
Quote:

Request Sale

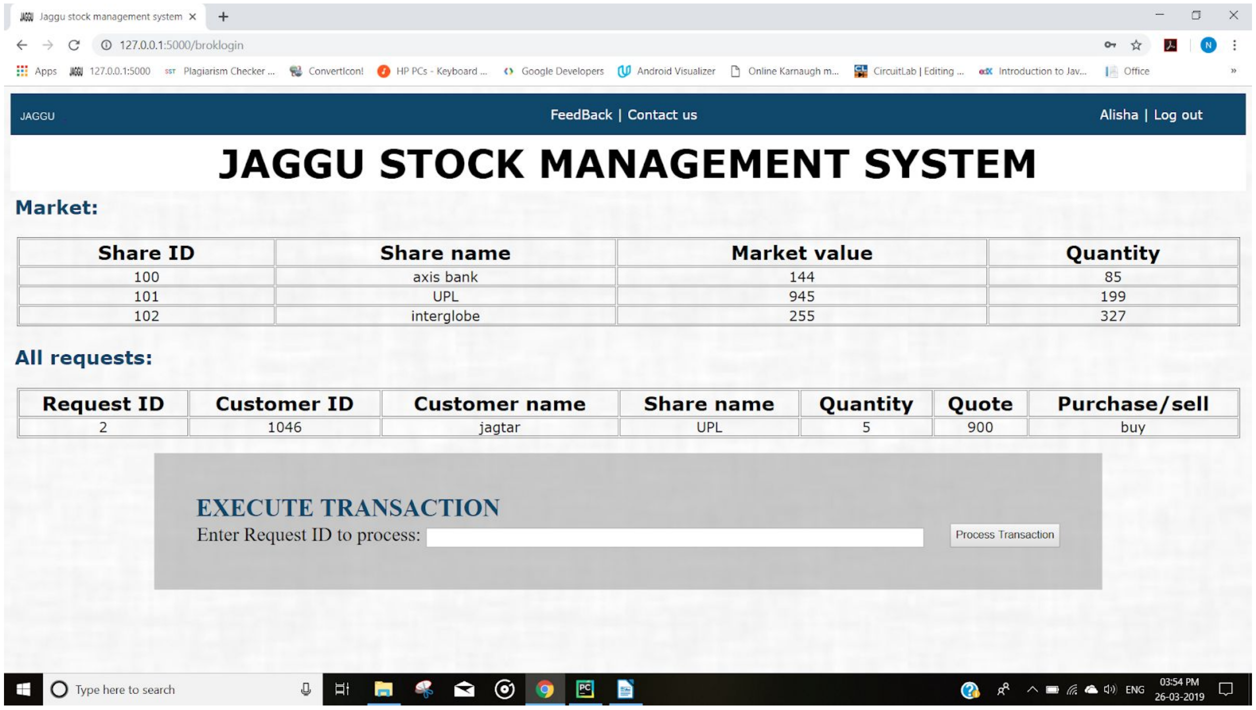
Broker name: Shah

Bank balance: 9545

Market

Share ID	Share name	Market value	Quantity
100	axis bank	144	85
101	UPL	945	199
102	interglobe	255	327

The customer homepage where customers can request purchase or sale of shares by looking at the current market values.



JAGGU STOCK MANAGEMENT SYSTEM

Market:

Share ID	Share name	Market value	Quantity
100	axis bank	144	85
101	UPL	945	199
102	interglobe	255	327

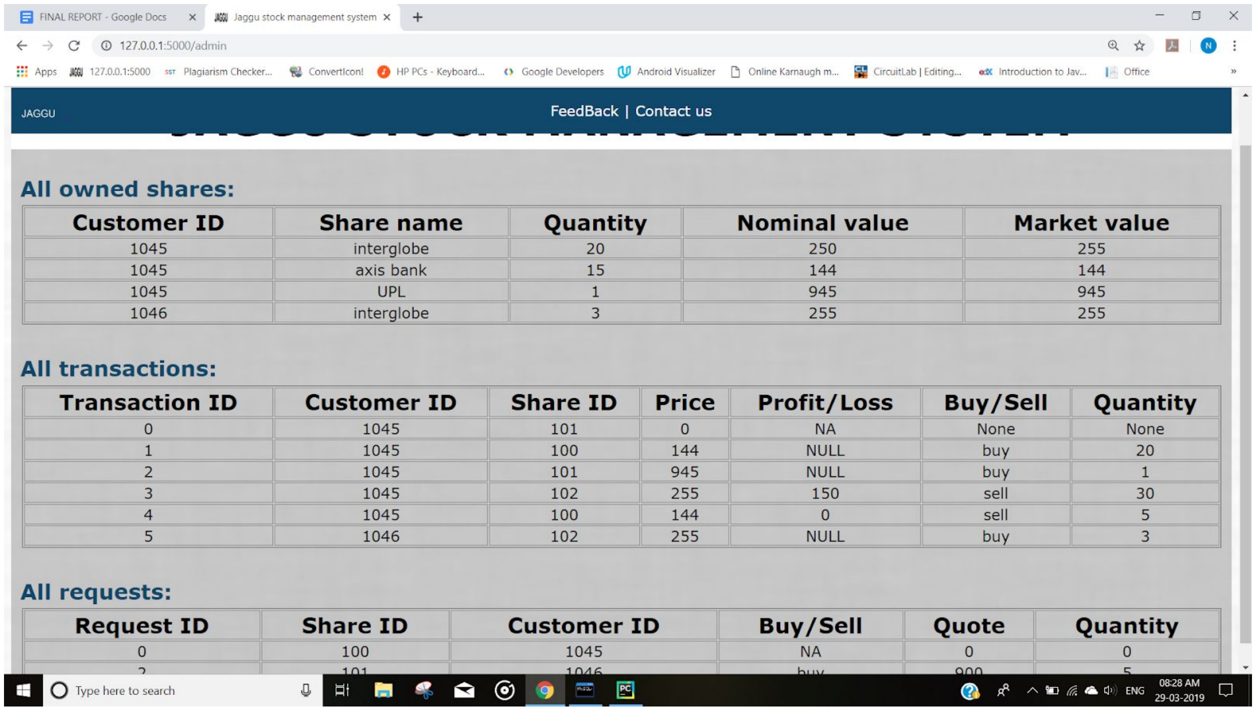
All requests:

Request ID	Customer ID	Customer name	Share name	Quantity	Quote	Purchase/sell
2	1046	jagtar	UPL	5	900	buy

EXECUTE TRANSACTION

Enter Request ID to process:

The broker homepage where broker can execute transactions that are requested to them by looking at the market values.



All owned shares:

Customer ID	Share name	Quantity	Nominal value	Market value
1045	interglobe	20	250	255
1045	axis bank	15	144	144
1045	UPL	1	945	945
1046	interglobe	3	255	255

All transactions:

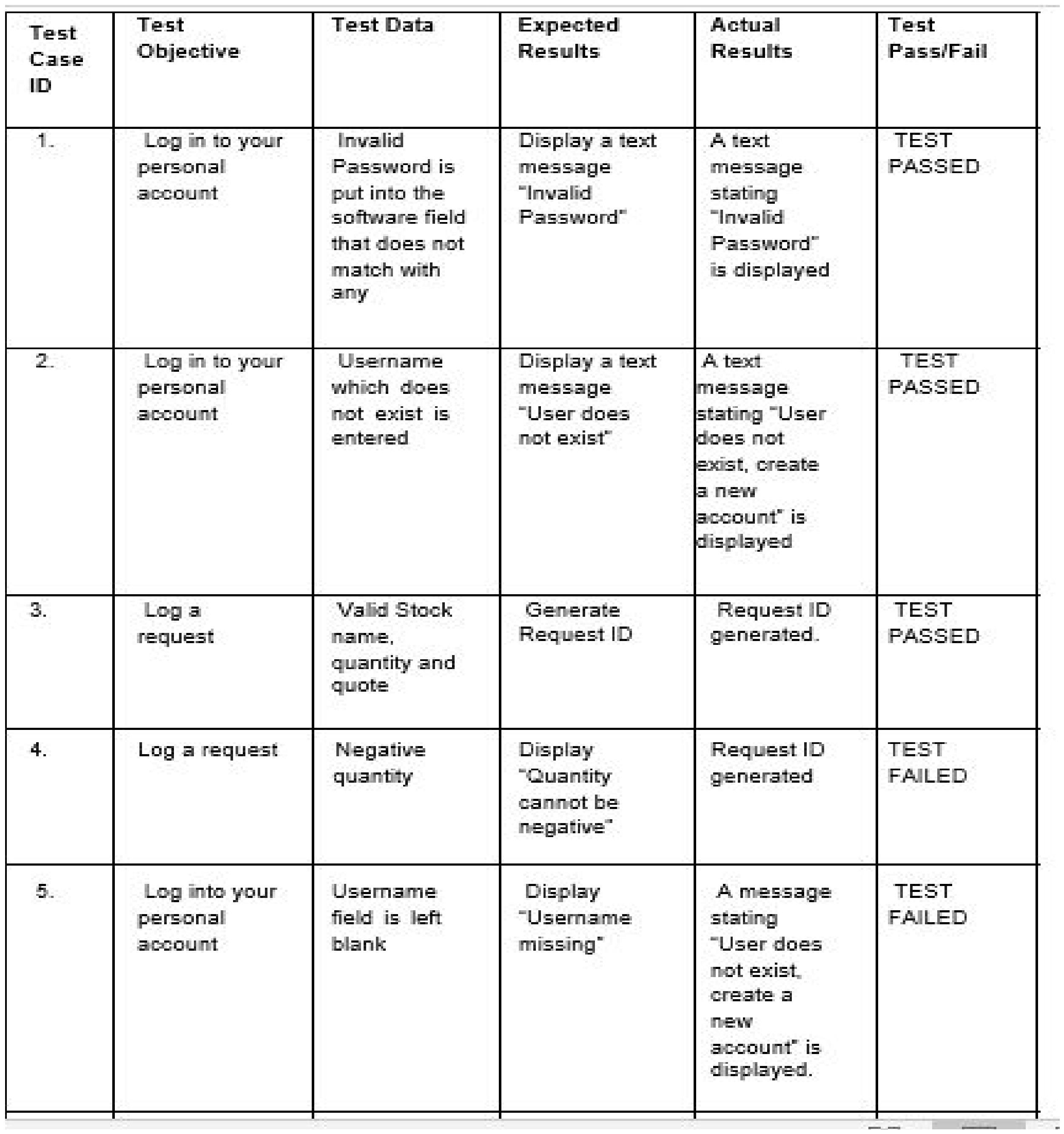
Transaction ID	Customer ID	Share ID	Price	Profit/Loss	Buy/Sell	Quantity
0	1045	101	0	NA	None	None
1	1045	100	144	NULL	buy	20
2	1045	101	945	NULL	buy	1
3	1045	102	255	150	sell	30
4	1045	100	144	0	sell	5
5	1046	102	255	NULL	buy	3

All requests:

Request ID	Share ID	Customer ID	Buy/Sell	Quote	Quantity
0	100	1045	NA	0	0
2	101	1046	buy	900	5

The webpage that the admin can view for obtaining the status of the system.

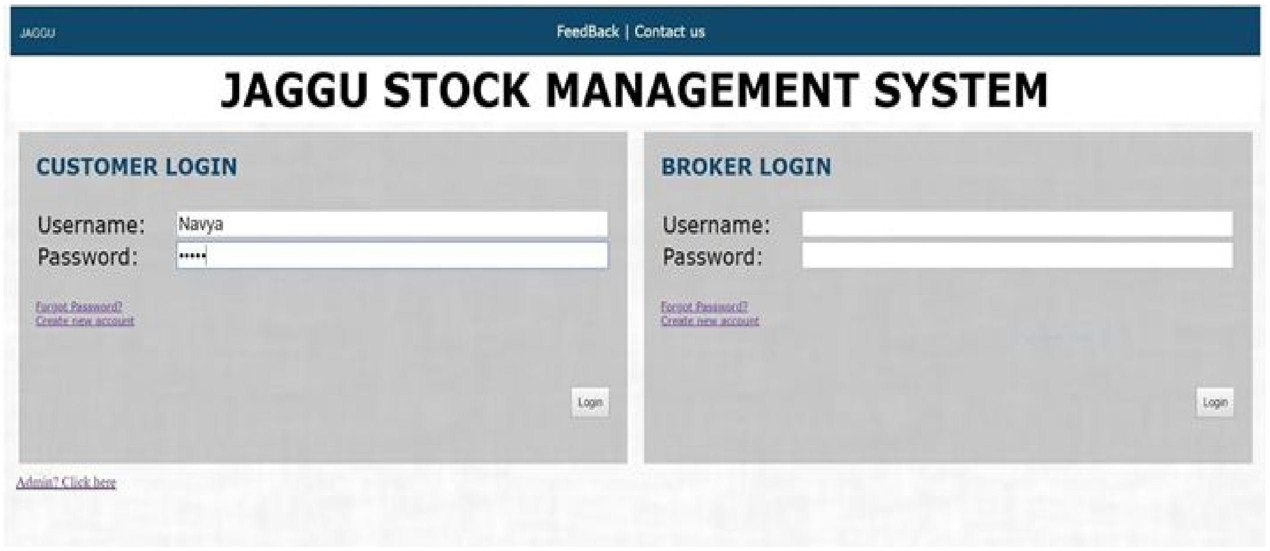
Testing



Test Case ID	Test Objective	Test Data	Expected Results	Actual Results	Test Pass/Fail
1.	Log in to your personal account	Invalid Password is put into the software field that does not match with any	Display a text message "Invalid Password"	A text message stating "Invalid Password" is displayed	TEST PASSED
2.	Log in to your personal account	Username which does not exist is entered	Display a text message "User does not exist"	A text message stating "User does not exist, create a new account" is displayed	TEST PASSED
3.	Log a request	Valid Stock name, quantity and quote	Generate Request ID	Request ID generated.	TEST PASSED
4.	Log a request	Negative quantity	Display "Quantity cannot be negative"	Request ID generated	TEST FAILED
5.	Log into your personal account	Username field is left blank	Display "Username missing"	A message stating "User does not exist, create a new account" is displayed.	TEST FAILED

		is put in as the request	"Share does not exist"	error is shown.	
7.	Portfolio Viewing	Valid Customer ID	Net value calculated along with the growth.	Calculated Net value displayed along with the growth.	TEST PASSED
8.	Portfolio Viewing	Invalid Customer ID	An apt text message to be displayed	No text message displayed	TEST FAILED

1. **Logging in to your personal account with invalid password**


JAGGU
FeedBack | Contact us

JAGGU STOCK MANAGEMENT SYSTEM

CUSTOMER LOGIN

Username: Navya
Password: •••••

Forgot Password?
Create new account

Login

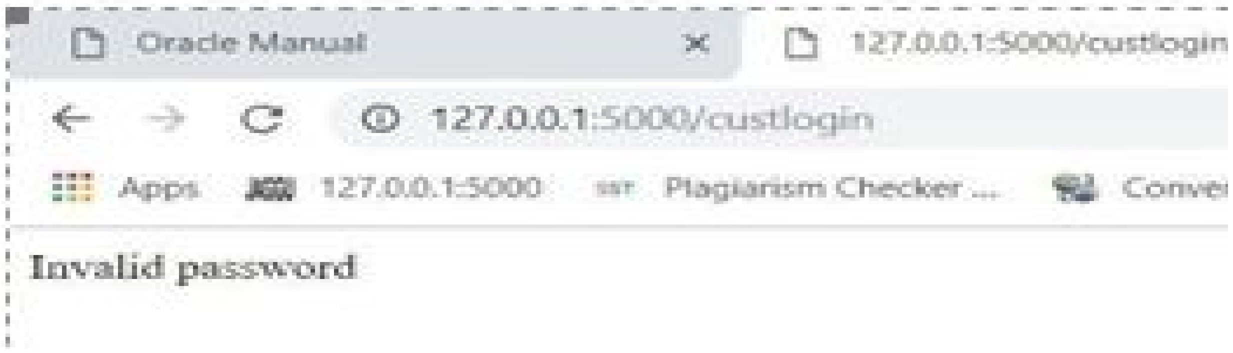
BROKER LOGIN

Username:
Password:

Forgot Password?
Create new account

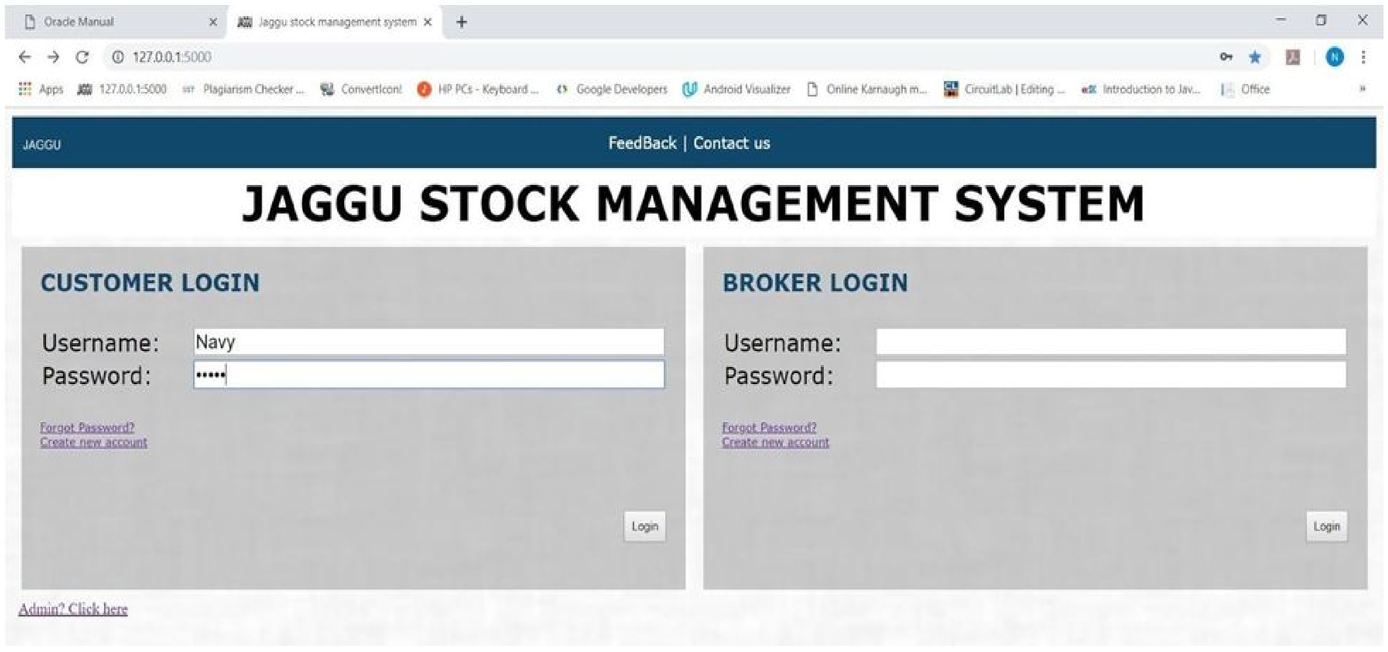
Login

Admin? Click here



Invalid password

2. **Logging in to your personal account with invalid username**



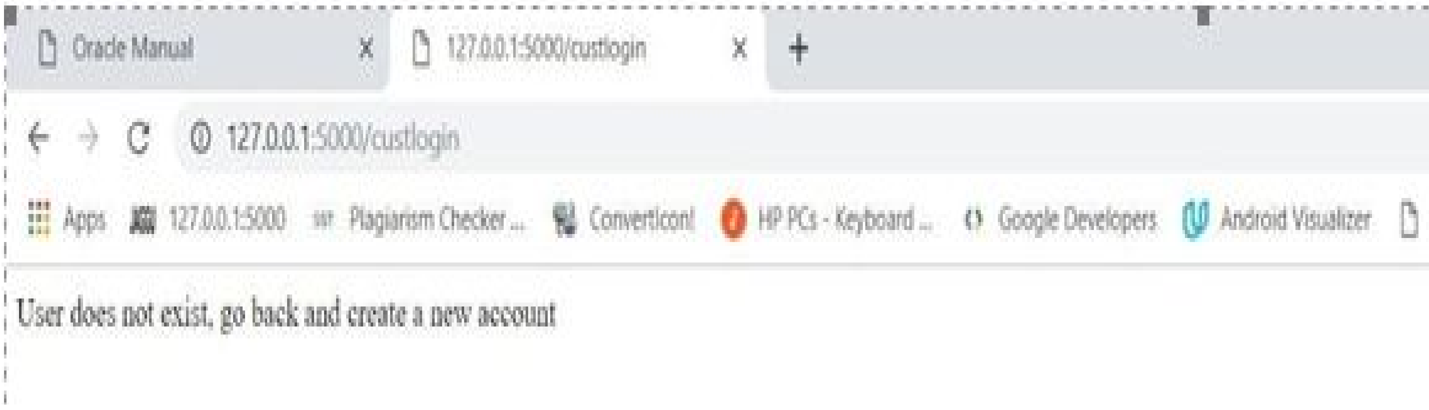
JAGGU STOCK MANAGEMENT SYSTEM

CUSTOMER LOGIN

Username: Navy
Password: •••••

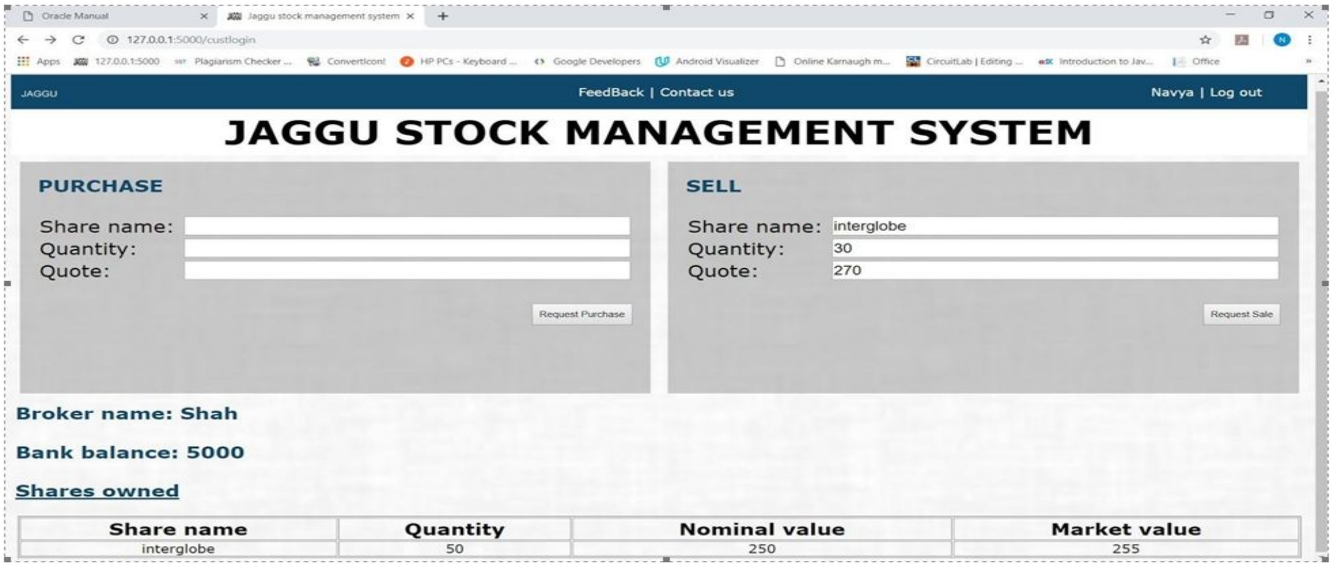
BROKER LOGIN

Username:
Password:



User does not exist, go back and create a new account

3. Log a request



JAGGU STOCK MANAGEMENT SYSTEM

PURCHASE

Share name:
Quantity:
Quote:

SELL

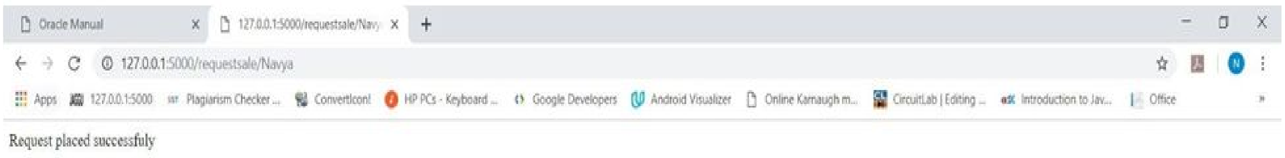
Share name: interglobe
Quantity: 30
Quote: 270

Broker name: Shah

Bank balance: 5000

Shares owned

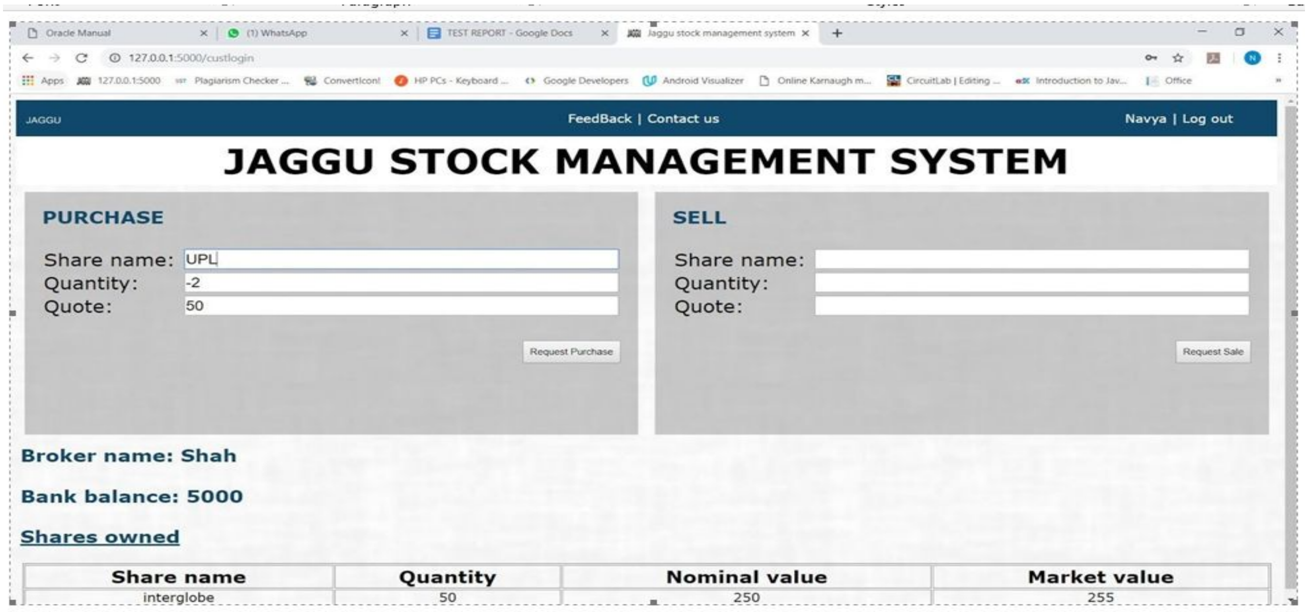
Share name	Quantity	Nominal value	Market value
interglobe	50	250	255



127.0.0.1:5000/requestsale/Navya

Request placed successfuly

4. Log a request with negative quantity



JAGGU STOCK MANAGEMENT SYSTEM

PURCHASE

Share name: UPL
Quantity: -2
Quote: 50

SELL

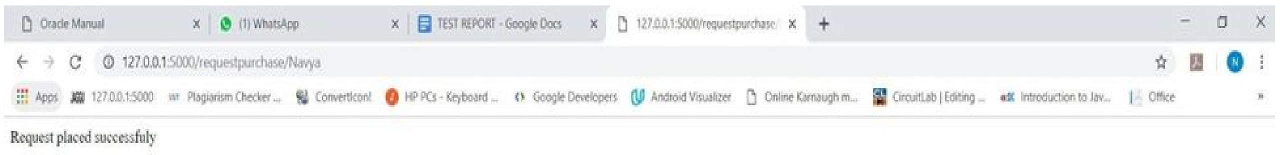
Share name:
Quantity:
Quote:

Broker name: Shah

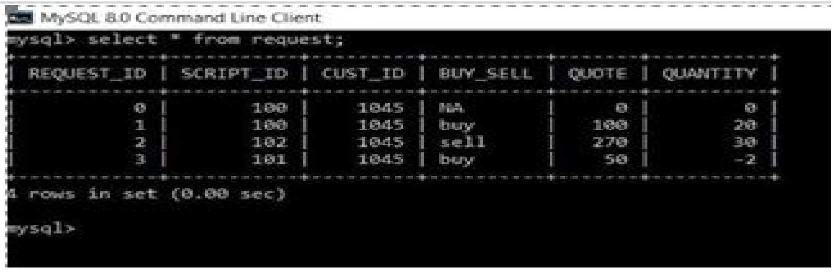
Bank balance: 5000

Shares owned

Share name	Quantity	Nominal value	Market value
interglobe	50	250	255


Oracle Manual
(1) WhatsApp
TEST REPORT - Google Docs
127.0.0.1:5000/requestpurchase/
127.0.0.1:5000/requestpurchase/Navya

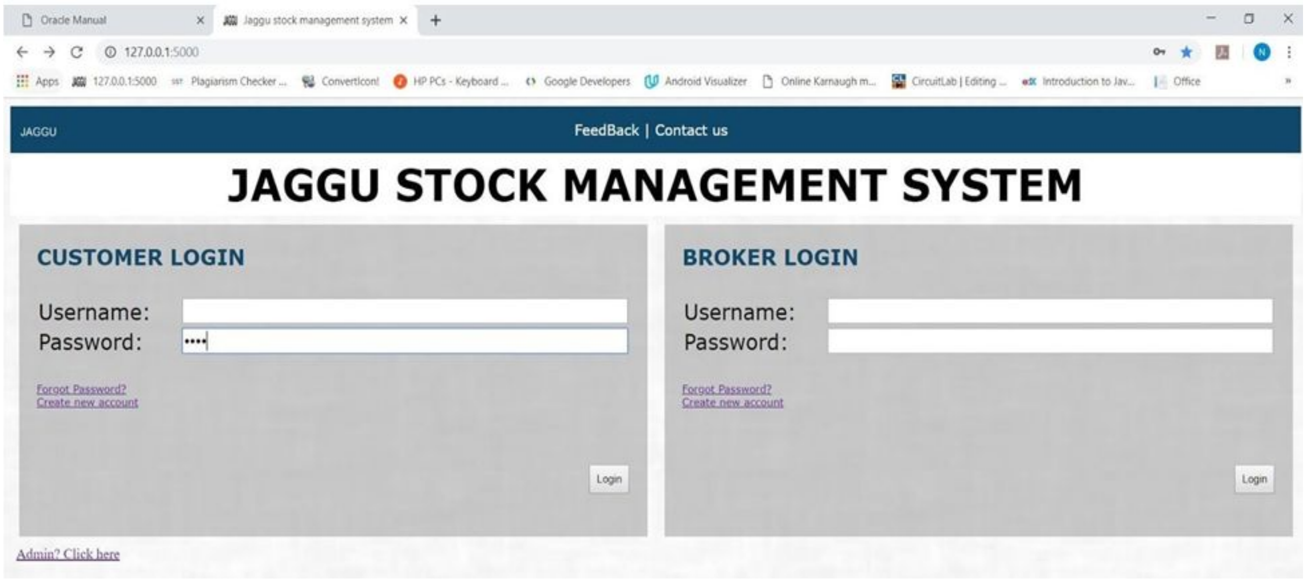
Request placed successfuly



```
mysql> select * from request;
+------------+-----------+---------+----------+-------+----------+
| REQUEST_ID | SCRIPT_ID | CUST_ID | BUY_SELL | QUOTE | QUANTITY |
+------------+-----------+---------+----------+-------+----------+
|          0 |       100 |    1045 | NA       |     0 |        0 |
|          1 |       100 |    1045 | buy      |   100 |       20 |
|          2 |       102 |    1045 | sell     |   270 |       30 |
|          3 |       101 |    1045 | buy      |    50 |       -2 |
+------------+-----------+---------+----------+-------+----------+
4 rows in set (0.00 sec)

mysql>
```

5.Login in to your personal account with missing username field



JAGGU

FeedBack | Contact us

JAGGU STOCK MANAGEMENT SYSTEM

CUSTOMER LOGIN	
Username:	
Password:	••••

Forgot Password?
Create new account

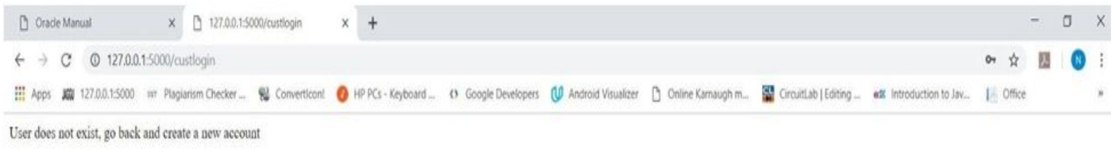
Login

BROKER LOGIN	
Username:	
Password:	

Forgot Password?
Create new account

Login

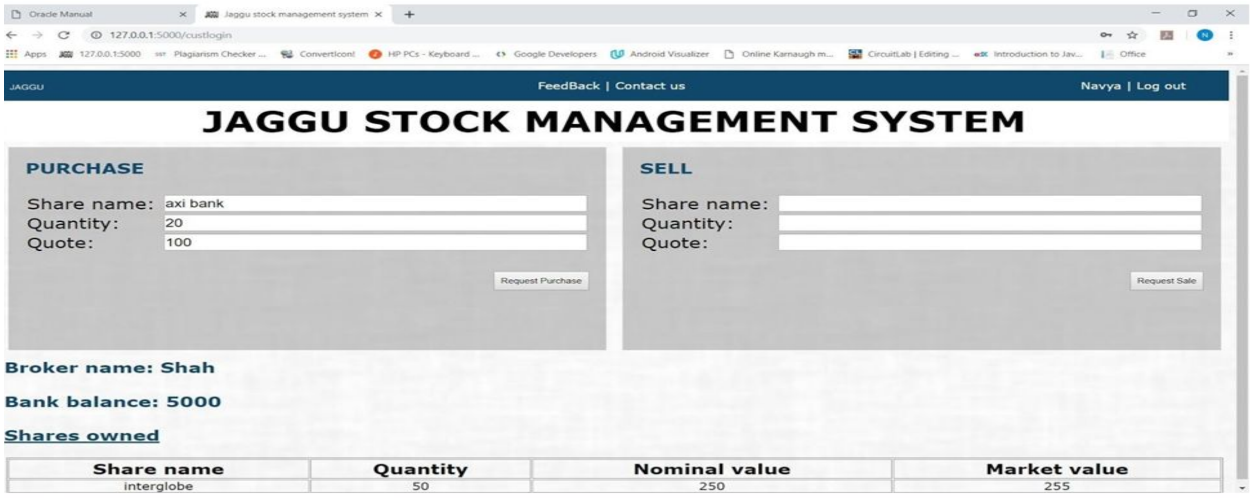
Admin? Click here



127.0.0.1:5000/custlogin

User does not exist, go back and create a new account

6. Login in a request with wrong share name



JAGGU

FeedBack | Contact us

Navya | Log out

JAGGU STOCK MANAGEMENT SYSTEM

PURCHASE	
Share name:	axi bank
Quantity:	20
Quote:	100

Request Purchase

SELL	
Share name:	
Quantity:	
Quote:	

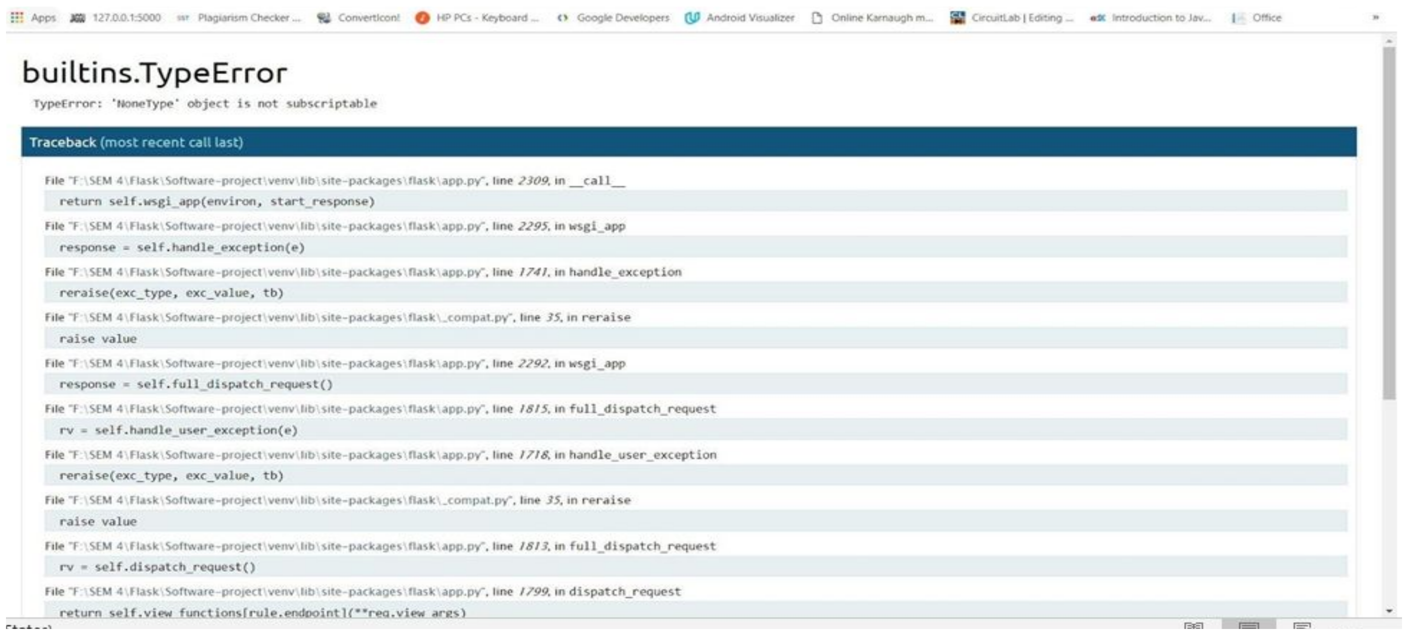
Request Sale

Broker name: Shah

Bank balance: 5000

Shares owned

Share name	Quantity	Nominal value	Market value
interglobe	50	250	255



builtins.TypeError

```
TypeError: 'NoneType' object is not subscriptable
```

Traceback (most recent call last)

```
File "F:\SEM 4\Flask\Software-project\venv\lib\site-packages\flask\app.py", line 2309, in __call__
  return self.wsgi_app(environ, start_response)
File "F:\SEM 4\Flask\Software-project\venv\lib\site-packages\flask\app.py", line 2295, in wsgi_app
  response = self.handle_exception(e)
File "F:\SEM 4\Flask\Software-project\venv\lib\site-packages\flask\app.py", line 1741, in handle_exception
  reraise(exc_type, exc_value, tb)
File "F:\SEM 4\Flask\Software-project\venv\lib\site-packages\flask\_compat.py", line 35, in reraise
  raise value
File "F:\SEM 4\Flask\Software-project\venv\lib\site-packages\flask\app.py", line 2292, in wsgi_app
  response = self.full_dispatch_request()
File "F:\SEM 4\Flask\Software-project\venv\lib\site-packages\flask\app.py", line 1815, in full_dispatch_request
  rv = self.handle_user_exception(e)
File "F:\SEM 4\Flask\Software-project\venv\lib\site-packages\flask\app.py", line 1718, in handle_user_exception
  reraise(exc_type, exc_value, tb)
File "F:\SEM 4\Flask\Software-project\venv\lib\site-packages\flask\_compat.py", line 35, in reraise
  raise value
File "F:\SEM 4\Flask\Software-project\venv\lib\site-packages\flask\app.py", line 1813, in full_dispatch_request
  rv = self.dispatch_request()
File "F:\SEM 4\Flask\Software-project\venv\lib\site-packages\flask\app.py", line 1799, in dispatch_request
  return self.view_functions[rule.endpoint](**req.view_args)
```