



Ahmedabad
University



Software Requirements Specification

for

TRADING SOFTWARE

20 March 2020

Group - 2

Member's Name

Dhara Vora	: 1741046
Mihir Kanjaria	: 1741065
Kevin Kukadiya	: 1741066
Gaurav Parmar	: 1741072
Shreyash Prajapati	: 1741081
Krushna Shah	: 1741086
Nisarg Gandhi	: 1741099

Table of Contents

Table of Contents	ii
1. Introduction	1
1.1 Purpose	
1.2 Intended Audience and Reading Suggestions	
1.3 Project Scope	
1.4 References	
2. Overall Description	2
2.1 Product Perspective	
2.2 Product Features	
2.3 User Classes and Characteristics	
2.4 Operating Environment	
2.5 Design and Implementation Constraints	
2.6 User Documentation	
2.7 Assumptions and Dependencies	
3. System Features	5
3.1 System Feature	
3.2 Requirement Traceability matrix	
4. External Interface Requirements	7
4.1 User Interfaces	
4.2 Hardware Interfaces	
4.3 Software Interfaces	
4.4 Communications Interfaces	
5. Other Nonfunctional Requirements	8
5.1 Performance Requirements	
5.2 Safety Requirements	
5.3 Security Requirements	
5.4 Software Quality Attributes	

1. Introduction

1.1 Purpose

To provide easy to use software for trading listed instruments like stocks online. Software will also ease brokerage firm burden by automating tedious process around trading.

1.2 Intended Audience and Reading Suggestions

Intended Audience : software developer, people with exposure to the capital market and stock traders

Reading Suggestion

NSE : National Stock Exchange

BSE : Bombay Stock Exchange

Demat Account : Account that holds financial instruments like stock, bonds in electronic form.

Trading : Buying/Selling of listed security

EOD data : End Of the Day rate of stock

MIS : Margin Intraday Square-off

1.3 Project Scope

Software is limited to the stocks and future trading only. Option contracts are out of the scope of this project. While EOD data will be provided as per exchange rate, live data feed will be generated using appropriate algorithms.

1.4 References

- <https://www.ece.rutgers.edu/~marsic/books/SE/projects/TradingLeague/2014-g1-report3.pdf>
- https://gephi.org/users/gephi_srs_document.pdf
- <https://kite.zerodha.com>
- https://www1.nseindia.com/technology/content/nnf/TP_CM_Trimmed_NNF_PROTOCOL_4.1.pdf
- https://www1.nseindia.com/technology/content/nse_schematic.htm

2. Overall Description

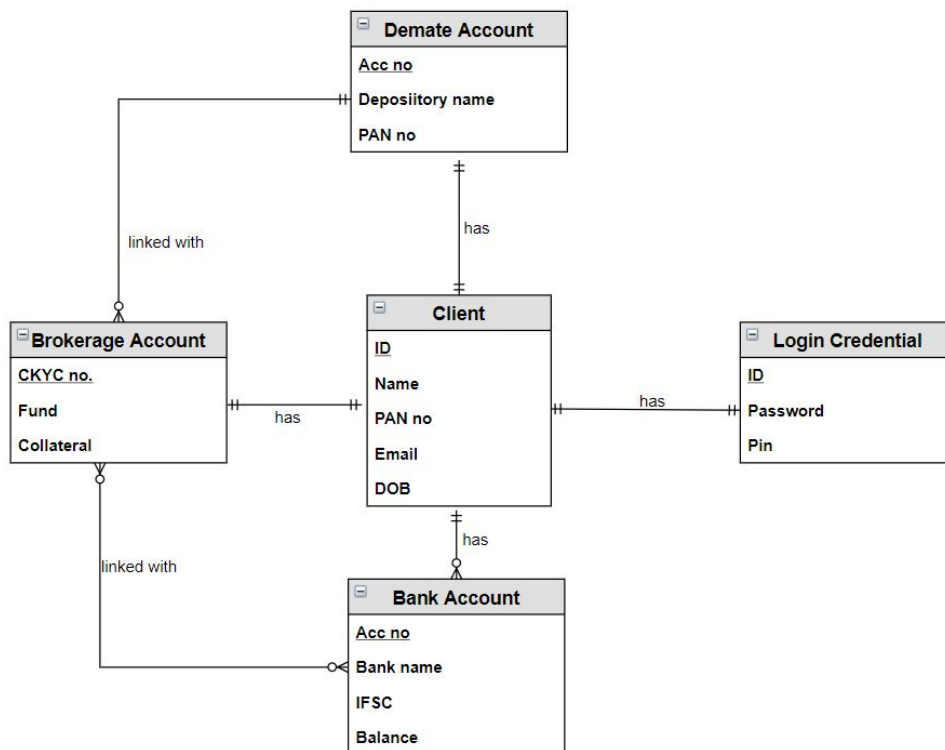
2.1 Product Perspective

A distributed Trading software able to show prices of various stocks in real time. Users can place various types of orders in our trading software. Software is working like a brokerage system in between us and NSE/BSE. After the transaction user can see it's PL statement, Margin to make a good decision for the future.

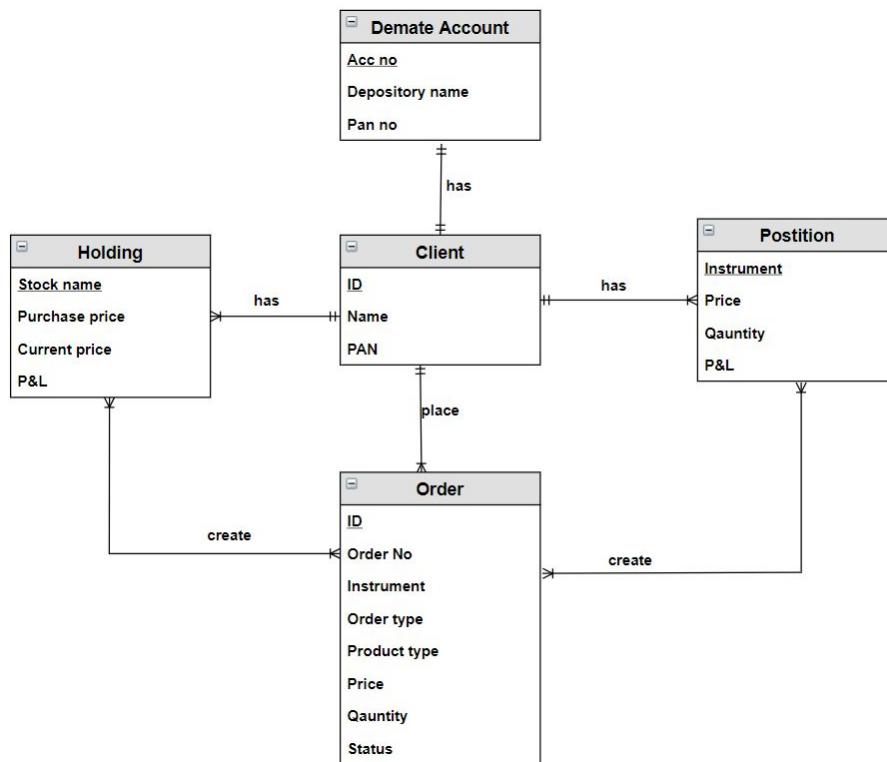
2.2 Product Features

1) ER Diagram:

Phase 1: Login/Sign-up system



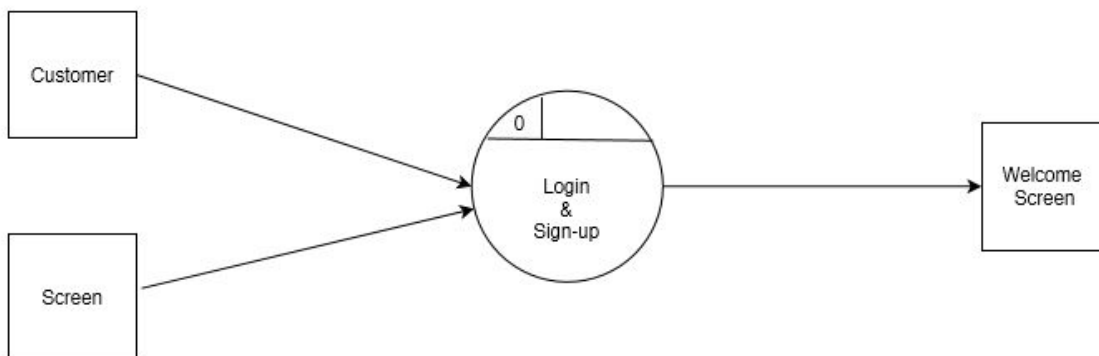
Phase 2: Order Placing and Execution



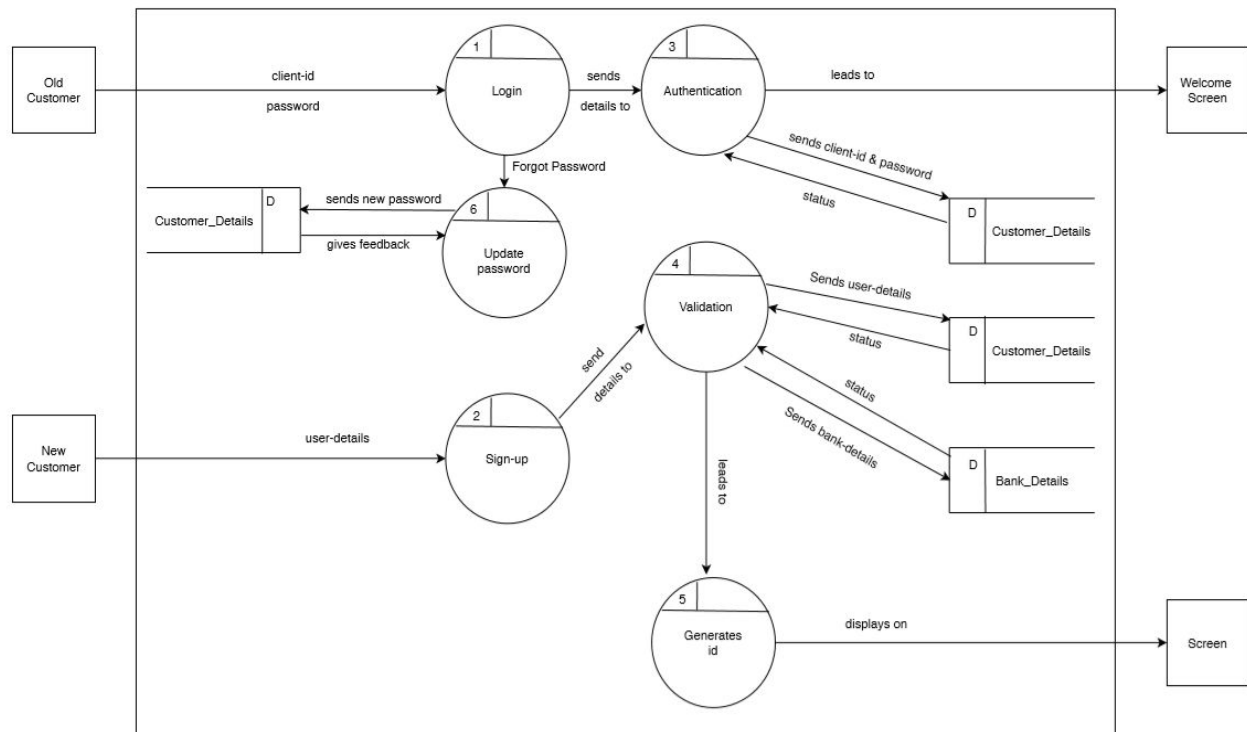
2) Data Flow Diagram (DFD)

Phase 1: Login/Sign-up System

LEVEL 0



LEVEL 1



2.3 User Classes and Characteristics

Here User are mostly the people who are interested to buy and sell stock. Users should not find any difficulty in placing an order. Users should be able to view the trends of prices of various stocks. All customers are treated equally across all demographics. To use this software all the customers need to have just a demat account and valid ID proof.

2.4 Operating Environment

Operating environment for trading software includes the following:-

- 1) Python 3 version on their desktop
- 2) Client/Server System
- 3) database : SQL + databases
- 4) Platform : Windows OS, Linux OS.
- 5) 1MBps Good Internet connection

2.5 Design and Implementation Constraints

We will be designing the GUI implementation for the Stock Exchange. We will be reflecting the non-real time update of the stocks of NSE and BSE as the expense of API is high. Also we will be implementing the charts reflecting the changes at a regular interval of 5 minutes but not candle chart. We will be using structured database SQL for our database management. We will be using python Tkinter for GUI. For security we will be authenticating the user by OTP(2-factor authentication). We will design the floating window view of stocks, equity wise allocation of pie-chart, and manage to display the live MTM margin quote.

2.6 User Documentation

Also with our software we will be providing the user manual of how to use our system(including supported OS, Hardware specification) step by step along with the video-tutorials. In the user manual and video-tutorials we will be providing each and every step for place/sell the order including all types, view different charts reflecting the market prices along with history.

2.7 Assumptions and Dependencies

We have strictly assumed that the prices that our software will be reflecting will be of past market values and not real time stocks values. Our software completely depends on alpha vantage api for our data. Software supports only windows and ubuntu latest versions. Also the transactions that are being done will be reflect only on virtual wallets that are stored in our database pertaining user.

3. System Features

3.1 Features

- Floating market watch window which shows prices of the stocks.
- To visualize price behaviour, we will display it through a live chart
- EOD (end of day) chart with profit and loss statement.
- We will generate reports like Sector wise allocation
- Equity wise allocation pie chart
- Account balance and live MTM margin quote

3.2 Requirement Traceability Matrix

Business requirement number	Technical requirement number	Test Case	Test Steps	Test Data	Expected
BR1	TR1	Verify Login	1). Enter user_id 2).Enter password 3).Click Login	id=example, password=xyz	Login Successful/Login Error
BR1	TR2	Update Login	1). Enter user_id 2).Forgot Password 3).OTP authentication 4).Generates new password	id=example, password=xyz	Update Successful/Update Error
BR1	TR3	Signup	1). Enter user detail 2). Enter Bank details 3). Generates client_id	id=example, password=xyz	Signup Successful/Sign Up Error
BR2	TR4	Buy	1). Does the login 2). Enter order detail 3). Click on buy 4) Validates with NSE/BSE	id=example, password=xyz	placed order successfully/Order failed
BR2	TR5	Sell	1). Does the login 2). Enter order detail 3). Sell the order 4) Validates with NSE/BSE	id=example, password=xyz	order sold successfully/Order failed

4. External Interface Requirements

4.1 User Interfaces

4.1.1 Login or Sign-up Interface :

- Using this interface a new user can enter his/her details to make an account. After making an account the system will give client_id to that user. Using this client_id user can logged into it.
- If the user already has an account then he/she can directly login into it by entering client_id and password.
- If the user forgot the password then by using 'Forgot Password Link' he/she can create the new password. For that, first system will ask the user to enter the email-id and the system will send OTP to that email-id to verify the user.

4.1.2 Order placing and execution Interface :

- User can place order of buy or sell on this GUI. In order user can place various types of order like normal, MIS etc. Once the Order is executed then the user cannot cancel the order in-between.
- After the execution of order, based on the past statistics software can calculate P/L statement, Graph of its purchase and sell and many more functionality. GUI visually shows the history of stock prices and it's trends from past to present.
- User can place order from the graph itself.

4.2 Hardware Interfaces

- PC or Smartphone
- Key-Board
- Ethernet or Wifi Module

4.3 Software Interfaces

Software Used	Description
Operating System	We have chosen Windows OS, Linux OS for its best support and user friendliness.
Database	To save the client detail, their orders records we have chosen SQL + database
Python 3	To implement the project we have chosen python language for its more interactive support

4.4 Communications Interfaces

- To implement this project we are using GUI. For creating client- server communication we are using TCP/UDP protocol.

5. Other Non-functional Requirements

5.1 Performance Requirements

- *The system must be interactive and the delay involved must be less than 6 seconds for establishing database connection.*
- *So in every action-response of the system, there are no immediate delays.*
- There will be a focus on scalability and an efficient system for passing message between the client and the server (ie. for transactions).
- Once the order is placed by a user, NSE/BSE must validate it within 2 or 3 minutes to execute the order.
- The server should be able to serve concurrent requests especially when a large number of users are logged in.

5.2 Safety Requirements

- Information transmission should be securely transmitted to the server without any changes in information.
- There should be a bug tracker available where users can report any bugs they have encountered so that the developers can fix it in the next release.
- Clients must not be able to login in different devices with the same client-id at the same time to avoid consistency.

5.3 Security Requirements

- User security will be a major priority with features for authentication and encryption for sensitive data such as password.
- All user information and records remains intact in our system.
- Our system will send OTP to the user's email-id to verify the user's account.

5.4 Software Quality Attributes

- **Usability :**
 - A key point in the design of this application is ease of use and appeal to the users.
 - We find 'Python GUI' easy to interact with the software.

- **Reliability :**

- In order to ensure that there is no confusion to the user in the case of the internet or server failure, all transactions end with a final confirmation, and no changes to the account are made until after this confirmation.
- Server failure should also be dealt with by keeping backups of user data.

- **Availability :**

- As long as the users have a PC or smartphone that supports this software and a good internet connection, this software is available and can be accessed by anyone anywhere and at any given time.

- **Correctness :**

- The ability of software products to perform their exact tasks, as defined by their specification.

- **Flexibility :**

- The software should be flexible enough to adapt to possible or future changes in its requirements.

- **Robustness :**

- The ability of software systems to react appropriately to abnormal conditions.

- **Maintainability :**

- Software is more easily maintainable if it has high-quality code that is readable and well-documented.
- And it is very important to continuously maintain the software to keep it free from any kinds of malfunctions.