Software Requirements Specification

for

Stock Market

Version 1.0 approved

Prepared by

P. sai kishore reddy **PES2201800096 (56)**

Manne Vasanth **PES2201800425 (12)**

Lekha suresh **PES2201800469 (34)**

Supervised By

Prof. Sudeepa Roy Deepa

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The product is a web application based stockbroker, release **1.0**. The SRS describes only the client end trading subsytem of the stock trading system. Our client end communicates with the central trading system managed by a stock exchange, such as NSE or BSE. It provides the users a convenient way to perform stock related activities, such as to check stock prices, and buy and sell shares.

1. Most of the Requirements related to software system will be realized in the system design itself.

2. A clear and detailed development of the system architecture.

3. We will leave it to the system to adapt to environment and improve its performance eventually.

4. Finally divide the system structure to modules and functions for clear and efficient coding.

1.2 Intended Audience

This document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. . The audience is suggested to read papers on software engineering, design for a greater depth into the subject.

This document will be of interest to the following audience:

- Developers: The application developers need to refer to this document to have clarity on the product they're building.
- Project Managers: They refer to this document to analyse its various requirements and features, and decide on the course of action to build the application.
- Users: Those clients who wish to have a deeper understanding of the working of the platform could refer to this document.
- Testers: The testers will find this script very helpful to build test cases to verify the system's integrity once it has been developed. The requirement traceability matrix provided will greatly ease the forward and backward tracing between requirements and test cases.
- Documentation Writers: Other documentation writers who wish to develop similar applications could find the description of the basic framework useful.

1.3 Product Scope

In this project we mainly intend develop a Subpart of Stock Trading System. As this subpart is Client End of Trading. The following tips will show its scope.

1. It must give users convenient and effective ways to deal with stocks. For instance client will be able to know the Price of one stock of a particular company at that point of time, and he can buy the amount of stocks he needed. The client can even sell the Stocks at any point of time.

2. Client will be displayed with all the companies that are willing to share their stocks.

3. Friendly interfaces are also necessary in this project. Client should be able to add money to his app wallet whenever he needs.

4. The objective is to allow users to trade at their convenience and open the stock market to the common public, helping it thrive. Trading through a broker like us removes the need to have specialized knowledge regarding the working of equity markets. The user is likely to have a higher Return On Investment and a more seamless experience to help fulfill their financial goals.

1.4 References

《Stock Trading System》 《Software Engineering》

Author: Roger S.Pressman Press: McGraw Hill

We also refered to a SRS document that is prepard by Jin Li on the Project Stock trading System.

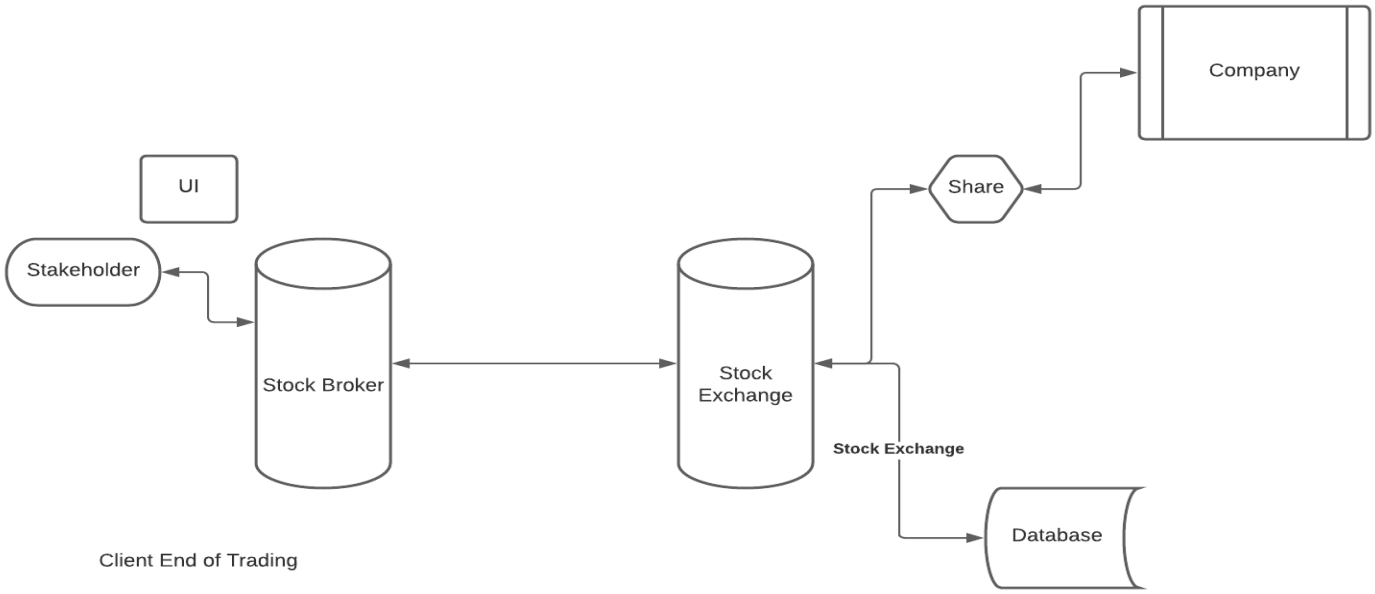
Securities and Exchange Board of India Act, 1992:
https://www.sebi.gov.in/sebi_data/attachdocs/1456380272563.pdf
Zero Commission Trading:
Bloomberg Businessweek:
https://www.bloomberg.com/news/articles/2019-10-10/brokers-profit-from-you-even-if-they-don-t-charge-for-trading
The Ascent by The Motley Fool:
https://www.fool.com/the-ascent/buying-stocks/articles/zero-commission-trading-what-you-need-to-know/

2. Overall Description

2.1 Product Perspective

We all know that stock market has been an important part for economy. Every day, millions of trades are competed in the stock market. It is very difficult to deal with all these trades manually. So it is very useful to develop a software for stock market system. This is what most of the developers plan to do: stock trading system. That helps the traders to invest their money in the stock market.

This project is a system that is used in many fields, including stock trade, trade control, and market management. A series of interfaces will be developed to suit different requirements of the clients. A predefined database will be set up to save the information internally. And all the operations are based on this same database. Besides the database and interfaces, a centralized trading system will be developed to deal with the trading and the changes on the database.



2.2 Product Functions

The user must be able to:

- create an account, and get it verified
- display currently owned stock quotes, previously owned stock quotes
- edit their personal information
- move funds from their bank account to demat account
- Buy and sell shares, execute various kinds of orders.

2.3 User Classes and Characteristics

No matter how advanced a computer interface is. Users' characteristics will always be the most important element rather than the designers. Our potential customers are those who are professional stockbrokers or public users. They may be Regular and Non-Regular users. They also have different education levels and professions. Our task is to design a general and easy-to-use system for the customer

➢ **Regular Users:**
Classification:

❖ computer knowledge – moderate/high
❖ stock trading knowledge - high
❖ frequency of use – high

Interaction with the system: They do transactions either by clicking buttons and mouse or by pressing hot keys. The latter one is a better choice for them.

➢ **Non-Regular users:**
Classification:

❖ computer knowledge - varies, low-high
❖ stock trading knowledge - varies, low-high
❖ frequency of use - low

Interaction with the system: Most of them use mouse rather than the hot keys. Only a few of them who know more about computers use hot keys. A suitable font size and color, large-enough buttons and helpful tool tips are required to meet the general requirements. Besides, error or warning messages must be clear and provide specific guidance.

2.4 Operating Environment

Any device which can support the latest versions of the most popular browsers can run the software. It is operating system agnostic, though apps may be made available for mobiles in the future. The working of the software is dependant on whether the API of the stock exchange is functioning as expected. If changes have been made to the way the API serves data, the software must adapt accordingly.

2.5 Design and Implementation Constraints

Time Constraint: The regulatory policy is that stock exchanges generally have their own operable hours, hence trading is restricted to only that time **.**

Load Constraint: Due to load on the Exchange servers, there may be network latencies. In the practical situation, the stock broker application must be able to handle such situations. The software must also comply with the API.
Programming is done in Python, while MySQL will be used to store the users' details. The working of the backend is completely abstracted from us.

Communication Constraint: The client end communicates with the backend servers via HTTP GET/POST requests.

The system has to be secure since these transactions deal with very large amounts of money, but this aspect will be handled by another team. For this phase, the software will be considered inherently secure.
Regular updates and feature enhancements will be performed.

Actual membership fees to become a licensed stock broker for a security exchange in India is generally upwards of 20 lakhs INR, so we may not be able to demonstrate an actual trade. The real time quotes may have to be scraped off the website.

2.6 Assumptions and Dependencies

There are many factors that have great influence on the project. The design must implement all of the explicit requirements contained in the analysis model, and it must accommodate all of the implicit requirements desired by the customer. So it is important to make sure the primary conditions of the customers and develop environments. That is, assumptions and dependences.

Assumption:

- In the analysis model, basic information of the requirements is clear. No big mistakes exist. If a big mistake is ignored in the beginning of the project, it will lead to a disaster that the whole work has to be checked.
- All the developers are trained and familiar with the project. And the number of people in the project team is adequate to do the job. When the situation of lacking developers occurs, the delivery time for the software may be delayed.
- The developer team has a good estimation for the technical problems and software size. When technology does not meet exception, there should be some alternative schemes.

- The scope and requirements of the project is stable. Because the model is similar with FLOW model. The final work is accomplished in the last stage of the develop process. Any change after the requirements analysis stage will force the developers to modify the architecture of the system, which takes a lot of time and human resources.

Dependence:

- Developers have had a clear view for the system and a detailed schedule has been made. Requirements analysis is treated carefully so that developers have the specification of software's operational characteristics.

- The technology developers prefer has been used in some similar systems and it proves to work in gear. And a lot of jobs have been done. They can offer us great experience and ideas.

- Well, this system is just a basic stock trading system. Developers can neglect the security. The number of the users is not very much, which reduce a lot of work on the communication.

3. External Interface Requirements

3.1 User Interfaces

The web application is accessed via web browser on any device. An app may be developed later but that is assigned low priority for now. There must be a navbar at the top with quick links. On the home screen the user will have a login option if he is a valid user. If a person is new and wants to login he can use sing up option to become a valid user. In the client home page it will display the amount he has, companies and their stocks and their stock price. Error messages that may occur during login/registration or other operations have to be concise and in red to draw attention towards it.

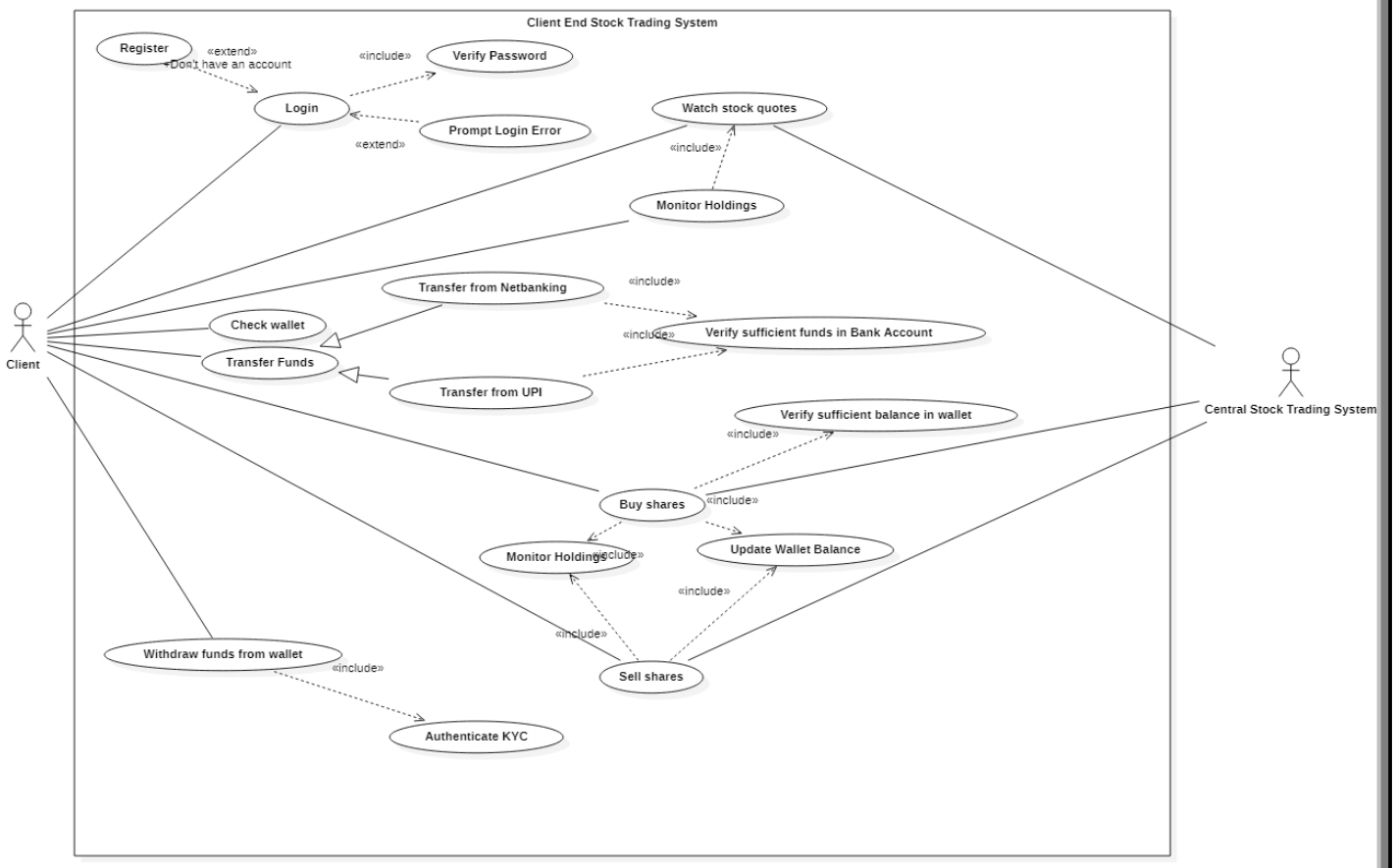
3.2 Software Interfaces

The data of the users and the stocks and the companies etc .. data is stored in sql database.
The information is stored in the database. This web application can run on any operating system that can support the latest version of the popular browsers. MySQL is the database used to store the users' information and Python 3.x is the object oriented programming language used. Messages come into the system in the form of CSV, XML or JSON formats, these include the prices of each share of the stock that has been queried, or the acknowledgements to the buy or sell transaction that has been executed.

3.3 Communications Interfaces

HTTPS must be the communication standard used to avoid MITM attacks since this involves potentially large amounts of cash, thus integrity and security of the system must be maintained. We can use some cryptographic algorithms.

4. Analysis Models



5. System Features

5.1 System Feature 1 (Login)

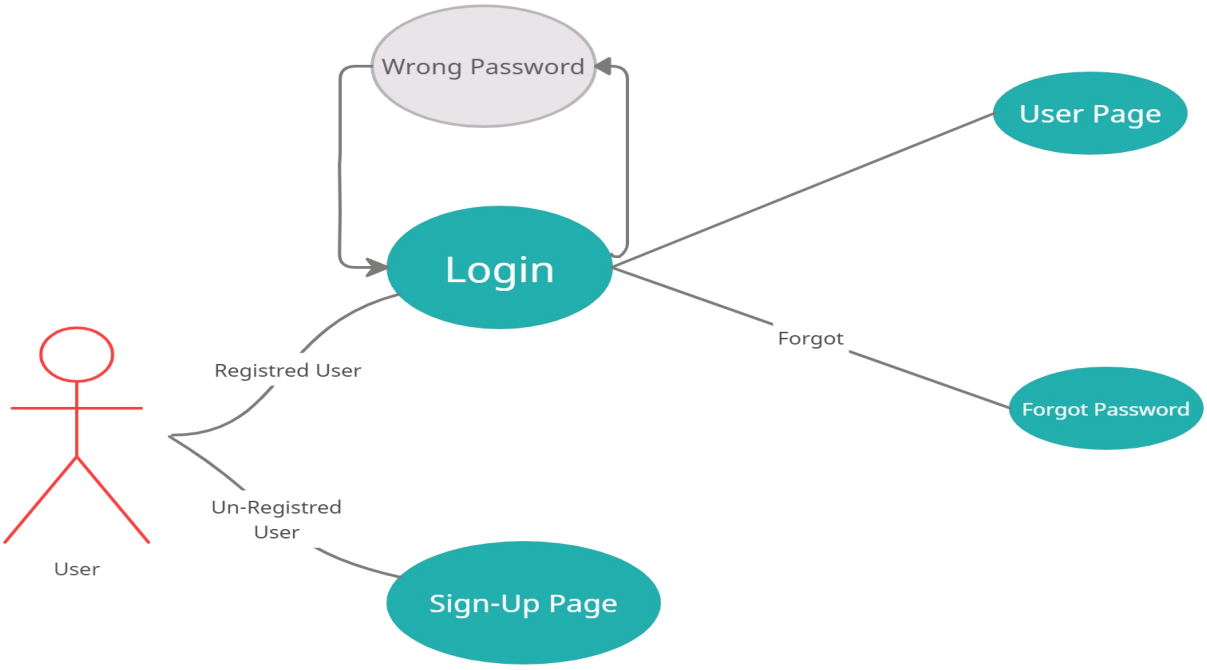
5.1.1 Description and Priority

Allows users to login into their registered account by providing their email id or user id and their respective password. The user can access their homepage after logging in to their account.

5.1.2 Stimulus/Response Sequences

✓ **User Login:**

Purpose	User can login to their existing account
User	An user with an existing account
Input data	Profile Username and password
Output data	User home page
Invariant	Profile table data and user information
Pre-conditions	User is not logged into a profile, input profile exists in data base, user password matches profile
Post-conditions	User's system has supplied with the appropriate data of the user
Flow	The login page is updated to match new user data. Invalid password, invalid username, or mismatched username and password redirected to error message.



User enters correct username and correct password->User can access their homepage

User enters invalid username-> User will be prompted to create a new account as it doesn't exist in the database

User enters incorrect password for a valid username->User will be prompted to enter the correct password

5.1.3 Functional Requirements

REQ 1: System must check whether the entered combination is present in the database, if invalid username is entered then the user must be prompted to create a new account. If the password is invalid then the user must enter the correct password.

5.2 System Feature 2 (Registration)

5.2.1 Description and Priority

Allows users to create an account by providing their email id as the username and their password. After filling up the registration form the users would need to verify their account. Then the user has to provide their phone number and verify it. These steps would lead to the creation of the account.

5.2.2 Stimulus/Response Sequences

Enter valid Email id -> Enter valid password -> Retype the password (Verify if both the passwords are matching) -> Account is successfully created.

User entered invalid Email id->returns an error ->enter valid email id

Enter valid Email->User entered invalid password-> system returns invalid password->user needs to enter valid password

Enter valid Email id -> Enter valid password -> Retype the password->Retyped password is not matching the original password->Retype the confirmation password

5.2.3 Functional Requirements

REQ 2: Non Registered users must enter the mail id. The mail id entered by the user must be unique the database and if the mail is already then he must enter an alternate mail id.

REQ 3: The password entered by the user must be validated and also ensure that the confirmed password is the same as the password entered by the user

5.3 System Feature 3 (Watch Stock Prices)

5.3.1 Description and Priority

Allows users to observe the current share price for a particular stock and appropriate decisions.

5.3.2 Stimulus/Response Sequences

User searches for a stock-> System returns the current share price for the particular stock.

User searches for a stock-> If the stock doesn't exist then the system must notify the user and the user must enter a valid stock name.

5.3.3 Functional Requirements

REQ 4: If the stock searched by the user doesn't exist then the system must notify the user about the absence of the stock and prompt the user to enter the valid stock name.

5.4 System Feature 4 (Transfer Funds)

5.4.1 Description and Priority

Allows users to transfer funds from their bank account to their wallet, the funds can be transferred

5.4.2 Stimulus/Response Sequences

Enter the correct details of the account -> amount will be credited to wallet

If user entered wrong details -> invalid details

✓ **Add Funds:**

Purpose	To add amount to the account
User	All valid users
Input data	Amount and the account details
Output data	Add specified amount to the user account
Invariants	Specified Amount
Pre-conditions	The given account details for adding money should be valid
Post-conditions	The amount should be added to the user account
Flow	The specified amount should be added to the user account. If the account details of the bank/any valet details are not given correctly then it should throw a notification.

5.4.3 Functional Requirements

REQ 5: Correct details of the account then the amount is added to wallet.

5.5 System Feature 5 (Buy stocks)

5.5.1 Description and Priority

The user must specify the number of shares and the name of the stock that he is interested to buy.

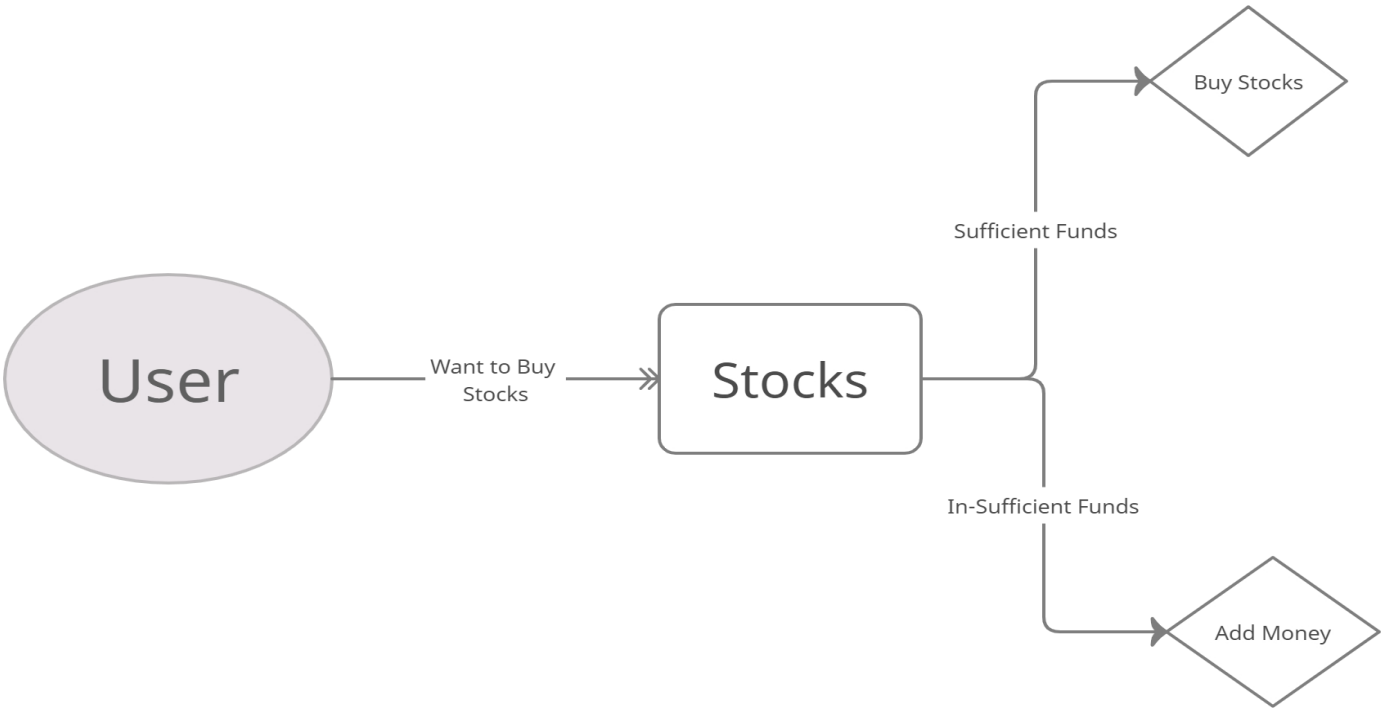
5.5.2 Stimulus/Response Sequences

✓ **Buy stocks:**

Purpose	To buy the company stocks
User	All valid users
Input data	company and Number of stocks
Output data	Deduction of amount and the stocks added to the user status
Invariants	Stocks and price
Pre-conditions	Valid user and the sufficient amount in the account
Post-conditions	Amount deducted and stocks are added to the user account
Flow	User logs in, add number of stocks form a specific company and buys the stocks. If the amount is in sufficient it notifies to add amount.

User should search/select the name of the stock and must specify the number of shares that the user wishes to purchase-> The user buys the shares

User purchases shares whose cost is greater than the funds in his wallet ->System cancels the transaction and notifies the user that his funds are insufficient.



5.5.3 Functional Requirements

REQ 6: System must verify that the cost of the shares don't exceed the funds in the wallet. If validated, the transaction is communicated to the backend via the API and the system awaits the confirmation from the backend that the transaction was successful. The user's holdings are updated to reflect the transaction.

5.6 System Feature 6 (sell stocks)

5.6.1 Description and Priority

The user must specify the number of shares and the name of the stock that he is interested to sell.

5.6.2 Stimulus/Response Sequences

✓ **Sell stocks:**

Purpose	To sell the company stocks
User	All valid users
Input data	company and Number of stocks
Output data	Addition of amount to the user and the stocks added to the company.
Invariants	Stocks and price
Pre-conditions	Valid user and the sufficient amount of stocks in the account
Post-conditions	Amount added to the account and stocks are added to the company
Flow	User logs in, add number of stocks form a specific company and sell the stocks.

User should select the stock and specify the number of shares he wishes to sell-> The user sells the shares

5.6.3 Functional Requirements

REQ 7: The system validates that user is trying to sell only what he owns. Once validated, the transaction is communicated to the backend via the API and the system awaits the confirmation from the backend that the transaction was successful. The user's holdings are updated to reflect the transaction.

5.7 System Feature 7 (Withdraw Funds)

5.7.1 Description and Priority

The user must specify the amount he wants to withdraw from the wallet.

5.7.2 Stimulus/Response Sequences

User specifies the amount to withdraw -> KYC has been done before -> The transfer is successful.

User specifies the amount to withdraw -> KYC hasn't been done before -> The system prompts the KYC

5.7.3 Functional Requirements

REQ 8: The system checks if the user has already completed his KYC verification. If so, the withdrawal is allowed. Else the user is directed to submit the necessary documents for KYC verification.

6. Other Nonfunctional Requirements

6.1 Performance Requirements

The performance requirements related to Client side of trading will be divided into these few parts:

1. The service life of the System that we are providing.
2. The Speed of efficiency at which the system runs.
3. Moreover the stability of the system is most important.

As the Client side of trading is one part of the Stocking trading application, the abilities of it depends on the center trading system. So to the Client side of trading, the data exchanging will be the most important. In the software design, orders which are accepted by both Client side of Trading and Central Trading System are used to meet the requirements. Apart from the orders, functions that are used to send or receive orders are also very important. In the software design, Object-oriented programming is chosen. All the users' interface will be objects to meet different functions. As the system is multithreading, data exchanges between different objects are under control strictly to ensure the stability of the system. Otherwise, functions of objects must work effectively and quickly to safeguard the run rate of the system.

6.2 Security Requirements

This Stock market Trading System is a small system, so the Instruction encryption may be abandoned. Therefore, the security requirements will be divided into only two parts:

1. The security of system.
2. The security of data.

To meet these requirements:

1. The security of system includes several aspects. For instance, one account can't login the system twice at the same time. To meet this requirement, the verify code is selected in our design.

2. The security of data includes three parts: Accounts, capital accounts and data in the database. To confirm the security of accounts and capital accounts, our design stipulates that users should input the account number and password again in every trade.

6.3 Software Quality Attributes

Servers are operable 24/7, hence access to account is always available. Trading, however, can only be done when the market is open. Since the data is real time and there may be slight latencies, so there is a slight margin of error. However, this is usually a problem of the client and not the application. The software must be continued to be updated and maintained by us.

6.4 Business Rules

Administrator can verify the KYC documents, ban users from trading. Users can only buy, sell and edit their profile, and are given no other privileges.

Appendix A: Glossary

BSE: Bombay Stock Exchange
NSE: National Stock Exchange
Share: In financial markets, a share is a unit used as mutual funds, limited partnerships, and real estate investment trusts.
Stock: Stock of a corporation is all of the shares into which ownership of the corporation is divided. Shares are usually collectively known as "stock". A single share of the stock represents fractional ownership of the corporation in proportion to the total number of shares.
MITM: Man-in-the-middle attacks
Stock Broker: A stock broker is a regulated broker or broker-dealer who may provide financial advisory and investment management services and execute transactions such as the purchase or sale of stocks and other investments to financial market participants in return for a commission, markup, or fee, which could be based on a flat rate, percentage of assets, or hourly rate. The term also refers to financial companies, offering such services.
Stock Exchange: A stock exchange or securities exchange is a facility where stockbrokers and traders can buy and sell securities, such as shares of stock, bonds, and other financial instruments.
API: Application Programming Interface, a computing interface that defines interactions between multiple software intermediaries.
CSV(Comma Separated Values), JSON(Javascript Object Notation), XML(Extended Markup Language): Response data formats used by an API to communicate with client.
HTTPS: HyperText Transfer Protocol Secured, a protocol used for secure communication over the Internet.

Appendix B: Field Layouts

An Excel sheet containing field layouts and properties/attributes and report requirements.

Sample sheet with information required to register the customer

Field	Length	Data Type	Description	Is Mandatory
Phone Number	10	Numeric	A valid phone number	Y
Name	-	String	Customer Name	Y
Email ID	8	Alphanumeric	A valid email address	Y
Password	>8	Alphanumeric	Date of Mandate Expiry	Y

Appendix C: Requirement Traceability Matrix

Sl. No	Requirement ID	Brief Description of Requirement	Architecture Reference	Design Reference	Code File Reference	Test Case ID	System Test Case ID