Software Design Document
for
TRADING SOFTWARE

(CSE300 : Software Engineering)

Dhara Vora : AU1741046
Mihir Kanjaria : AU1741065
Kevin Kukadiya : AU1741066
Gaurav Parmar : AU1741072
Shreyash Prajapati : AU1741081
Krushna Shah : AU1741086
Nisarg Gandhi : AU1741099

Date: (05/15/2020)

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Purpose

This software design document for Client End of Trading aims at presenting a detailed view on the whole design about the subsystem of Stock Trading System. There are four intentions on the following.

1. All the software system requirements will be realized in the system design.
2. The development of the system architecture.
3. Let the system adapt the environment and improve its performance
4. Divide the system structure to modules and functions.

1.2 Scope

Purpose of this document is to present a detailed description of conceptual design of the Trading software according to IEEE Std 1016-1998.
The following tips will show its scope.
1.It must give users convenient and effective ways to deal with stocks.
2.Friendly interfaces are also necessary in this project.
3.It can't visit the database directly for it is only a Client End.

1.3 Overview

This document prepared to help reader to understand and visualize the solution to the project presented. This SDD shows how the system software will be structured to satisfy requirements stipulated in the SRS document through viewpoints.

1.4 Reference Material

1) IEEE. IEEE 1016 Software Design Document (SDD) Template for CENG491
2) IEEE. IEEE Std 1016-2009 IEEE Standard for Information Technology –System Design –Software Design Descriptions. IEEE Computer Society, 2009

1.5 Definitions and Acronyms

Client End of Trading	All the client related operations done in this part. The client-end offers a user interface and it has all the functions to provide all the information for the trading software.

Central Trading Software	It is the central module. In our case it is the localhost for the whole system. From which all orders will be decided. This is also responsible for maintaining the database.
ACD	Architecture Context Diagram
GUI	Graphical USer Interface
ER	Entity Relationship Diagram
CD	Class Diagram

2. SYSTEM OVERVIEW

Our system aims to collect people who want to buy/sell orders at market price or limit price on a webApp platform which provides them a user friendly application on this purpose.

The system will be designed with functionalities to help the user to:

2.1.1 Functionality

The following diagrams will show the relation between users' input and each function

Function name	Login the system	
User's input	Username	Password
General output	Success (Function interface)	Failed(Error Dialog Box)

Function name	Register for the system				
User's input	Username and Password	First name, Middle name, Last name	Gende, Phone No.	email-id, Pan number, city	Bank name, Bank Account No.
General output	Success		Failed		

Function name	Buy stocks			
User's input	Stock number (stock name)	Quantity	Capital account	Capital Password
General output	Success(owned stocks and capital change)			Failed(Error Dialog Box)

Function name	Sell stocks			
User's input	Stock number (stock name)	Quantity	Capital account	Capital Password
General output	Success(owned stocks and capital change)			Failed(Error Dialog Box)

Function name	Change Password	
User's input	Enter new password	Re-enter new password
General output	Success	Failed(Error Dialog Box)

Function name	View Stock Information	
User's input	Click on Orders	Click on Portfolio
General output	Success	Failed(Error Dialog Box)

Function name	Buy Order	
User's input	Click on New Order	
General output	Success	Failed(Error Dialog Box)

Function name	Sell Order

User's input	Click on Portfolio and then click on button sell this stock	
General output	Success	Failed(Error Dialog Box)

Function name	See WatchList	
User's input	Add symbol	
General output	Success	Failed(Error Dialog Box)

Function name	Log Out	
User's input	Click on profile and then click on log out	
General output	Success	Failed(Error Dialog Box)

2.1.2 Key Elements of Project Context

The key elements of project context are divided into the problem and goal.

Problem:

Due to falling commission costs over the years, more traders and investors have moved to doing at least some of their own trading and analysis using self-directed trading accounts. This has increased the demand for software that provides trading capabilities, as well as analysis and information resources within the software.
Trading software can provide users with pricing information for assets, special order types, fundamental data, charts, technical analysis indicators, statistics, chat rooms, and other proprietary tools or functions that brokers and software developers use to draw traders to their service.

Goal:

The goal of of our system is:

- ❏ Model with a purpose
- ❏ Assume simplicity
- ❏ Embrace change

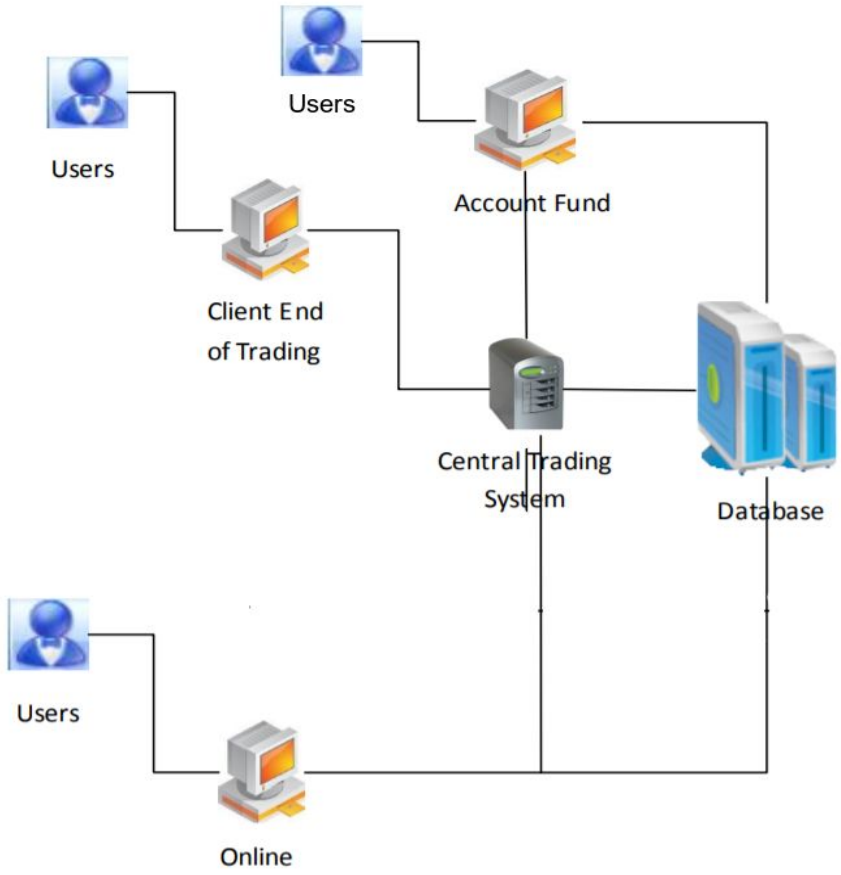
- ❏ Enabling the next effort is our secondary goal
- ❏ Incremental change
- ❏ Maximize stakeholder requirements
- ❏ Multiple models
- ❏ Quality work
- ❏ Rapid feedback
- ❏ Software is the primary goal
- ❏ Travel Light

2.1.3 Technologies used

- **PHP:** To connect front-end and backend
- **HTML:** To create a web page
- **Javascript:** To create responsive, interactive elements for web pages, enhancing the user experience
- **CSS:** To describe the presentation and design of web pages
- **MySQL database :** To store data into tables
- **SQL query:** To fetch data from database
- **Simple Mail Transfer Protocol (SMTP):** For sending OTP to user via mail.

3. SYSTEM ARCHITECTURE

3.1 Architectural Design

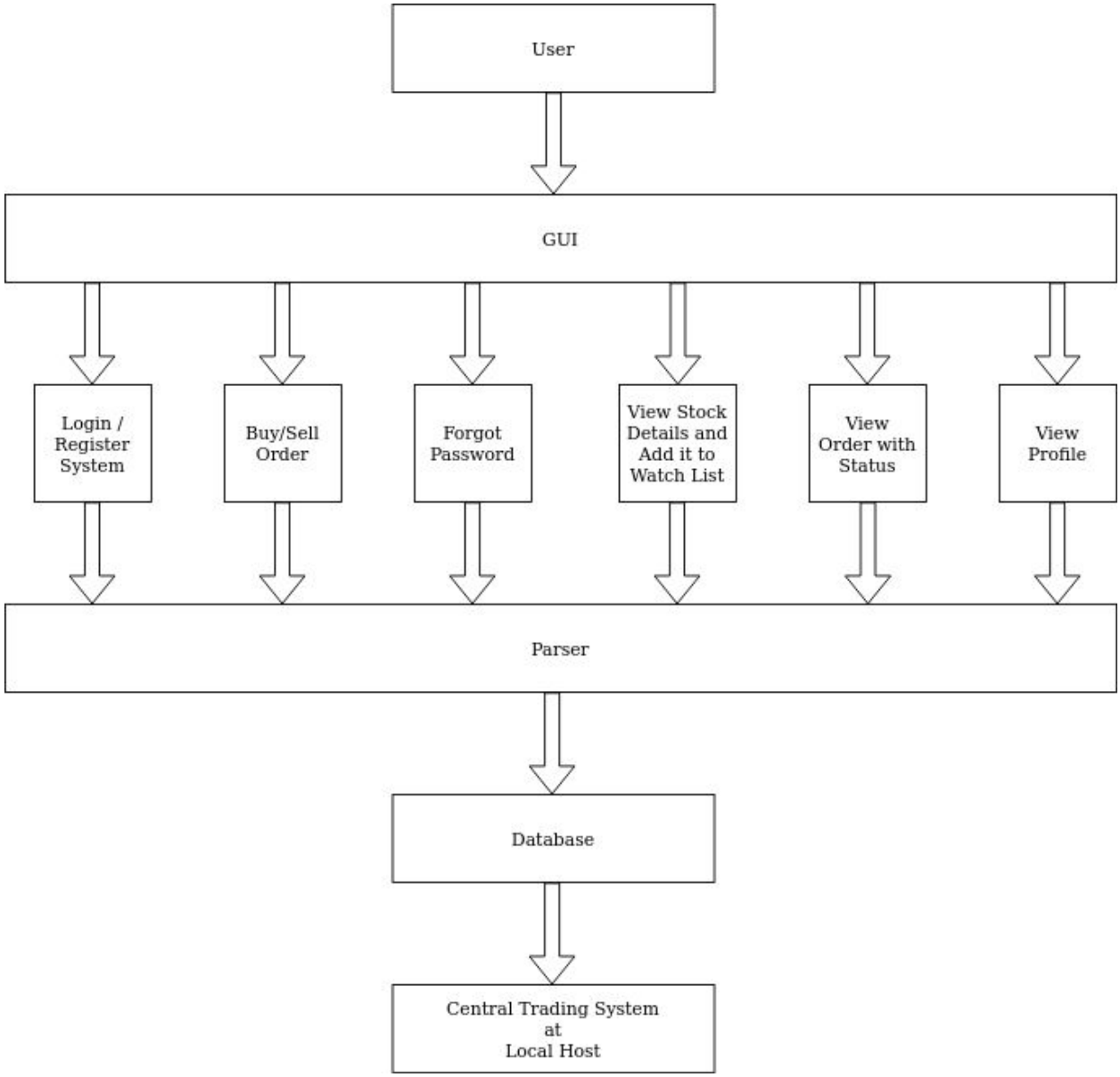


Architecture Diagram of the whole system

In the above diagram, Database contains all kinds of information: Accounts, information of stocks, relations between them and so on. Here, the Client End of Trading is used for users to carry on the transaction operations. Account Fund is used to manage each kind of account information. Online is used to issue the stock information. The Central Trading System is used to handle all kinds of orders from other modules.

As this design document is for the Client End of Trading, relations between Client end of trading and other modules will be analyzed here. The Client end of trading will mainly exchange orders with the Central Trading System and Account Fund to get information of stock, and account from them. Online service is the premise for users to buy or sell stocks because the detailed information of all stocks will be shown there. In a word, all the other modules will influence our design.

We have used layered architecture because it is simple and easy to implement and naturally, most applications work in layers.



Layered Diagram
(Main system with subsystems)

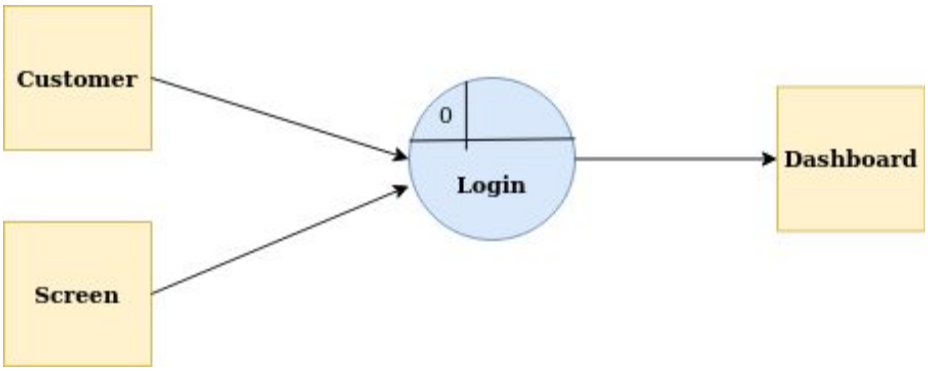
The above diagram shows the layer relations of Client End of Trading. User interface is the out most layer from where user can interact with the system. User interface divided into different components as subsystems. Parser passes the data of these subsystems to the database And it is used to communicate with localhost. We have divided our whole system into subsystems and layers so it is easy to test as

components belong to specific layers. As such, they can be tested separately.

3.2 Decomposition Description

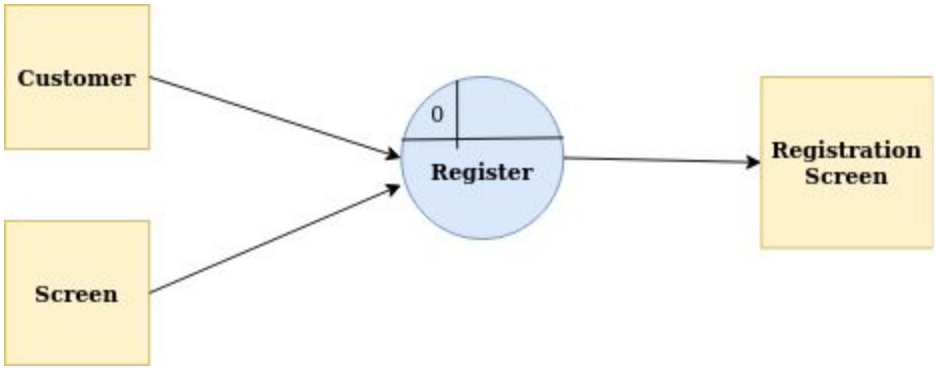
We have divided our main system into different subsystems.
Below is the functional description of each subsystem.

1. **Login System**



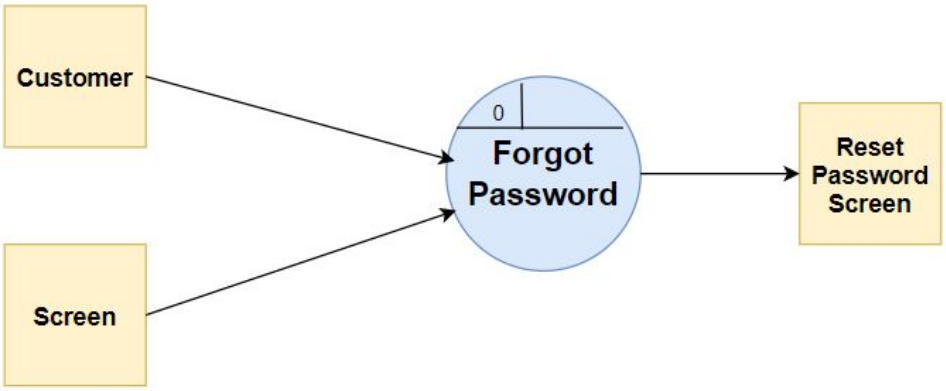
In this case the user enters username and password and goes into Dashboard.

2. **SignUp / Register System**



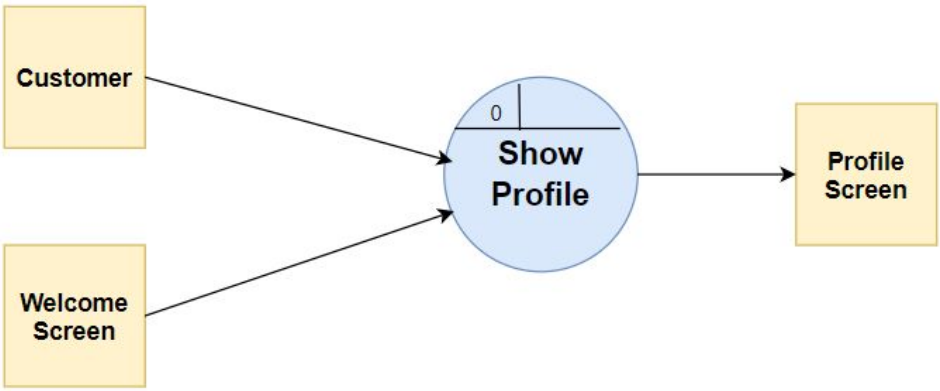
In this case the user feels his details in the registration screen to create a new account.

3. **Forgot Password**



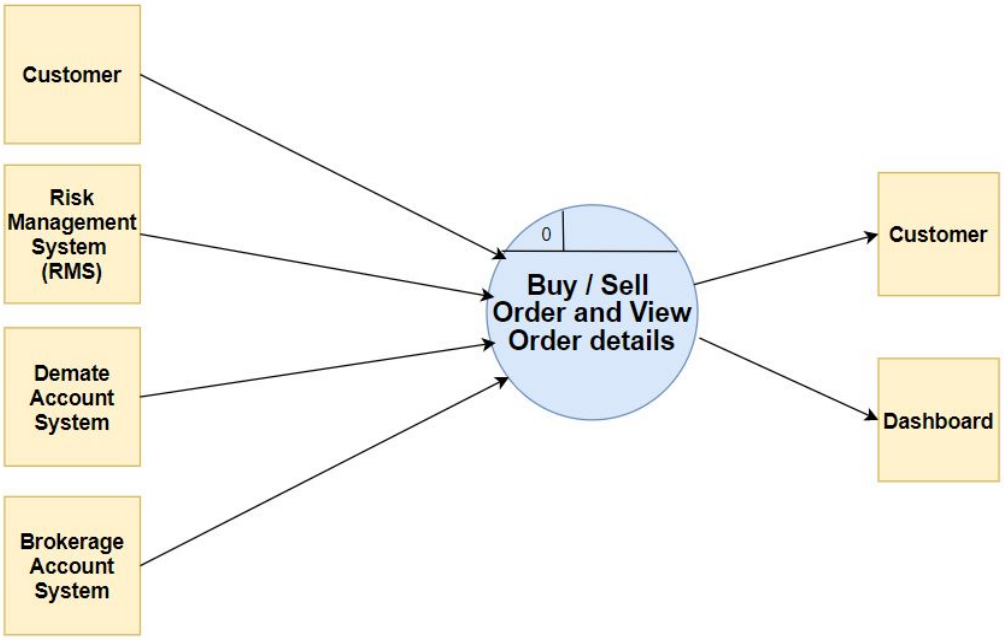
In this case, the user can select the forgot password link to set the new password. And goes to Reset Password screen.

4. **View Profile**



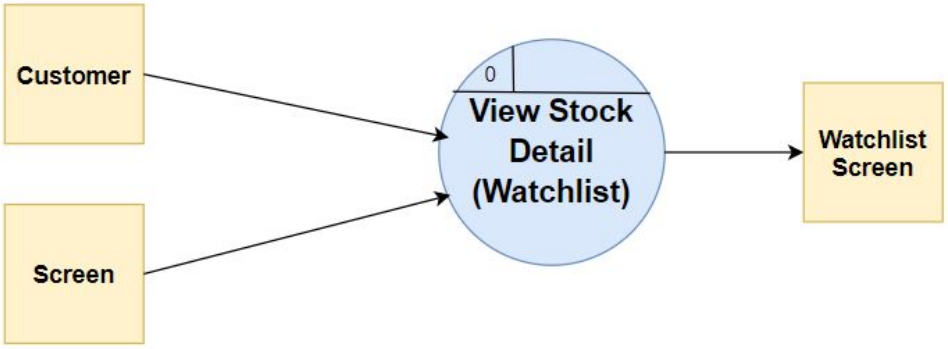
In this case, the user can see his profile by selecting that button on screen.

5. **Buy/Sell Order and View Order details**



In this case, the user can place the order and also view the order details.

6. **View stock details (Watch List)**



In this case, the user can see stock details and also specify the stock in the watch list.

3.3 Design Rationale

Rating and analysis of the common architecture characteristics for the layered architecture pattern are given below :

1. **Overall agility**

   Rating : LOW

Analysis : It is time consuming to make changes in this architecture pattern because of the monolithic nature of most implementations as well as the tight coupling of components usually found with this pattern.

2. **Ease of deployment**
   Rating : LOW
   Analysis : One small change to a component can require a redeployment of the entire application. As such, this pattern does not easily lend itself toward a continuous delivery pipeline, further reducing the overall rating for deployment.

3. **Testability**
   Rating : HIGH
   Analysis : Because components belong to specific layers in the architecture, other layers can be mocked or stubbed, making this pattern is relatively easy to test. A developer can mock a presentation component or screen to isolate testing within a business component, as well as mock the business layer to test certain screen functionality.

4. **Ease of development**
   Rating *:* HIGH
   Analysis *:* Ease of development gets a relatively high score, mostly because this pattern is so well known and is not overly complex to implement

5. **Performance**
   Rating *:* LOW
   Analysis *:* While it is true some layered architectures can perform well, the pattern does not lend itself to high-performance applications due to the inefficiencies of having to go through multiple layers of the architecture to fulfill a business request.

   From the above characteristics, we can see that testability v/s ease of deployment is one of the trade offs. And testability v/s performance is also another trade off.

4. DATA DESIGN

4.1 Data Description

All Data is stored and processed in the Mysql database. Data is organized in various tables. We have normalized our database to the 5th level so we can reduce redundancy in storing large data and access it faster and easier. All entries in tables have at least one entity that uniquely identifies each record.

4.2 Data Dictionary

Table name: bank_account_detail

Field Name	Data Type	Size
client_id	varchar	20
bank_acc_no	varchar	20
balance	float	Not defined
bank_name	varchar	30
flag	tinyint	2

Table name: client

Field Name	Data Type	Size
client_id	varchar	20
pan_number	varchar	20
city	varchar	20

Table name: demate_account

Field Name	Data Type	Size
client_id	varchar	20
demate_acc_no	varchar	20

Table name: demate_account_balance

Field Name	Data Type	Size
account_no	varchar	20
fund	float	Not defined

Table name: email

Field Name	Data Type	Size
client_id	varchar	20
email	varchar	30

Table name: kyc_detail

Field Name	Data Type	Size
pan_number	varchar	10
f_name	varchar	15
m_name	varchar	15
l_name	varchar	15
gender	varchar	10

Table name: login_credential

Field Name	Data Type	Size
client_id	varchar	20
username	varchar	25
password	varchar	25

Table name: mobile_number

Field Name	Data Type	Size
client_id	varchar	20
mobile_number	varchar	10

Table name: orders

Field Name	Data Type	Size
order_id	varchar	20
client_id	varchar	20
order_type	tinyint	2
position_type	tinyint	2
ticker_name	varchar	10
ticker_size	bigint	Not defined
price	float	Not defined
status	varchar	20

Table name: p_L

Field Name	Data Type	Size
client_id	varchar	20
order_id	varchar	20
Total_PL	bigint	Not defined

Table name: stock_detail

Field Name	Data Type	Size
ticker_name	varchar	15
price	float	Not defined

Table name: Holdings

Field Name	Data Type	Size
client_id	varchar	20
ticker_name	varchar	20
ticker_size	bigint	20
price	float	Nort defined

Table name: Positions

Field Name	Data Type	Size
client_id	varchar	20
ticker_name	varchar	20
ticker_size	bigint	20
price	float	Nort defined

Table name: curusr

Field Name	Data Type	Size
id	int	Not defined
usr	varchar	10

Function,trigger and procedure list:

Triggers:

1. Validition_order before insert on order
2. Validition_sellOrder before insert on sellOrder
3. new_demate_account after insert on Client

Functions:

1. is_valid()

   arguments: orderId,clientId,total_money

   return: integer 0 or 1

2. is_Sellvalid()

   Arguments: orderId,clientId,ticker_name,ticker_size,p_type

   return: integer 0 or 1

Procedures:

1. clearing_market_order() arguments: None
2. clearind_limit_order() arguments: None
3. add_position_holdings() arguments: clientId, orderId, price
4. clearing_market_Sellorder() arguments: None
5. clearing_limit_Sellorder() arguments: None
6. sell_position_holdings() arguments: clientId, orderId, closePrice

5. COMPONENT DESIGN

1. Validition_order and Validition_sellOrder
   This two two triggers willc check constraints for valid orders. If all constraints satisfy then it will allow entry to the table else raise required error.

2. is_valid() function
   It will check whether clients have sufficient funds to complete purchase orders.

3. is_Sellvalid() function
   It will check whether clients have sufficient quantity in their account to fulfill sell orders requirements.

4. clearig_market_order() and clearing_market_Sellorder() function
   It will clear pending market order transection in the system

5. Clearing_limit_order() and clearing_limit_Sellorder() function
   It will clear pending limit order transactions according to business logic defined.

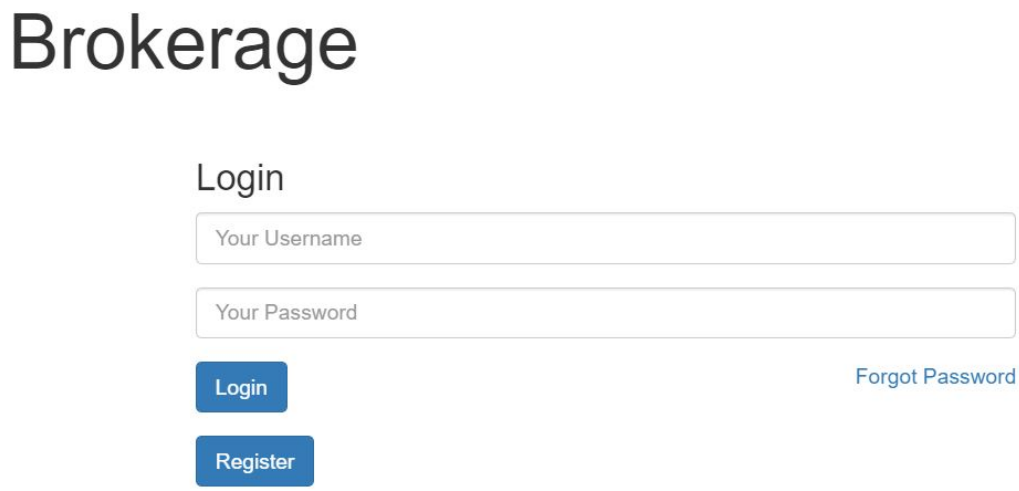
6. HUMAN INTERFACE DESIGN

6.1 Overview of User Interface

When the user opens the web application, it directs to the register / login / forgot password screen. If the user already has an account, then he can directly go to the dashboard after entering the correct username and password. If the user forgot the password then he can use the ‘Forgot Password’ link also to set the new password. For that he has to enter the email id to get OTP. And after verification he can set the new password. User can also use the Register button to create a new account. After successfully login, user can place the MIS / Delivery order of buy and sell at market price or by limiting the price. Based on the price thr order will execute or go in pending status. User can add ticker name in the watch list to view only that stock details. User can also view his profile. And from that he can also logout. Moreover, user can see all his buy order details with its status order window.

6.2 Screen Images

1. **Login Page**



A login interface is necessary in the design. In order to meet the users' requirement, a friendly and clear interface is necessary. Therefore, only the username and password added to the login interface to confirm the security of the user's accounts. After users input all the information, they can click the "login" button. If the user is new then he/she can make the new account by clicking 'Register' and if he/she has already had an account and does not remember the password, then he/she can click the 'Forgot Password' link.

2. **Registration**

Brokerage

Username:

Choose Username *

Password:

Choose Password *

Personal Information:

First Name:

First Name *

Middle Name:

Middle Name

Last Name:

Last Name *

Gender:

- ○ Male
- ○ Female
- ◉ Other

Phone No.:

Phone Number *

Phone No. 2:

Phone Number 2

Email Address:

Email Address *

Pan Number:

Pan number *

City:

City *

Bank Information:

Bank Name:

Enter your Bank Name *

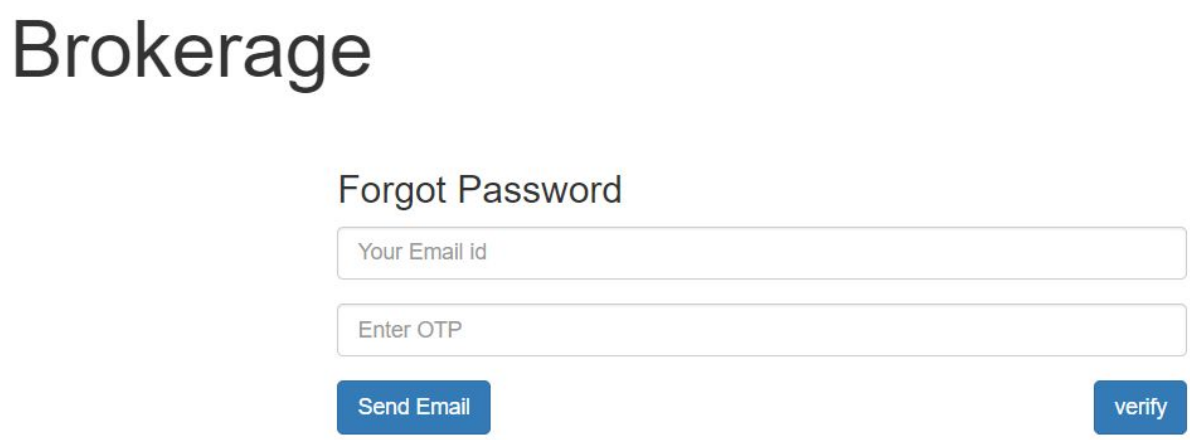
Bank Account Number:

Enter your Bank Account Number *

Submit

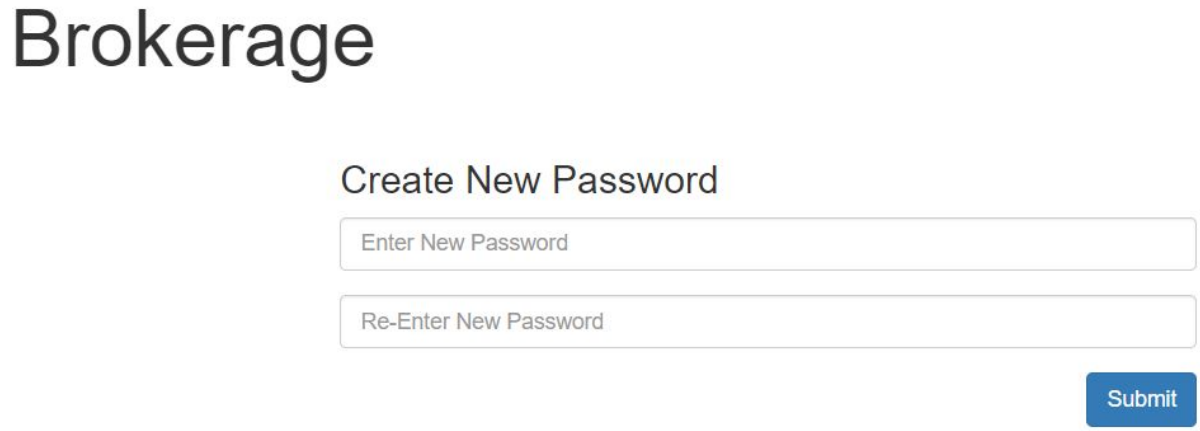
If the user clicks on the ‘Register’ button, then the above screen will be displayed to him. And the user can enter his details and make an account.

3. **Forgot Password**



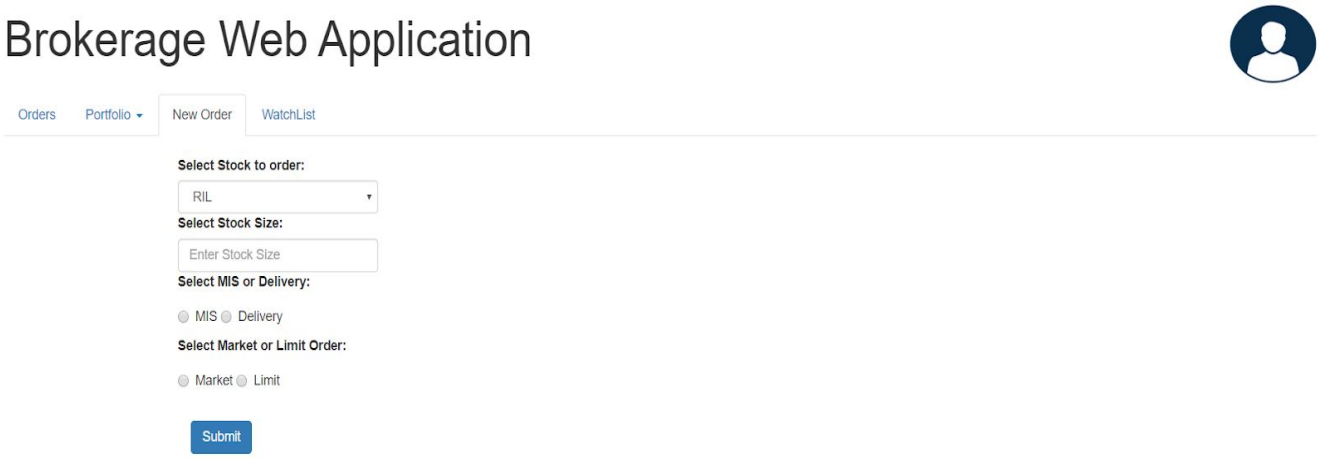
If the user selects the ‘Forgot Password’ link, then the above screen will be displayed. And the user can enter his email id to get otp. Here we have kept OTP verification for the security purpose.

4. **Create New Password**



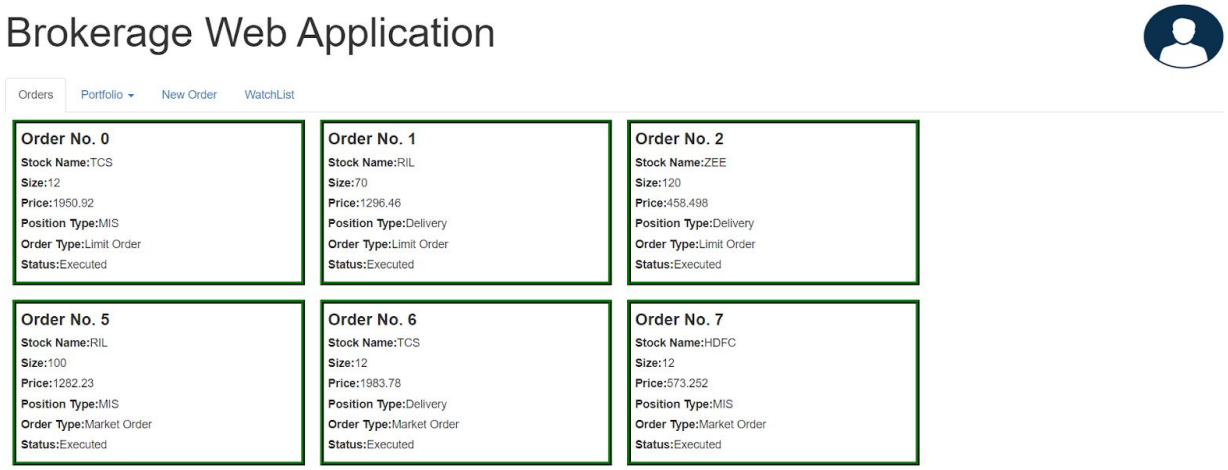
If the user enters the correct OTP then he/she can create his new password from the above window.

5. **Order Placing**



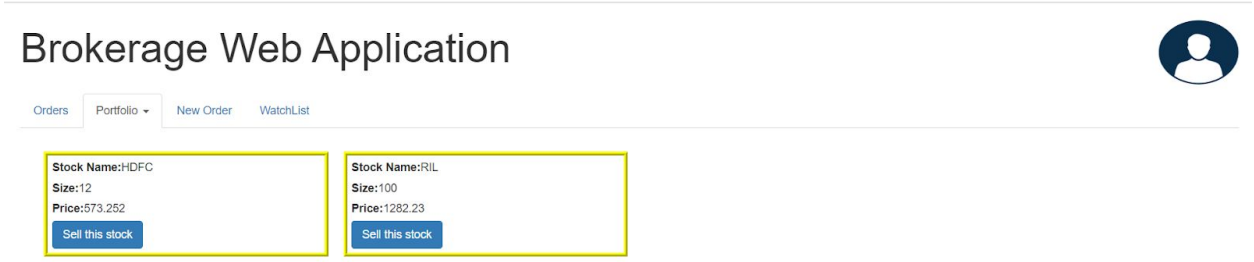
After successfully login, users can place the different types of orders from the above window.

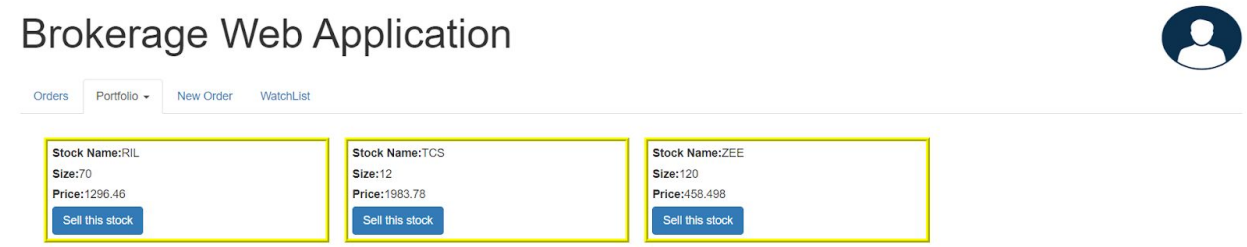
6. Order Window



In the ‘Orders’ tab, the usr can see his order status whether it is executed or pending.

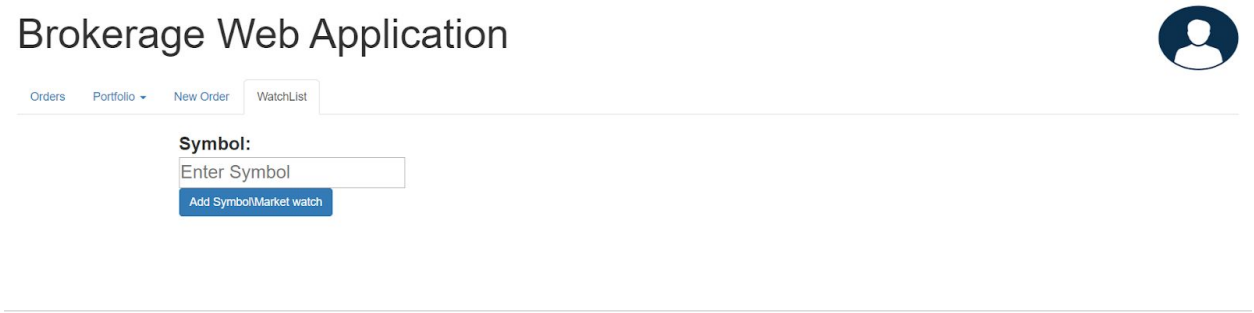
7. Portfolio





From the ‘Portfolio’ link the user can see his positions and holding and from that he can also sell the stocks.

8. **Watch List**



Back

ticker_name	price	Delete
TCS	1991.76	X
INFY	772.384	X
RIL	1276.65	X
ZEE	502.094	X
HDFC	581.393	X

From the watch list user can see stock details and also set the ticker name to see only specific stock details.

9. **Profile**

Your Details	
First Name:	
Middle Name:	
Last Name:	
Email id:	
Mobile number:	
City:	
Client ID:	
Demate Account Number:	
Pan Number:	
Funds:	
Bank Details	
Bank Account Number:	
Bank Name:	

close logout

Using this ‘Profile’ functionality, the user can see his/her details and also logout.

6.3 Screen Objects and Actions

Page	Button/Hyperlink	Action
Welcome Page	Register	Go to Register page
	Log in	Go to the main page - Dashboard when valid username and password is entered.
	Forgot Password	Go to ForgotPassword Page
Register Page	Submit	Go to the previous login page when the user fills form completely.
ForgotPassword Page	Send Email	Send OTP to user via email when user enters correct email id.
	Verify	Go to CreateNewPassword Page when the user enters the correct OTP.
CreateNewPassword Page	Submit	Go to the first Welcome page when the user sets the new password.
Dashboard	Orders	Display the status of all orders to user.
	Portfolio (Drop down list)	Display the holdings and positions.
	New Order	Opens a form, from which user can enter details of order and buy it.
	Watch List	Displays the stock details.
	My Profile (Image)	Go to Profile page

Profile Page	Close	Go to Dashboard.
	Logout	Go to Welcome page.

7. REQUIREMENTS 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	id=example, password=xyz	placed order Successfully /Order failed

			4)Validates with NSE/BSE		
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 / Ord er failed