A

PROJECT REPORT

ON

Trading Bot

Submitted by

Ashishkumar Dobariya (18IT445)
Mayur Jiyani (18IT446)
Ritul Bathani (18IT447)

For Partial Fulfillment of the Requirements for Bachelor of Technology in Information Technology

Guided by

Prof. K.G. Patel

Dec, 2021

Information Technology Department

Birla Vishvakarma Mahavidyalaya Engineering College

(An Autonomous Institution)

Vallabh Vidyanagar – 388120

Gujarat, INDIA

Birla Vishvakarma Mahavidyalaya Engineering College

(An Autonomous Institution)

Information Technology Department

AY: 2021-22, Semester I

CERTIFICATE

This is to certify that the project work entitled **Trading Bot** has been successfully carried out by **18IT445-Ashishkumar Dobariya 18IT446-Mayur Jiyani 18IT447-Ritul Bathani** for the subject **Project I (4IT31)** during the academic year 2021-22, Semester-I for partial fulfilment of Bachelor of Technology in Information Technology. The work carried out during the semester is satisfactory.

Prof. K.G. Patel
IT Department
BVM

Dr. Keyur Brahmbhatt
Head, IT Department
BVM

Acknowledgement

We have designated our entire time and efforts in this project along with its research. However, it would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them for their valuable assistance.

We are extremely grateful to our project guide, **Prof.K.G. Patel**, faculty of Information Technology, for guiding us throughout the project and for the effective doubt solving sessions with patience and knowledge.

We appreciate the people who have willingly helped us out with their abilities. Also thanks to some of our friends and family. who gave us moral support.

Abstract

Algorithmic trading has been blamed for an increasing level of volatility in a number of financial markets. Adoption and deployment of algorithmic trading systems has increased and this is likely to continue, as regulation, competition and innovation drive the development of advanced technological tools. Expert and intelligent systems provide the mechanics for both reacting to and affecting a financial market that is now significantly faster and operating across multiple time zones and markets. Surprisingly, much of this innovation has escaped discussion within the Information Systems research community.

INDEX

List of Figures

List of Tables

Chapter 1:Introduction

1.1 Brief overview of the work:

Algorithmic trading become new and necessary, even compulsory, field in intersection point of computer and finance worlds. Algorithmic trading, also called algotrading, is the use of electronic platforms for entering trading orders with an algorithm which executes pre-programmed trading instructions whose parameters may include timing, price, or quantity of the order, or in many cases initiating the order by automated computer programs.

1.2 Objective:

Algorithmic trading (also called automated trading, black-box trading, or algo-trading) uses a computer program that follows a defined set of instructions (an algorithm) to place a trade. The trade, in theory, can generate profits at a speed and frequency that is impossible for a human trader.

1.3 Scope:

The name of the software product is Trading bot. This product uses computational algorithms to make trading decisions, submit orders and guide the orders after submission. Software is considered to have two kind of analysis which are making decision instantly (HFT) and making decision at the end of the trading day to give proposals. The software product is aimed to complete whole trading transactions such as buy or sell securities. During operating of algorithms, the software records analysis of all profits and losses, logs of each transaction and the monitoring of successive, outstanding or unfilled orders into the database.

1.4 Project Module:

1.4.1. Module description:

1.4.1.1. User module

This module provides functionality of user. To use this system user has to login to system. After that user can define new trade according to his/her choice like adding different type of data and manage their portfolio.

1.4.1.2. System module

This module receives market data according to user's need and it analyses securities according to trade strategy, create trade orders and mange trade orders.

1.4.2. Detailed functions of various modules:

1.4.2.1. User functions: -

- **Login to system**
  When user starts the application, he/she sees login screen first. User is expected to enter username and password. Login is successfully completed if username and password are matched after the confirmation from broker.

- **Define new trade**
  After successful login, user can create new trade by using new trade button.

- **Manage portfolio**
  User can manage his/her capital by adding new capital or removing old capital. He/she can list the active open trades, calculate profit/loss amount of them and add new trades or cancel active ones. Besides, user can list close trades, calculate profit/loss values of them. User can display graphically current status of any open trades when he/she clicks related trade from open trades menu of main screen.

1.4.2.2. System functions: -

- **Get market data**
  Market data is received instantly from stock market via FIX protocol.

- **Analyze securities according to trade strategy**

  Each trade has certain constraints such as strategy, maximum loss bound. Market data are received according to specified trade type. Upon getting certain market data, these market data and constraints of the trade are evaluated in an algorithm. Note that this algorithm is determined based upon the strategy of trade. At the end of evaluation in algorithm, there are three possible results. First of all, software can decide to sell securities according to analysis. Secondly, software can decide to do nothing because of there is no benefit of sell/buy of securities. Finally, software can decide to buy new securities due to expected rise in future.

- **Create trade orders**

  According to analysis of market data based upon trade constraints, there will be created a trade order. Trade order can be sell or buy securities and they can have two types which are pending and post. Note that status of trade order is pending, initially.

- **Manage trade orders**

  After trade order is created with pending status, there are three steps which are receive, validate and confirm order. If these steps are completed successfully, the status of pending order becomes post. After post order is created, it is sent to market exchange via FIX protocol. A response is expected from stock market. If response is positive and order type is buy, capital of user decreases and specified securities of user increases. If response is positive and order type is sell, capital of user increases and specified securities of user decreases. If response is not positive, there is no action. These all communication messages between stock exchange system and the software via FIX protocol, is recorded in related array locations during run time memory. All actions such as transactions, sent and received messages etc. made in program are recorded instantly to database as logs. Any kind of detailed information about system and operations can be found by tracing logs in database.

1.5 Project Hardware/Software Requirements

1.5.1 Hardware Requirements

Sr No.	Hardware Product:	Features
1	Processor	Dual Core processor
2	RAM	Minimum 4GB RAM
3	Hardware Disk	80 GB HDD

1.5.2 Software Requirements

Sr no.	Software Product:	Source:
1	Windows 10	https://www.microsoft.com/en-in/software-download/windows10
2	PyCharm(Editor)	https://www.jetbrains.com/pycharm/
3	Python 3.9.1	https://www.python.org/downloads/release/python-391/
4	NumPy	https://numpy.org/

Chapter 2: Literature Review

Literature Review:

Literature 1:

There are some projects which are similar to our project. However, we will mention from two most popular ones. Firstly, there is a platform named as ALGOTrader. AlgoTrader is a Java-based algorithmic trading platform that enables trading firms to rapidly develop, simulate, deploy and automate any quantitative trading strategy for any market. It gives users flexible control of high-speed, fact based trading for consistent, good results.

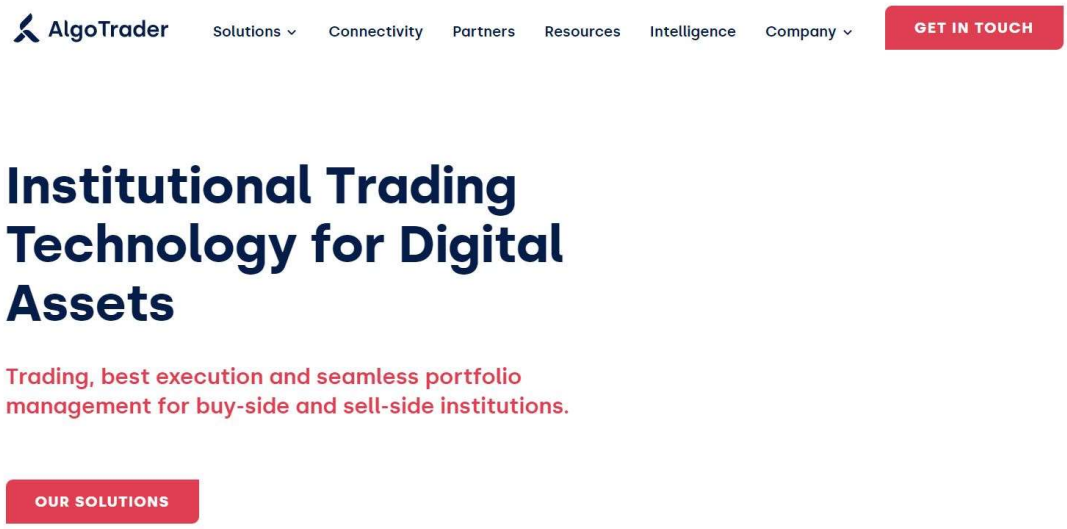


Figure 2.1: AlgoTrader home page

Literature 2:

Secondly, there is a platform named as 5Paisa Algo Trading. It is one of the premier services and facility provided by the firm. The Algo trading platform of 5Paisa has been applauded for its abundance of features and type of strategies which helps the investors to earn well.

The 5Paisa Algo Trading features include –

- The infrastructure of the Algo trading system offered by 5Paisa is advanced and has some cutting-edge technologies in it
- APIs are there for the .NET, C++, and other programming languages
- There is assistance feature for the strategy coding
- You can develop and personalize strategies on this platform
- Assistance available for the stock exchange approval
- For testing strategies, there is a live paper trading facility available as well.

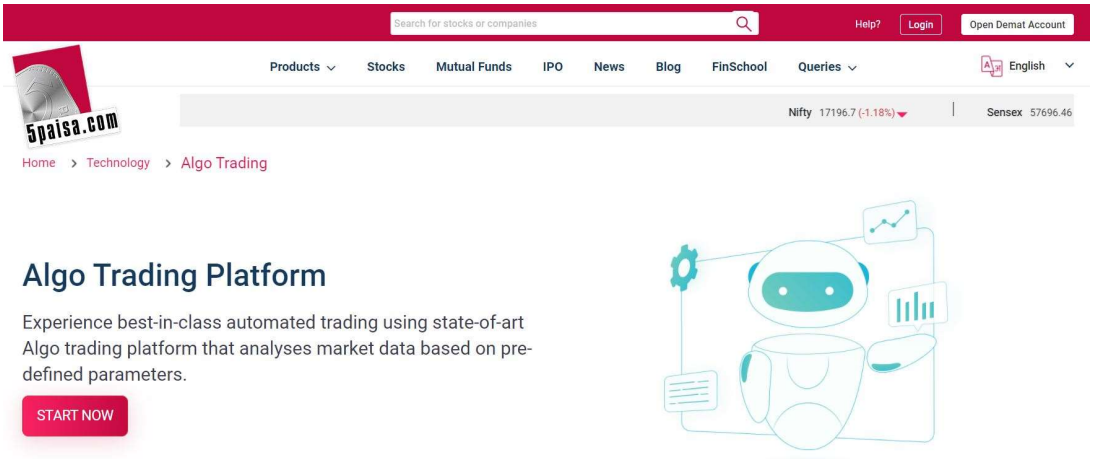


Figure 2.2: 5Paisa home page

Literature 3:

SquareOff provides fully automated Trading Bots that will place all trade entries without any manual intervention in your own Trading Account based on inbuilt strategies.

Trading strategies built on statistical and mathematical models have historically offered higher returns than their benchmarks and mutual funds.

SquareOff built mathematical & statistical models that can provide superior returns and we completely automated it, so the system will place entry, stop loss & targets automatically based on the strategy.

Link: https://squareoff.in

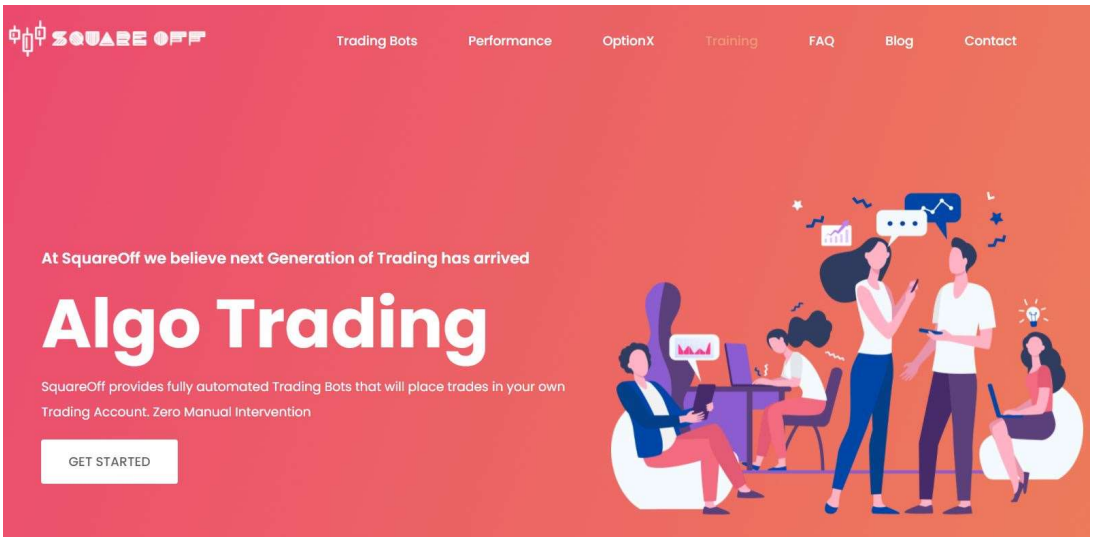


Figure 2.3: SquareOff home page

Literature 4:

Tradetron is one of the best algo trading platform in the algo market where they are providing bundel of the strategies apart from straddels and straingels like Bank Nifty Wave Hamper, Nifty Kuber v2 etc. which are depending upon certain rule of them.

Link: https://tradetron.tech

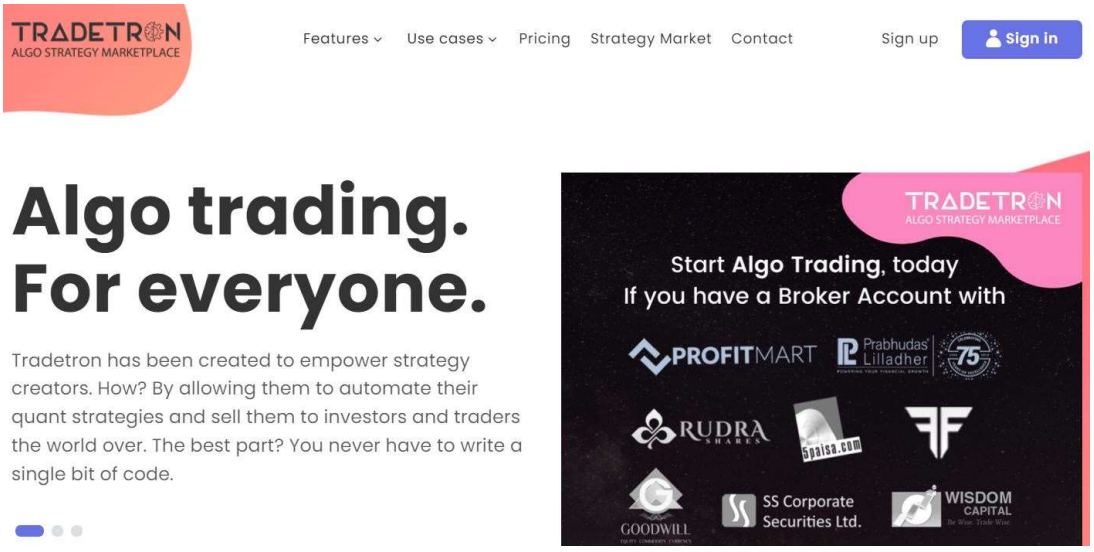


Figure 2.4: Tradetron home page

Literature 5:

Algobull is also one of the best algo trading platform for trading in indian and us stock market as well as commodity market. They take a charge for using their platforms and their strategies. charges may depend on strategies as well as investor categories like retail, HNI (High net worth investors) etc.

Link :https://algobulls.com

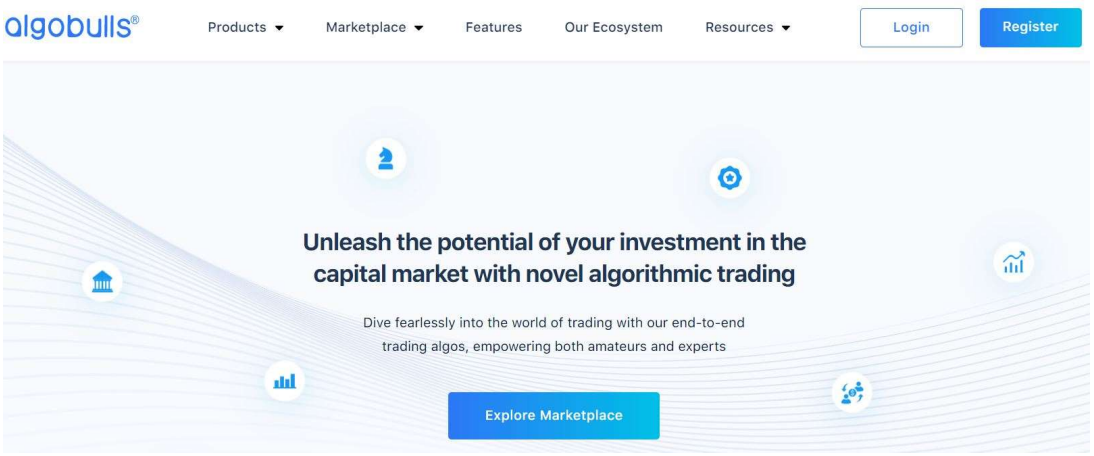


Figure 2.5: 5Paisa home page

Chapter 3: System Analysis & Design

3.1. Comparison of Existing Applications with your Project with merits and demerits

The sensibull platform suggests a list of strategies based on a trader's market view. Further, it provides user with all essential information like trade, strike prices, risk, profit and loss potential etc. User can also compare different Option strategies to find the right one.

- Merits
  - Brings the best of a strategy engine and trading platform
  - Packed with useful features like Option Analyzer, Futures Conversion, Event Calendar etc.
  - Makes executing complex strategy simpler with a single click
  - Sleek and simple interface

- Demerits
  - Filtering of strategies based on risk and reward is not possible.
  - Traders need a good understanding of option strategies to optimally use the platform
  - Platform pricing makes it unsuitable for small and occasional traders.

this system is similar with our project.

3.2. Project Feasibility Study

Preliminary investigation examines project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All systems are feasible if they are given unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility

- Operational Feasibility
- Economic Feasibility
- Implementation Feasibility
- Market Feasibility

3.2.1. Technical Feasibility:

Technical feasibility determines whether the work for the project can be done with the existing equipment, software technology, and available personnel. The technical feasibility of the proposed project refers to the software and hardware requirements. Project will be developed using frontend technologies like HTML, CSS and bootstrap and for backed Django Framework will be used. The proposed project can be access by any type of browser.

3.2.2. Operational Feasibility:

As the system provides various functions, it is important to measure the feasibility of each function. We tested this system in different browser and if user want to really enjoy accessing the website we recommending Google Chrome or Mozilla Firefox.

3.2.3. Economic Feasibility:

Economic feasibility determines whether there are sufficient benefits to make the cost acceptable, or is the cost of the system too high. The software used to develop this application are free and open source. It is assumed that the users are already having a desktop with internet connection.

3.2.4. Resource Feasibility

This is also an essential part of a feasibility study. It includes questions regarding the time required to complete the project, type and amount of resources required and dependent factors. For this, to develop this system resource like brokerage account is available and the user just needs to have a desktop with an active internet connection.

3.2.5. Market Feasibility

A market feasibility study is for analyzing that the system we made is according to the trends of the current time or there should be some changes in the system. Market

feasibility can be a deciding factor before the deployment of the system. It decides that the system is as per the market requirements or it has to be modified for some more extends.

3.3. Project Timeline chart

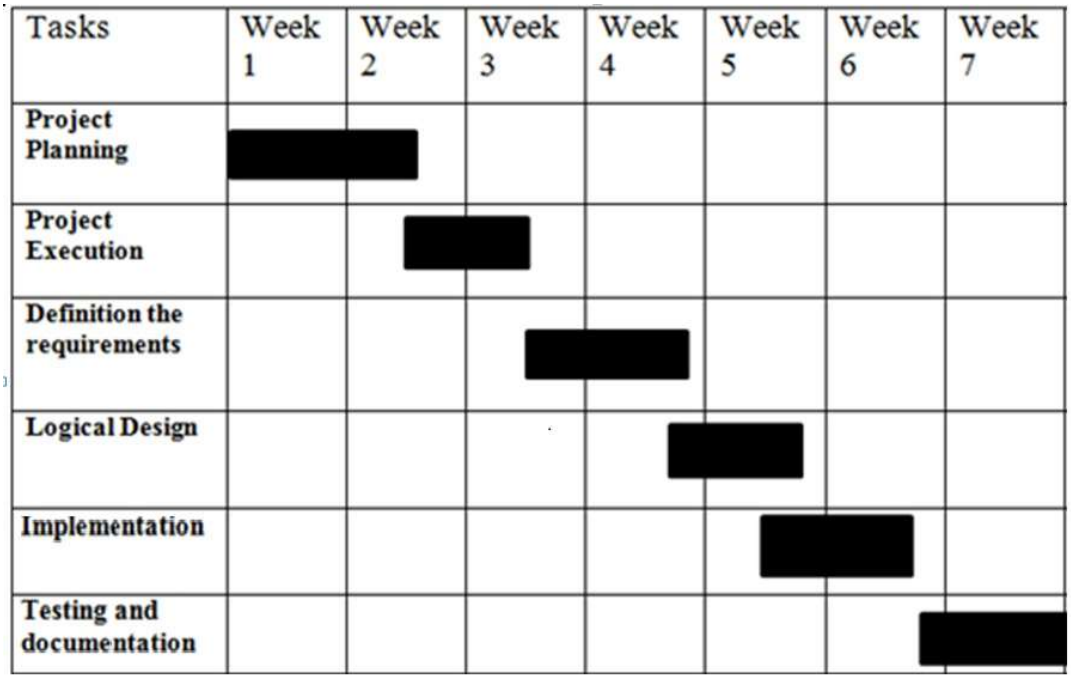


Figure: Project timeline chart for trading bot

This is the weekly basic schedule of the project starting from the first SRS Document.

1 – 2 Week: Team management and project selection. Searching required information from related internet source pages and source books to understand system execution and design.

2 – 3 Week: Finishing search and starting to design of the system for server side.

3 – 4 Week: Finishing system design for client side and starting implementation of server side. Preparation and writing of software requirements specification.

4 – 5 Week: Finishing both client and server side demos and starting to test the system. Also starting to bug fixing in same time with testing.
5 – 6 Week: Finishing algorithm searches and starting to implement them on system.

6 – 7 Week: Finishing implementation and starting to bug fix of analyzed data results from algorithms. Fixing all bugs.

3.4. Detailed Modules Description

3.4.1. User functions: -

- **Login to system**
  When user starts the application, he/she sees login screen first. User is expected to enter username and password. Login is successfully completed if username and password are matched after the confirmation from broker.

- **Define new trade**
  After successful login, user can create new trade by using new trade button.

- **Manage portfolio**
  User can manage his/her capital by adding new capital or removing old capital. He/she can list the active open trades, calculate profit/loss amount of them and add new trades or cancel active ones. Besides, user can list close trades, calculate profit/loss values of them. User can display graphically current status of any open trades when he/she clicks related trade from open trades menu of main screen.

3.4.2. System functions: -

- **Get market data**
  Market data is received instantly from stock market via FIX protocol.

- **Analyze secureties according to trade strategy**
  Each trade has certain constraints such as strategy, maximum loss bound. Market data are received according to specified trade type. Upon getting certain market

data, these market data and constraints of the trade are evaluated in an algorithm. Note that this algorithm is determined based upon the strategy of trade. At the end of evaluation in algorithm, there are three possible results. First of all, software can decide to sell securities according to analysis. Secondly, software can decide to do nothing because of there is no benefit of sell/buy of securities. Finally, software can decide to buy new securities due to expected rise in future.

- **Create trade orders**

  According to analysis of market data based upon trade constraints, there will be created a trade order. Trade order can be sell or buy securities and they can have two types which are pending and post. Note that status of trade order is pending, initially.

- **Manage trade orders**

  After trade order is created with pending status, there are three steps which are receive, validate and confirm order. If these steps are completed successfully, the status of pending order becomes post. After post order is created, it is sent to market exchange via FIX protocol. A response is expected from stock market. If response is positive and order type is buy, capital of user decreases and specified securities of user increases. If response is positive and order type is sell, capital of user increases and specified securities of user decreases. If response is not positive, there is no action. These all communication messages between stock exchange system and the software via FIX protocol, is recorded in related array locations during run time memory. All actions such as transactions, sent and received messages etc. made in program are recorded instantly to database as logs. Any kind of detailed information about system and operations can be found by tracing logs in database.

3.5 Project SRS

3.5.1 Use Case Diagrams

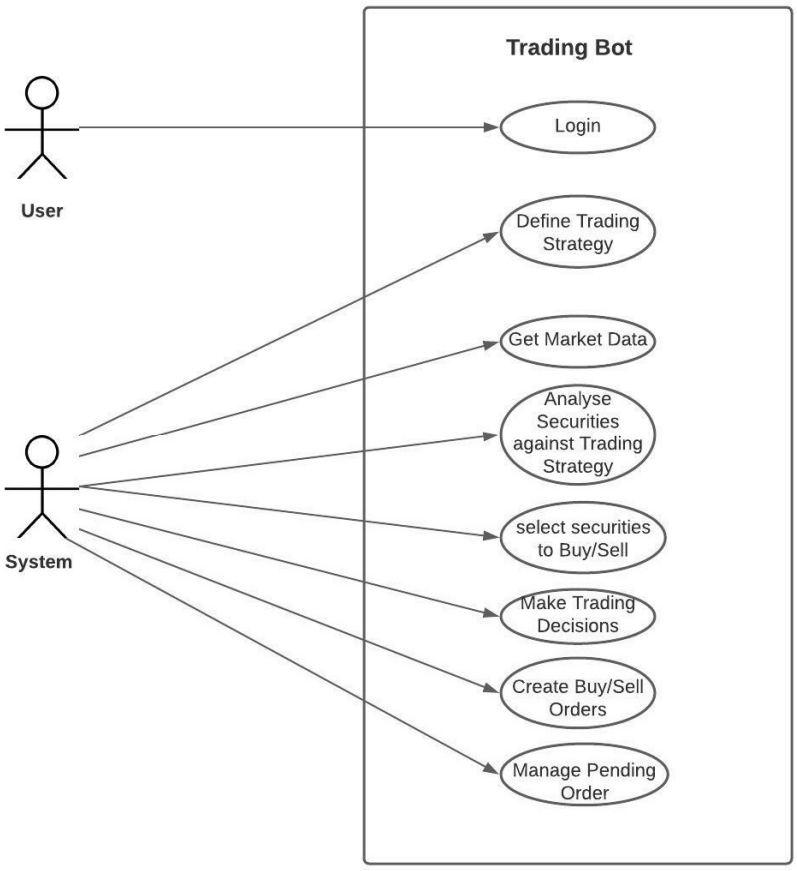
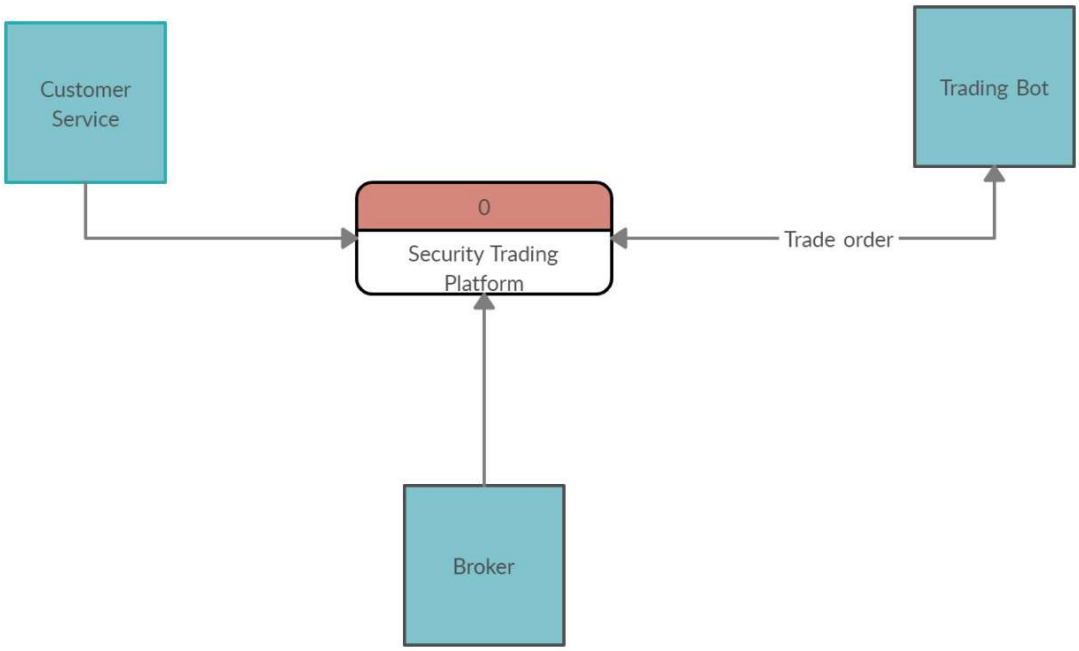


Figure: Trading bot use case diagram

Use case diagrams are a common way to communicate the major functions of a software system. A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involve. Above figure we have two actors one is User and another one is System. Here User associates with only login and all other task is take placed by System actor.

3.5.2 Data Flow Diagrams

Level 0



Level 1

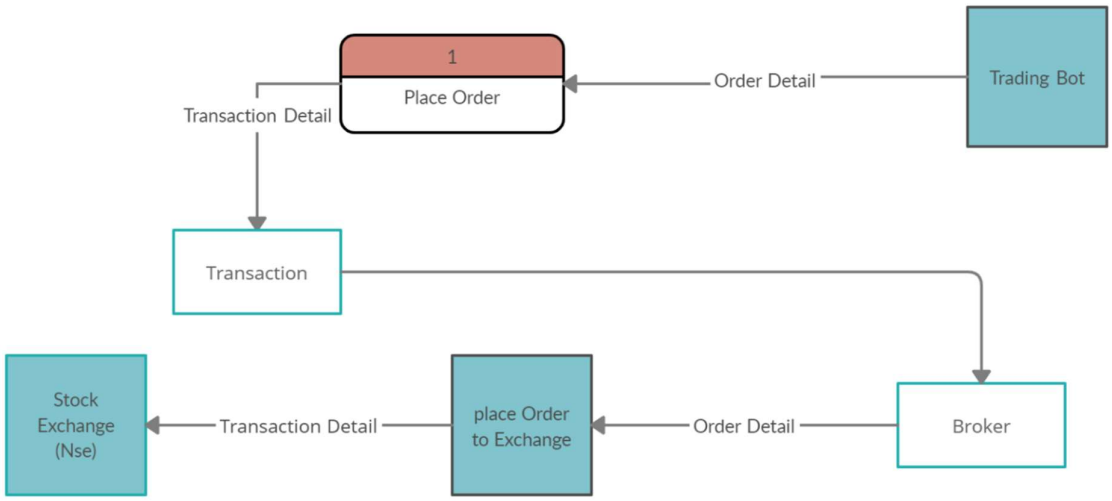


Figure: Dataflow diagram for trading bot

DFD provides the functional overview of a system. Starting from an overview of the system it explores detailed design of a system through a hierarchy. DFD shows the external entities from which data flows into the process and also the other flows of data within a system. It also includes the transformations of data flow by the process and the data stores to read or write a data. In Level 0 diagram external entities customer service, broker and is data flowed with centered security trading platform. In Level 1 shows external entity Trading bot data flowed to end point Stoke Exchange.

3.5.3 Class diagram

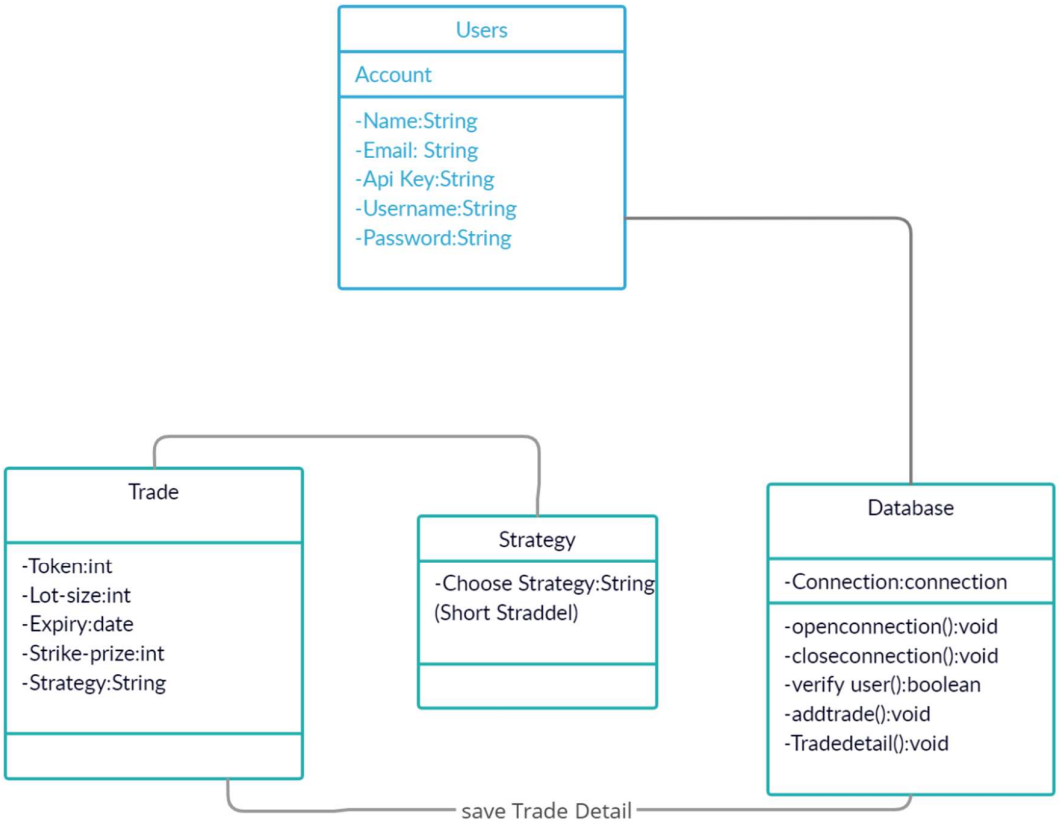


Figure: Class diagram for trading bot

The class diagram is the main building block of object-oriented modelling. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed. The top compartment contains the name of the class. Here we have Trade, Strategy, Database and Users class. The middle compartment contains the attributes of the class. Here we have private visibility type of attribute so it represents '-' before attribute. In Database class openconnect(), closeconnection(), addtrade() are the operations.

3.5.4 Entity Relationship Diagrams

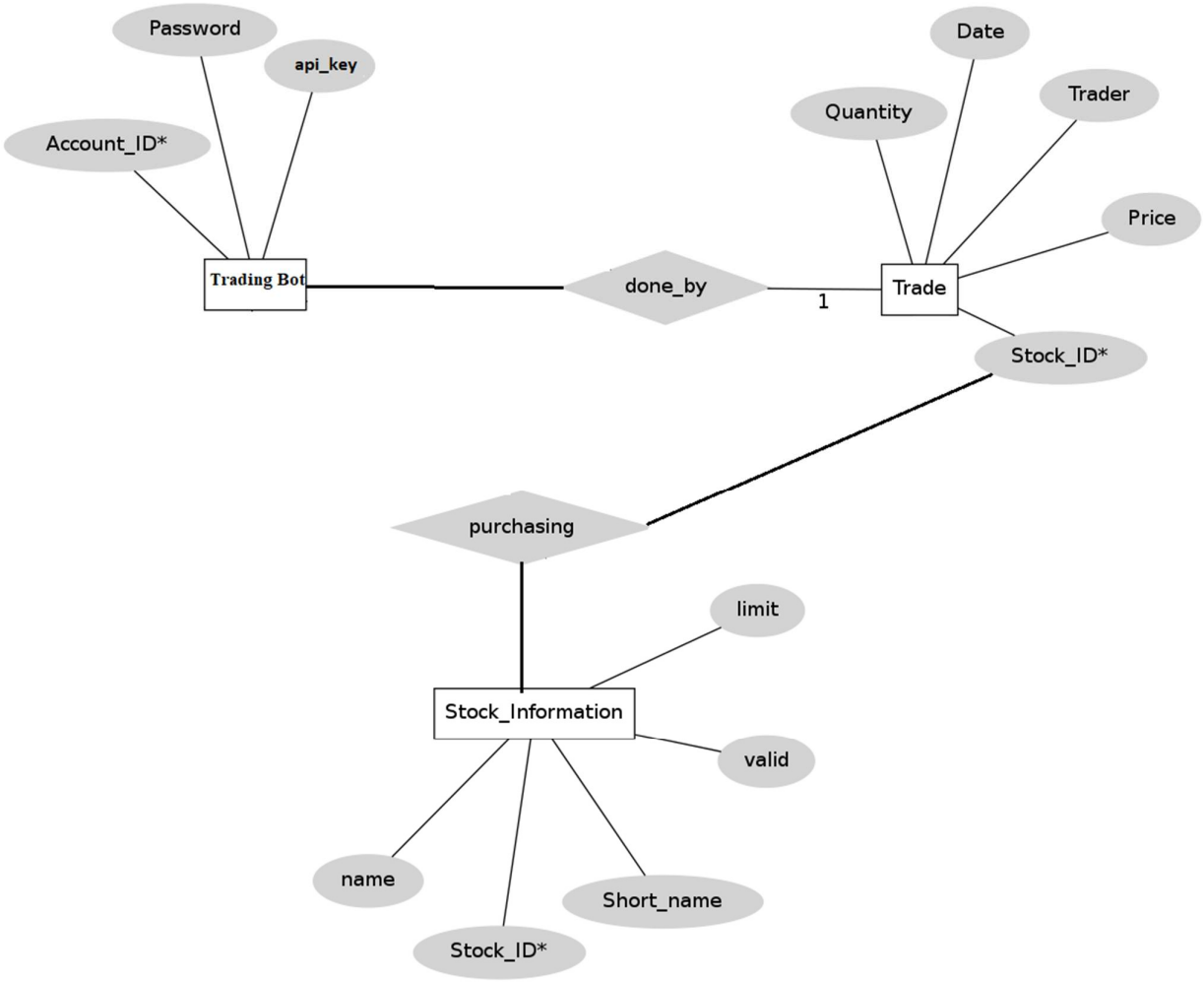


Figure: Entity relationship diagram for trading bot

Entity-Relationship model is used to represent a logical design of a database to be created. We represent the attributes, entities and relation using the ER diagram. Using this ER diagram, table structures are created, along with required constraints. Here we have Trading Bot, Trade and Stock_Information are entities. Stock_ID attribute is purchasing relation with entity Stock_Information.

3.6 Data Dictionary

Table name: Customer Information

Sr. No.	Name	Datatype	Constraint	Description
1	Name	Varchar()	Primary key	Name of the user
2	Email	Varchar()	Not null	Email
3	Broker Name	Varchar()	Not null	Name of the broker
4	Broker Username	Vatchar()	Not null	Client id
5	Broker Password	Varcher()	not null	Broker terminal password
6	Broker Api	Varchar()	not null	Broker algo platform api key
7	Password	Varchar()	not null	Terminal password

Table 3.1: Customer Information

Table name: Order Information

Sr. No.	Name	Datatype	Constraint	Description
1	Email	Varchar()	Not null	Email
2	Username	Varchar()	Not null	Client id
3	Order id	Number	Not null	Order Number
4	Variety	Varchar()	Not null	Intraday or Cary forward
5	Trading Symbol	Varchar()	Not null	Symbol of Security
6	Transaction Type	Varchar()	Not null	Buy or Sell

7	Exchange	Varchar()	Not null	Nse,bse,nfo
8	Ordertype	Varchar()	Not null	Market or limit
9	Product type	Varchar()	Not null	Intraday of holding
10	Duration	Varchar()	Not null	Day or Ioc
11	Quantity	Number	Not null	Number of lots(1Lot=25 share of bnf)

Table 3.2: Order Information

Chapter 4: Implementation and Testing

Homepage

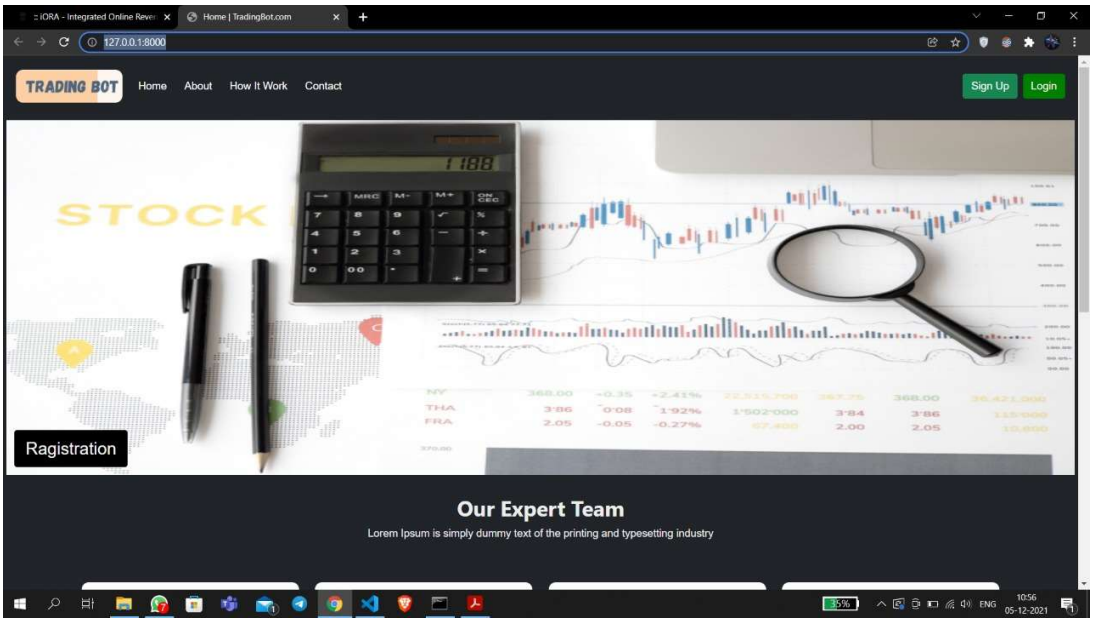


Figure 4.1: Home page.1

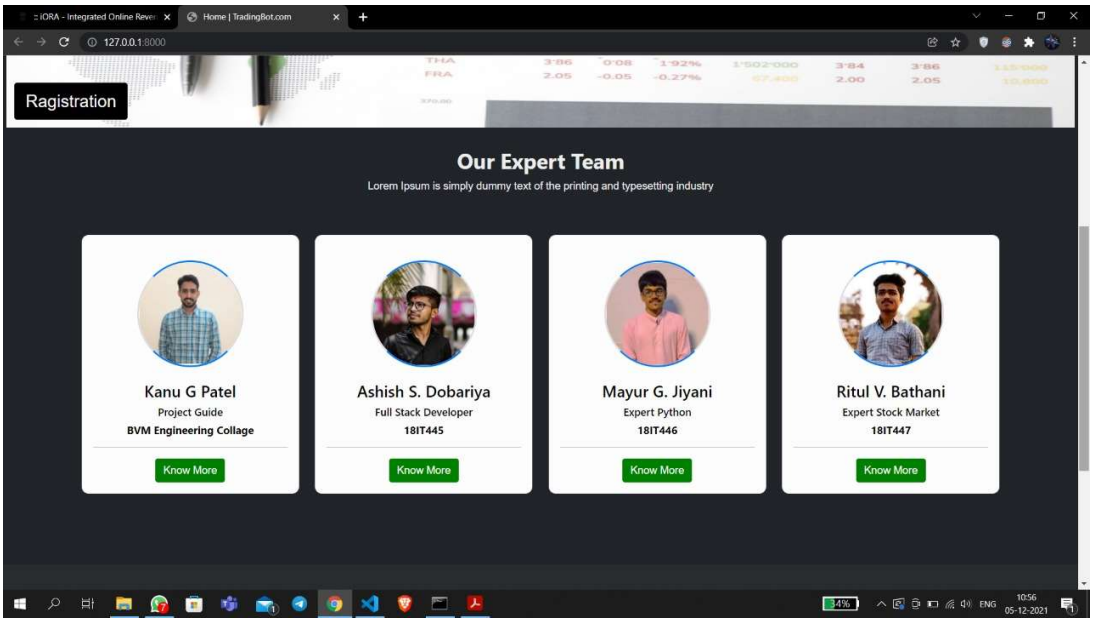


Figure 4.2: Home page.2

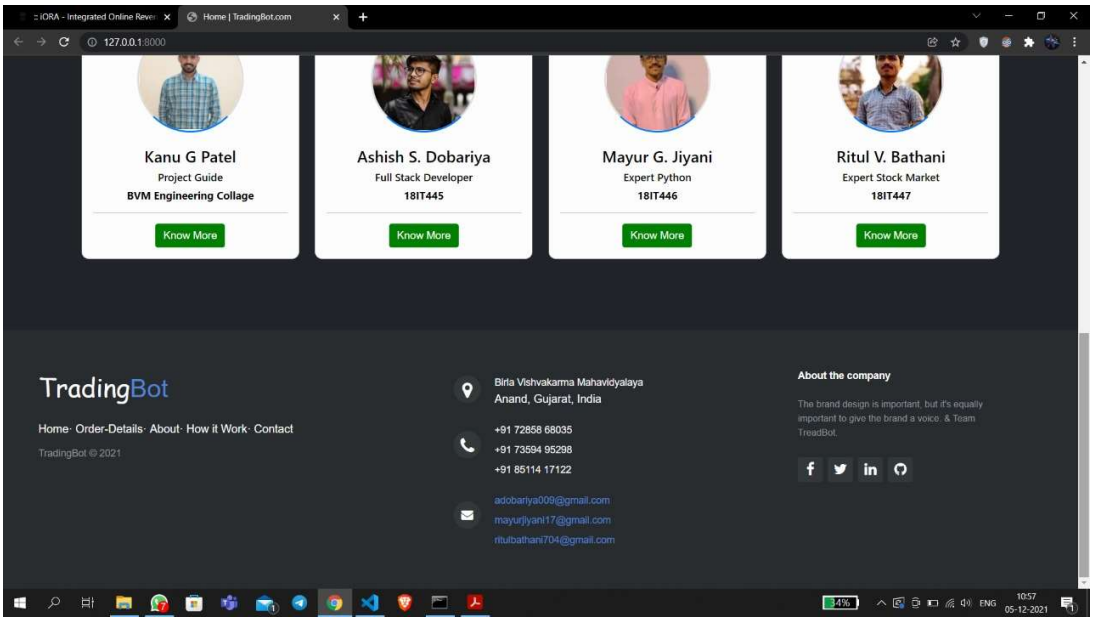


Figure 4.3: Home page.3

Here above figure shows home page of the system. In header section, in menu bar we have Home, About, How it works and Contact link and in right hand side Sign Up and Login button is there. At middle section we have information of expert team. In footer section contact information and social media connect icon is there.

Login page

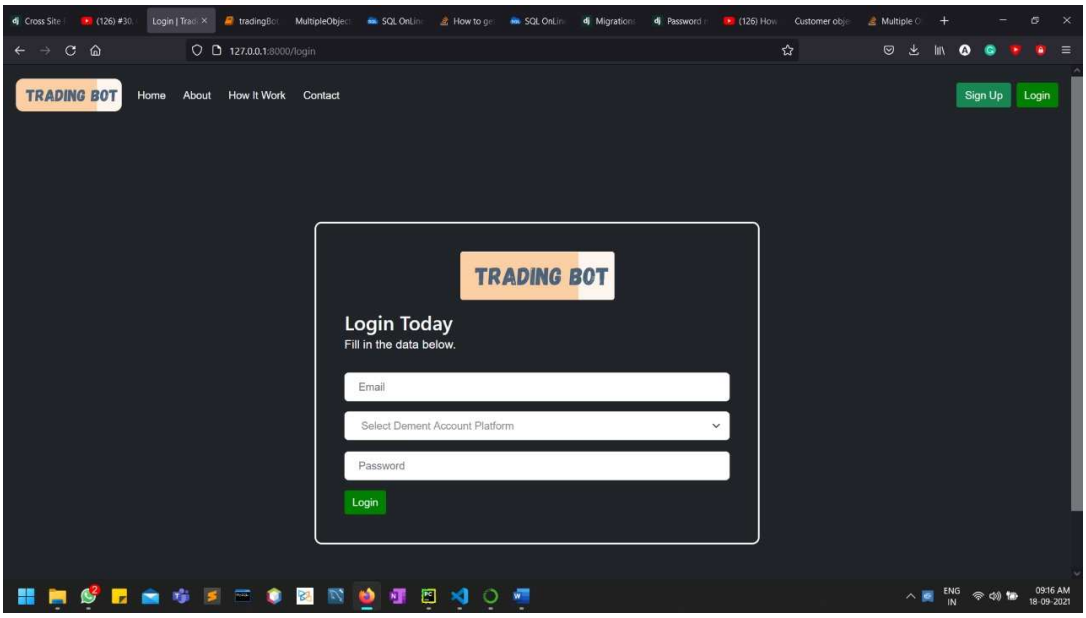


Figure 4.4: Login page

Above figure 4.4 represents login page. After clicking on Login button user redirect to above page. Here user has to enter email, password and have to add demat account platform. If user has not registered than has to register first to access the system.

Register page

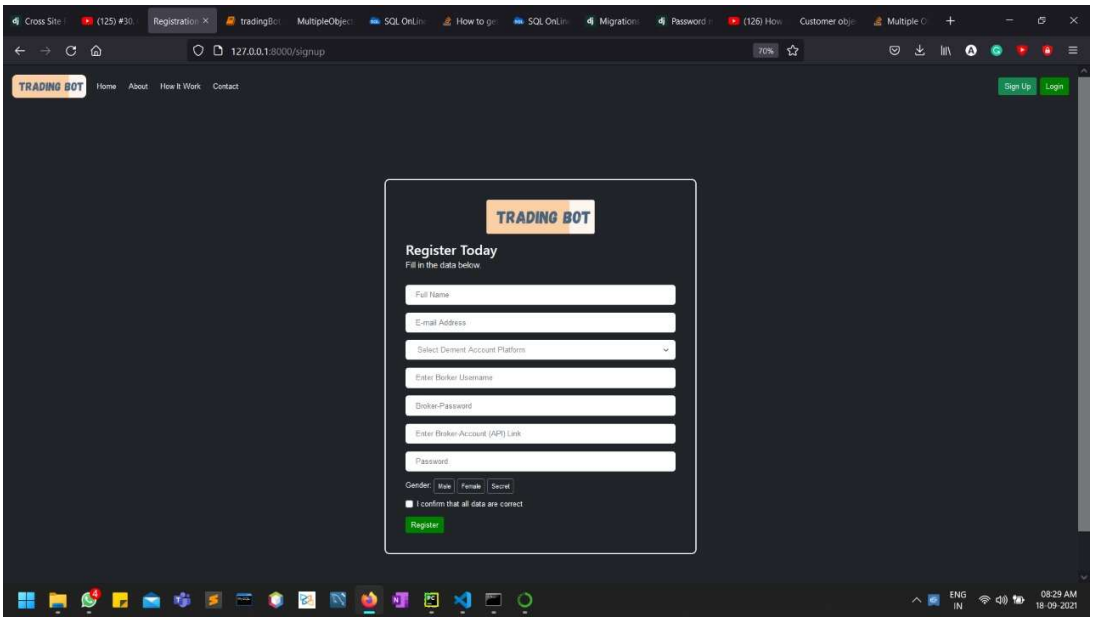


Figure 4.5: Register page

Above figure 4.5 show register page. Here has to submit the input fields. First is to add full name, second is email address, then has to select demat account platform like zerodha, angel one, 5paisa etc. Next field is Broker username which is provided by broker. Then password, at the end after selecting gender has to click on Register button.

Tradebook

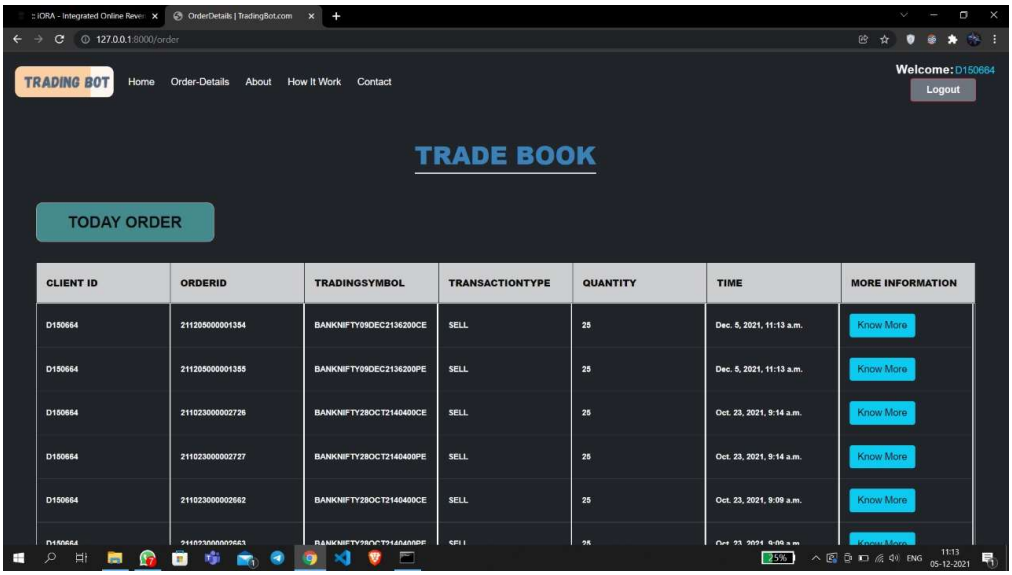


Figure 4.6: Tradebook page

Above figure 4.6 shows Trade book of the system. In table format there is a field like client id, order id, trading symbol, transaction type like buy or sell, then quantity (number of share) then next field is time when trade is take placed and last field is More information to know more information about order.

SQLite db

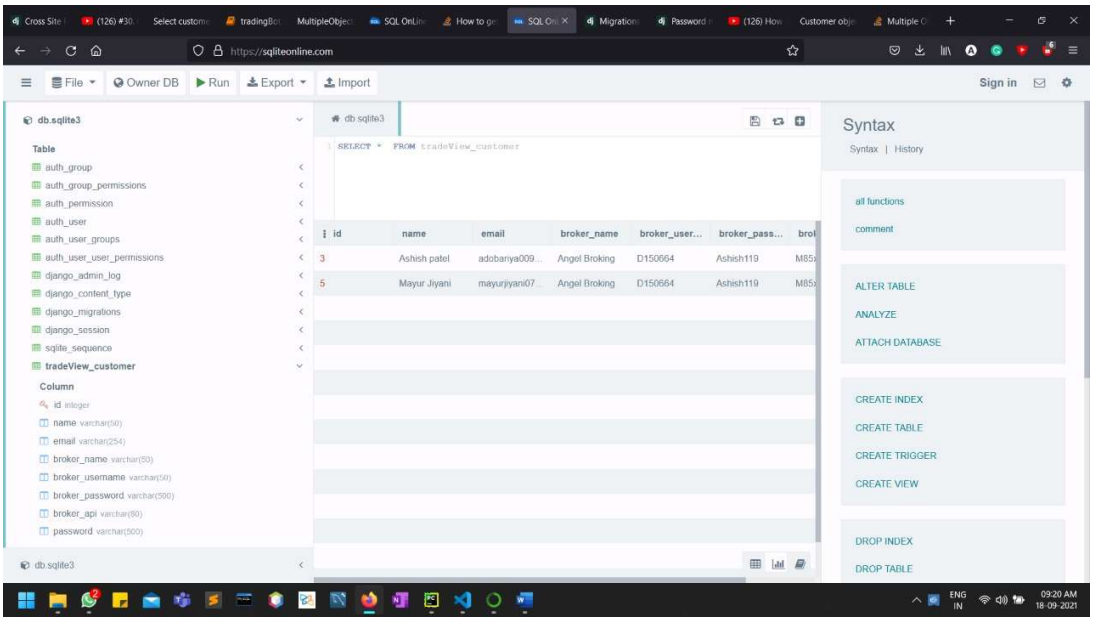


Figure 4.7: SQLite db view

Above figure 4.7 show the view of SQLite db. In Our system we have used SQLite db. sqlite3 provides a SQL-like interface to read, query, and write SQL databases from Python. Here all the information of user is stored and Trading bot access the information from here. In db all the table with necessary field is located.

Testing:

Backtesting profit by monthly

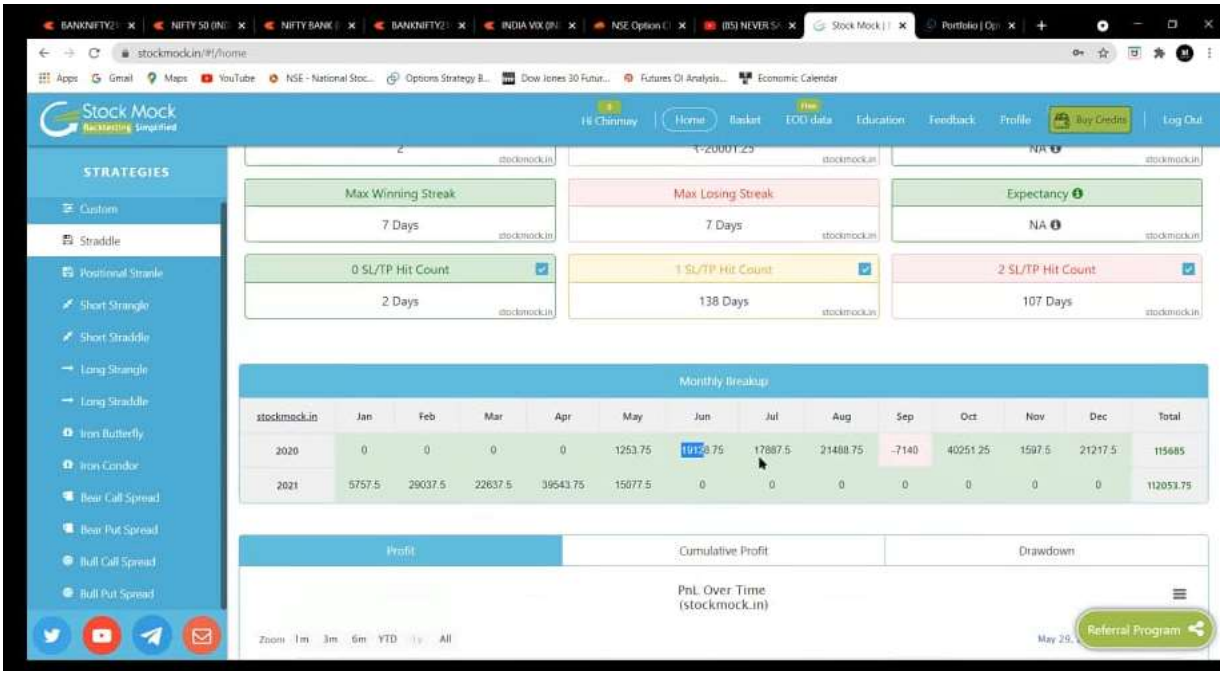


Figure 4.8: Backtesting profit - monthly

Above figure 4.8 shows, by this strategy how much profit was generated in a particular month for the last year. From month January to December total profit is 112053.75 for year 2021.

Backtesting profit of one year

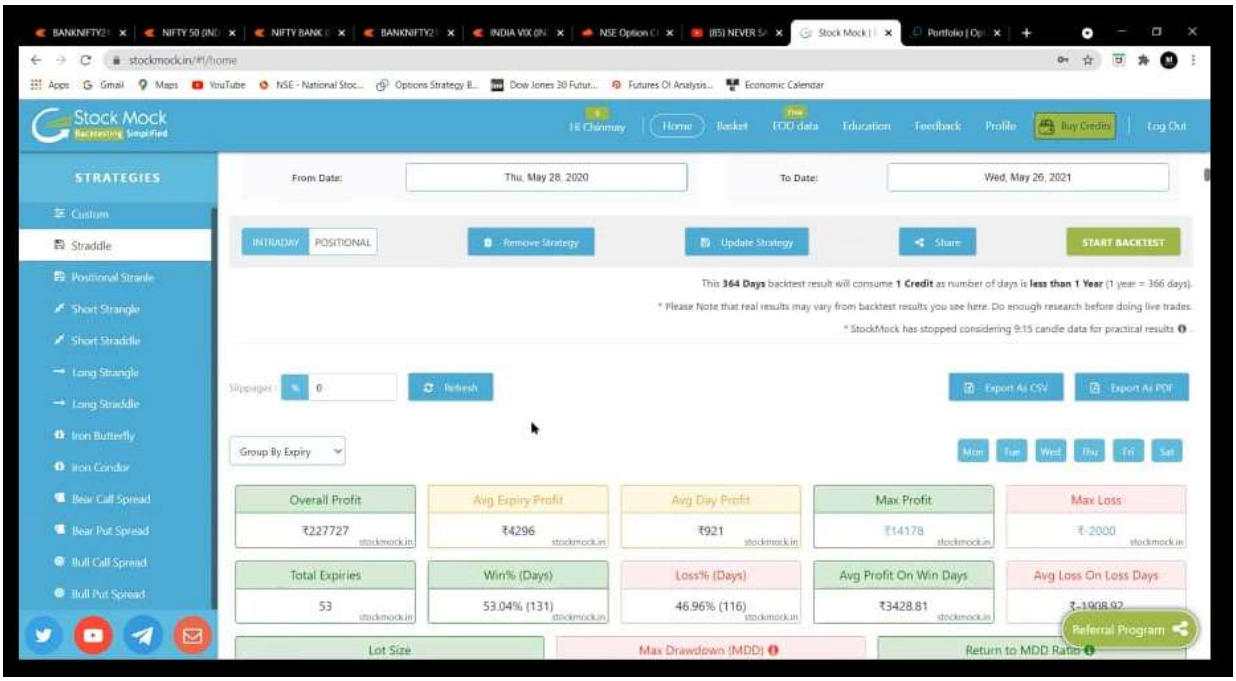


Figure 4.9: Backtesting profit - one year

Above figure 4.9 shows the last year's backtesting profit by this strategy. The total profit by applying this strategy is 227727 rs.

Chapter 5: Conclusion & Future work

Algorithmic trading system architectures are complicated because of the strict quality requirements of the system. Also, the wide range of regulatory and compliance requirements which manage automated trading makes the design of this software harder. Because of these complexities, careful attention should be paid to the design and implementation of the system architecture. This software requirements specification document has been created through the help of various researches. In this document, the general information about product description, data elements that the product deals with, specific requirements like product's interfaces and the functions that will be implemented are provided. However, some specifications are prone to be changed in the future.

Chapter 6: References

The resources listed below are references used in requirement analysis: IEEE Standard Documents:

- [1] IEEE. IEEE STD 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.
- [2] StarUML 5.0 User Guide. (2005). Retrieved from http://staruml.sourceforge.net/docs/user-guide(en)/toc.html
- [3] Hull, J. (2009). Options, futures, and other derivatives; seventh edition (7th edition). Upper Saddle River, N.J.: Prentice Hall.
- [4] Using Genetic Algorithms To Forecast Financial Markets. (n.d.). Retrieved November 30, 2014, from http://www.investopedia.com/articles/financialtheory/11/using-genetic-algorithms-forecast-financial-markets.asp
- [5] Binary option. (2014, November 29). Retrieved November 30, 2014, from http://en.wikipedia.org/wiki/Binary_option
- [6] RELEASE: Fraudadv_binaryoptions. (n.d.). Retrieved November 30, 2014, from http://www.cftc.gov/PressRoom/PressReleases/fraudadv_binaryoptions