Share-Market prediction system

Software Requirements Specification Document

Submitted by:

Hersh Kenkre(B043)
Sharvil Kotian(B045)
Hrishil Maliwal(B049)
Madiha Mansoori(B051)

Under Software Engineering Course for partial fulfillment of B. Tech Program

Under Guidance of

Dr. Dhirendra S Mishra
Professor, Computer Engineering
&
Manisha Tiwari
Asst. Professor, Computer Engineering

At

SVKM's NMIMS
Mukesh Patel School of Technology Management and Engineering
Vile Parle West –Mumbai-56
A.Y. 2020-2021

Version: (1) **Date: (04/05/2021)**

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to build an online system to predict share market trends. Share Market is an untidy place for predicting since there are no significant rules to estimate or predict the price of shares in the share market. Many methods like technical analysis, fundamental analysis, time series analysis and statistical analysis, etc. are all used to attempt to predict the price in the share market but none of these methods are proved as a consistently acceptable prediction tool. This software is aimed at Indian users as the prediction is done on the listed companies of the Bombay stock exchange Ltd.

1.2 Scope

The purpose of the stock market prediction system is to ease stock trend prediction and to create a convenient and easy-to-use application for investors, trying to look for trend analysis. The system is based on a long, short term memory ANN and has functions which allow you to view past trends and future predictive trends. The software consists of pre-trained models for top 30 companies with a future scope of improvement with models trained for more companies. Above all, we hope to provide a comfortable user experience with maximum convenience.

1.3 Definitions, Acronyms, and Abbreviations.

ANN: Artificial Neural Network
API: Application Programming Interface
CSS: Cascading Style Sheet
YF: Yahoo Finance
LSTM: Long Short-Term Memory
HTML: Hyper Text Markup Language
BSE: Bombay stock exchange
NSE: National stock exchange
SQL: Structured Query Language
DB: Database

OS: Operating System

Ticker: Used to identify stock from Yahoo finance

Df: Data frame

1.4 References

References used to make the software –

1. https://pypi.org/project/yfinance/
2. https://flask-doc.readthedocs.io/en/latest/
3. https://www.tensorflow.org/guide

1.5 Overview

Chapter 2 contains a brief description about our product. It contains information about everything right from the features, functionalities, interface, dependencies, and constraints of the website to the facts known and assumptions made while developing the website. This section is useful for the users/future customers.

Chapter 3 contains a more specific and in-depth study of the requirements and caters towards the product development team. The logical description section contains various diagrams like Dataflow Diagram, State Transition diagram, Use Case diagram etc. The System Attributes section focuses on Reliability, Availability, Security and Maintainability of the website. The SRS is documented keeping in mind the needs of both the users and the development team's information requirements.

2. The Overall Description

Predicting how the stock market will perform is one of the most difficult things to do. There are so many factors involved in the prediction – **physical factors vs. psychological, rational and irrational behavior**, etc. All these aspects combine to make share prices volatile and very difficult to predict with a high degree of accuracy. Using features like **the latest announcements about an organization, their quarterly revenue results**, etc., machine learning techniques have the potential to unearth patterns and insights we didn't see before, and these can be used to make unerringly accurate predictions.

The software will work with **historical data about stock prices of a publicly listed company** which is collected using the yahoo finance API. It will implement a mix of machine learning algorithms to predict the future stock price of a company.

The software aims at providing an accurate prediction for trends in the company's stock price for 30 days into the future by basing its prediction on the previous 500 days data.

The objective of the software is -

- To generate the pattern from a large set of data of the stock market for prediction of BSE.
- To predict an approximate change in trend of share price.
- To provide analysis for users through web application.

2.1 Product Perspective

The product is not completely independent and is dependent on the functioning of the finance API to source its data thereby making it not fully self-contained.

The product is an open to use prediction web app which currently allows support for the top 30 NIFTY listed companies. It allows the user to select his company and view a predicted trend for the company to analyze and assist him in investments.

Main features of Investo -

- All platform support: Since Investo is a web-based application, it is available for use on any browser.

- User Account: The system allows the user to create their accounts in the system to enable usage of the prediction feature.
- Number of users being supported by the system: Though the number is precisely not mentioned, the system is able to support a large number of online users at a time.
- Viewing Analysis: users are presented with the option of just viewing the past performance of a company in a graphically represented manner.
- Prediction: This is the key feature of the product and shows the future prediction in the form of a graph. This prediction is based on an LSTM model which is retrained and stored in the backend.

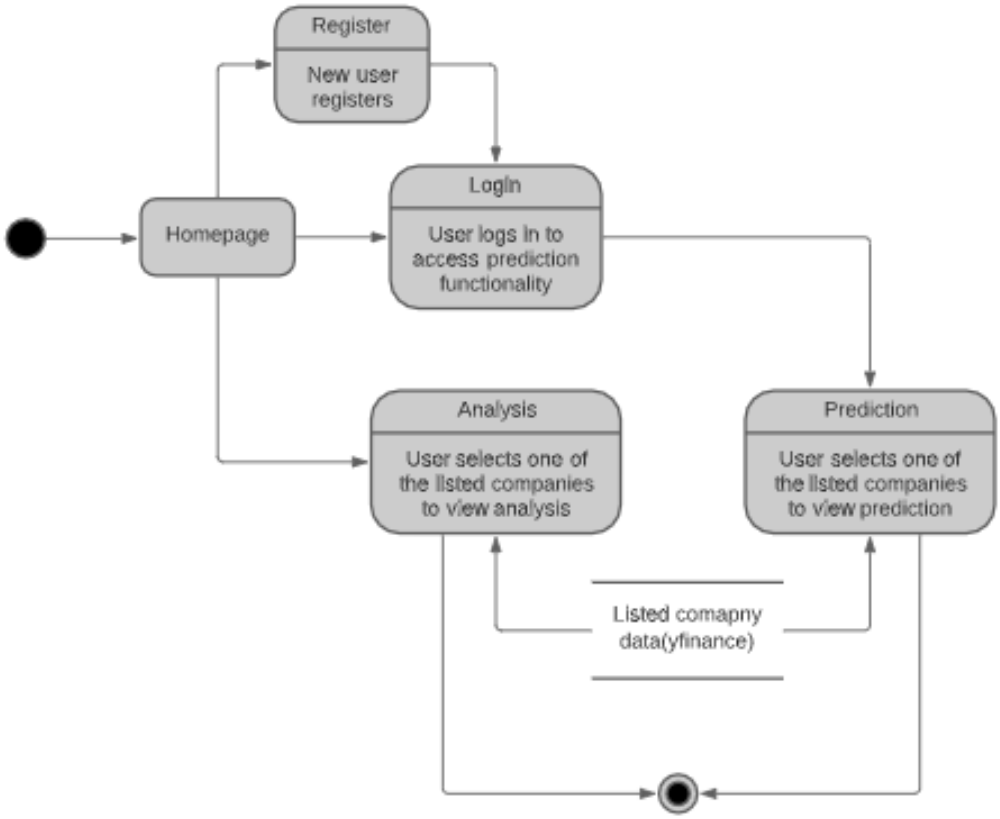


Fig. 2.1 Product perspective.

2.1.1 System Interfaces

The system interface required -

- Flask is a web framework we used for developing the database, functionalities and frontend of the Web application.

- To implement the user login and registration authentication.
- To take user input and display output images of graphs.

2.1.2 Interfaces

The interfaces for the software are -

- The user will interact with the system with a GUI built on HTML, CSS and Flask which will provide an interactive interface to the user.
- There are no other special requirements required for interacting with the systems other than what are provided on screen to the user.

2.1.3 Hardware Interfaces

The software is a web-app and has no specific hardware interface requirements other than a stable internet connection and browser.

2.1.4 Software Interfaces

2.1.4.1. Flask:

The system uses flask to integrate all the html templates as well as the database in the web framework. Flask is further used to integrate all the prediction models into the software as well.

2.1.4.2. SQLite:

The system uses SQLite database to store its user login data.

2.1.4.3. HTML5:

The system must use HTML5 as its main component for website construction.

2.1.4.4. CSS3:

The system must use CSS3 as its main component for website designing.

2.1.4.5. Python3

The backend is coded using python3 and its various libraries.

2.1.4.6. TensorFlow and keras:

The models used for prediction are coded using TensorFlow and keras libraries.

2.1.5 Communications Interfaces

Since the project uses web services there is no need for a communication interface.

2.1.6 Memory Constraints

Since the software is a Web application there are no memory constraints.

2.1.7 Operations

The operations are -

- The product mainly only operates in the user mode and an admin mode is not currently available.
- The models are pre trained and exist in the backend, these models require to be retrained every 3 months or post any major change in the stock market, whichever occurs earlier. During this period of retraining, no major interruption to regular functioning of the software will occur as the newly retrained models will be deployed causing an interruption of at most 1 hour every 3 months.
- The data that is used for every prediction is scraped off the finance website and errors in the API can cause major disruption to the functioning of our software.

2.1.8 Site Adaptation Requirements

There are no site adaptations required for the product.

2.2 Product Functions

The product is available for functioning by two types of users namely, registered and unregistered users.

The two functions of the module are analysis and prediction, an unregistered user is only allowed to access the analysis mode while a registered user is allowed to view the prediction as well.

Some other functionalities are –

1. **Registration:** New users are provided with the option of registering to access complete product functionalities
2. **Login:** Registered users need to log in for every session
3. **Analysis:** The product provides a function to view the past statistical analysis in a graphical manner for convenience of users, this information is provided from the company's IPO to the last recorded functional day.
4. **Prediction:** A registered user has access to the prediction functionality of the product and is given a list of 30 companies to choose from. These models are trained on data from its IPO to up to 3 months ago and make predictions on the basis of the previous 500 days of data.
5. **Blog:** Redirects to external finance blogs.
6. **About:** About page has information about the project developers.

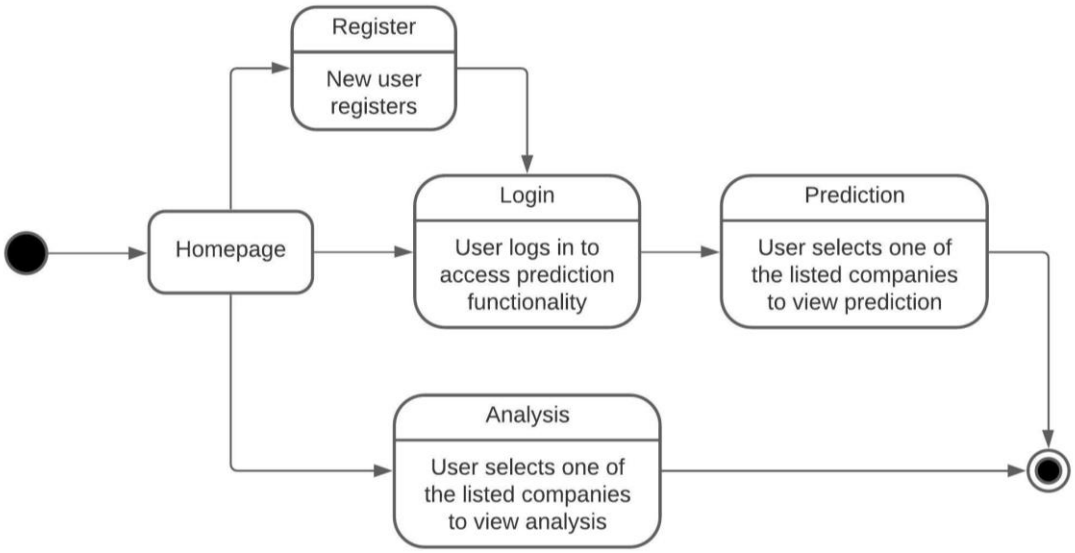


Fig. 2.2 Functional diagram.

2.3 User Characteristics

The targeted user base is -

- Experienced traders
- Novice traders
- Individuals wishing to learn stock market trends.

2.4 Constraints

Constraints that will limit the functionality of the web-app are -

- Usage of an outdated/unsupported browser.
- Failure to access data due to API error.
- Slow internet of the user which slows the web-app.
- The current version of the product is a local version so the user should have the supportive software installed, failure of which will lead to the product not running.

2.5 Assumptions and Dependencies

Some assumptions and dependencies that are taken into consideration are -

- The current SRS states that the version of Languages used are up to date and are compatible with the browsers. There might be a chance that in the future, there might be some methods or functions which might lose their support in the browser, hence might affect the design of the project, hence the SRS will be updated if this happens.
- This along with the dependency on the API, failure or depreciation of the API will lead to changes in the SRS.
- It is assumed that the user will have supportive software versions installed.
- The API returns values in an efficient time range.

2.6 Apportioning of Requirements.

Portfolio management:

The future version of the software would add a functionality for registered users to manage their portfolio and view trends for all their investments simultaneously.

Recommending investment opportunities:

Recommending hot stocks on the basis of the current risk status of the user.

Update profile feature:

The option to modify the profile is not available in this version.

Email Verification:

Email verification is not implemented in the current version.

3. Specific Requirements

3.1 External Interfaces

3.1.1 User Interfaces

3.1.1.1. Homepage -

- The homepage will have a Navbar containing links to the different sections of the webpage: Login, registration and about us pages.
- This page will contain the search bar where the user can select the stock, he/she wants to view.
- This particular homepage can be accessed by all users without logging in.

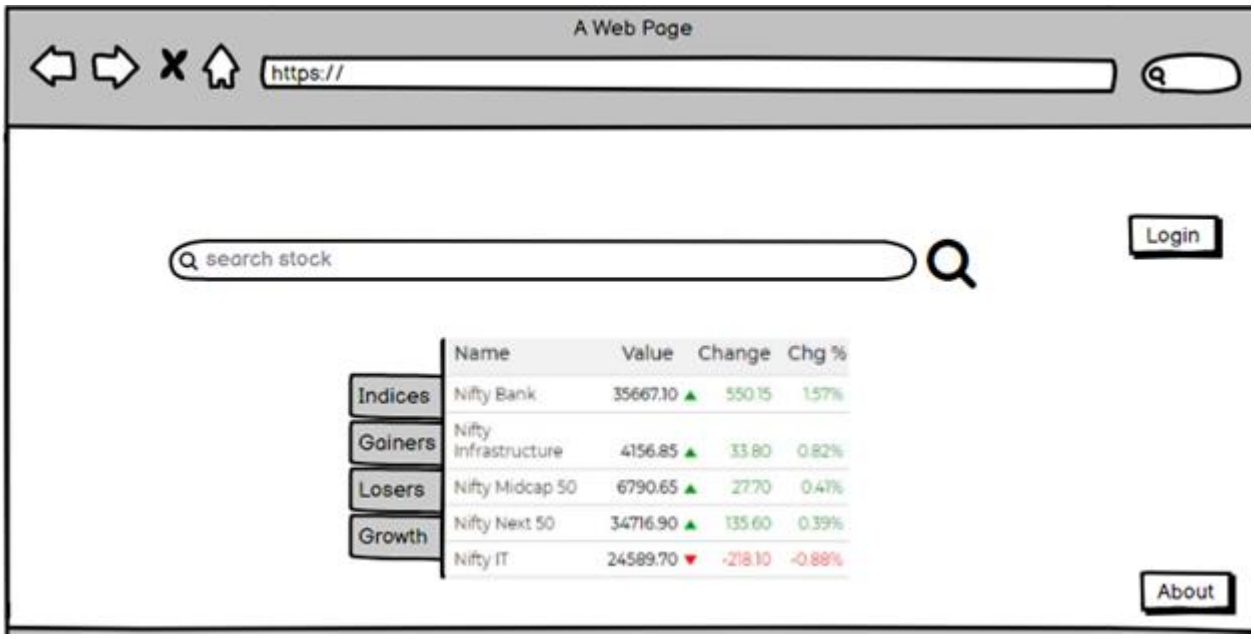


Fig. 3.1 The homepage interface design.

3.1.1.2. Prediction page: -

- This page will allow the user to be able to see the chart of the stock he/she selected.
- To get access to the prediction graph of the stock the user will have to login.
- It will also contain the navigation bar, and which is generic to all our interfaces and the links for registration, login and home pages.

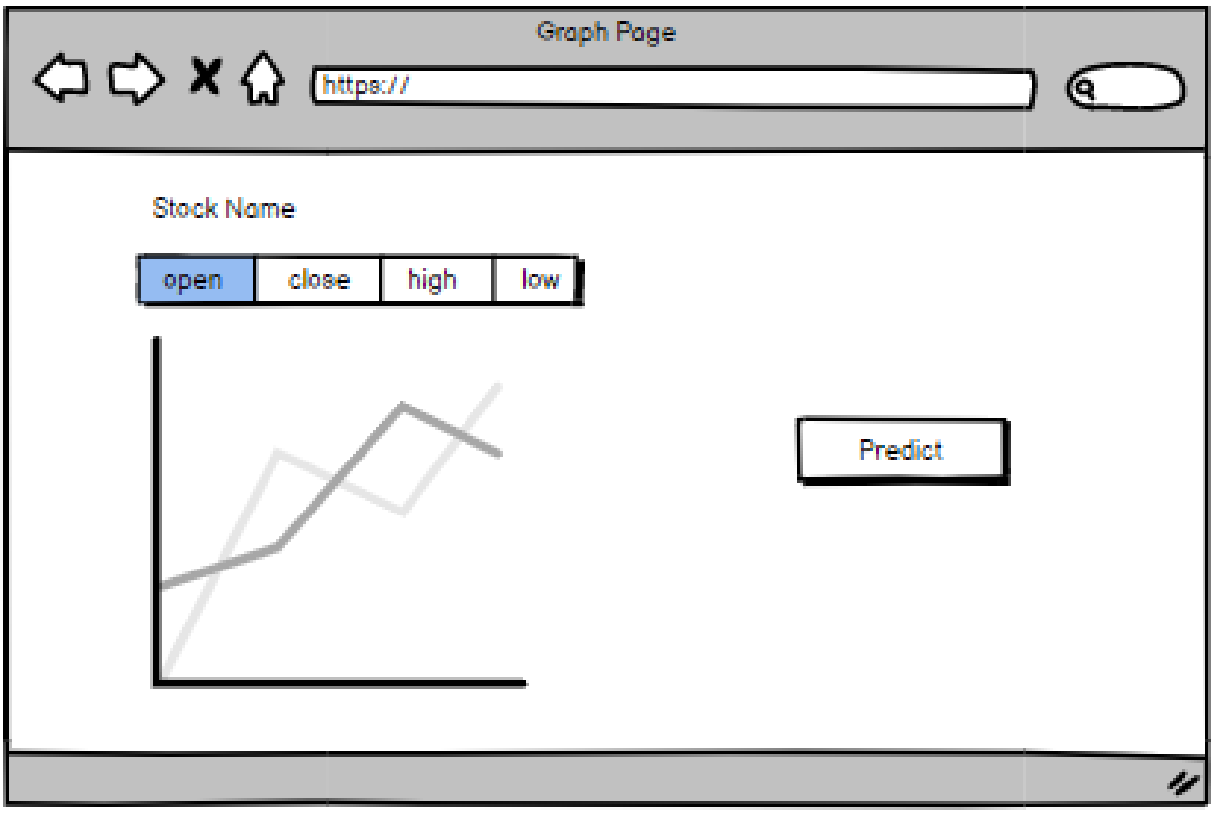


Fig 3.2. The prediction page interface design.

3.1.1.3. User Register:

- This page will enable the users to register by entering their details into a form, which should contain the fields of name, password, phone number and email which then would get sent into the database.
- The user cannot use an email Id, Username or phone number that has already been used.
- A link to the login page must also be given to the user who already has an account at the registration page.

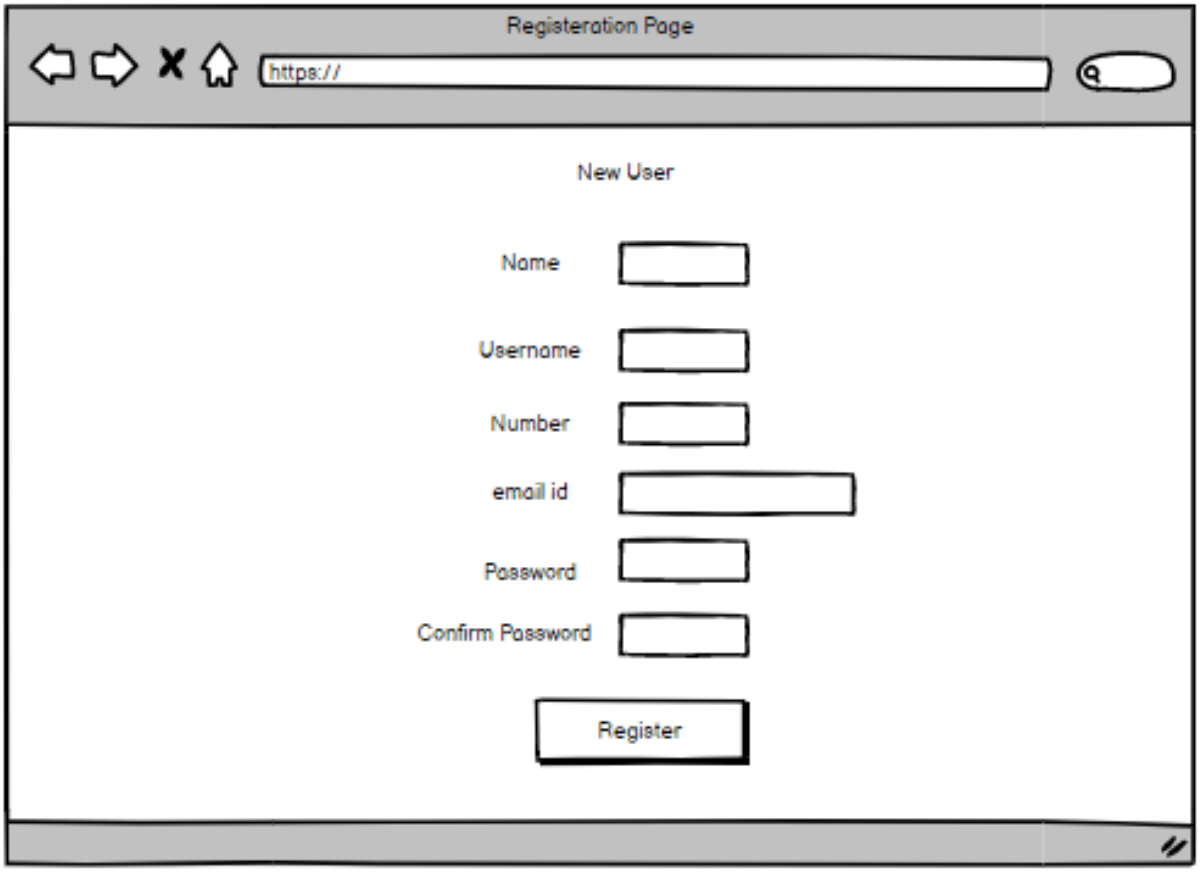


Fig. 3.3 The registration page interface design.

3.1.1.4. User Login:

- The user should be able to see two fields for input namely Email Id and password.
- The user will be able to select the “Remember me” option to keep them self-logged in.
- A link to the registration page must also be given to the user who does not have an account at the login page.
- It will also contain the navigation bar and the links for registration, about and home pages which is generic to all our interfaces.

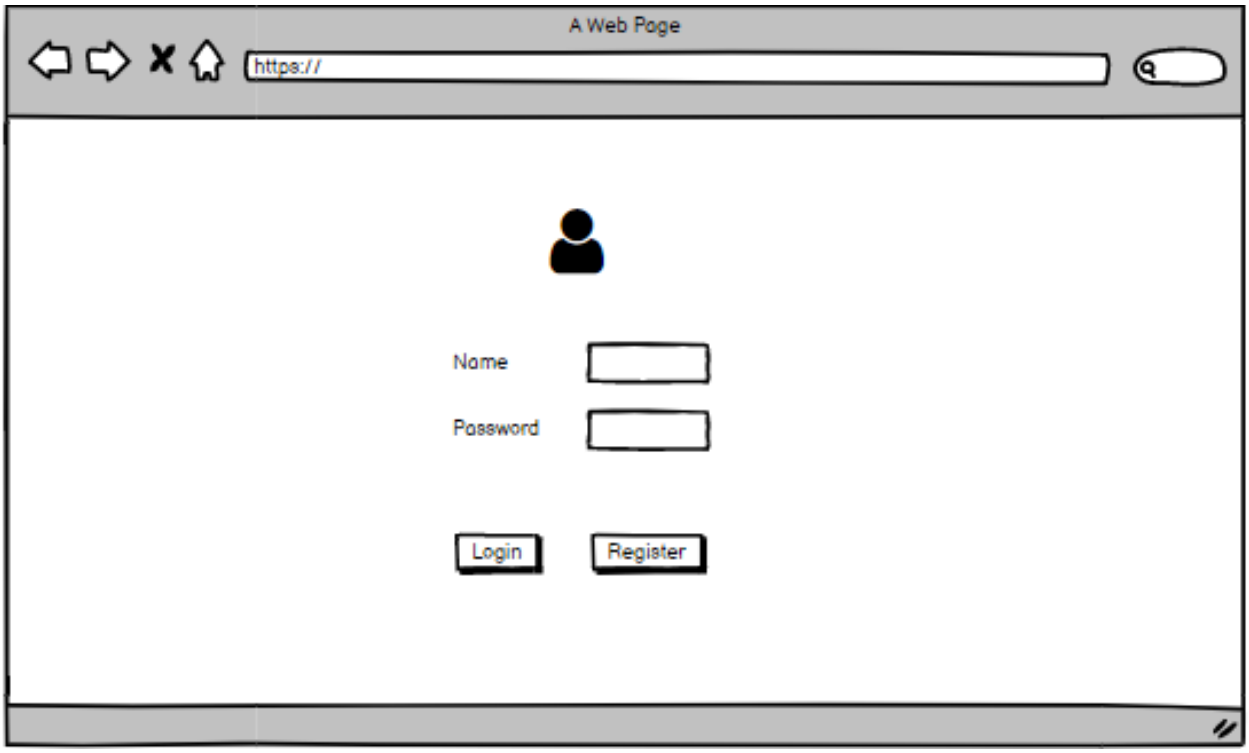


Fig. 3.4 The user login page interface design.

3.1.1.5. About us:

- This page contains the information about the members of the group.
- It also contains the navigation bar, and which is generic to all our user interfaces and the links for Login, registration and home pages.

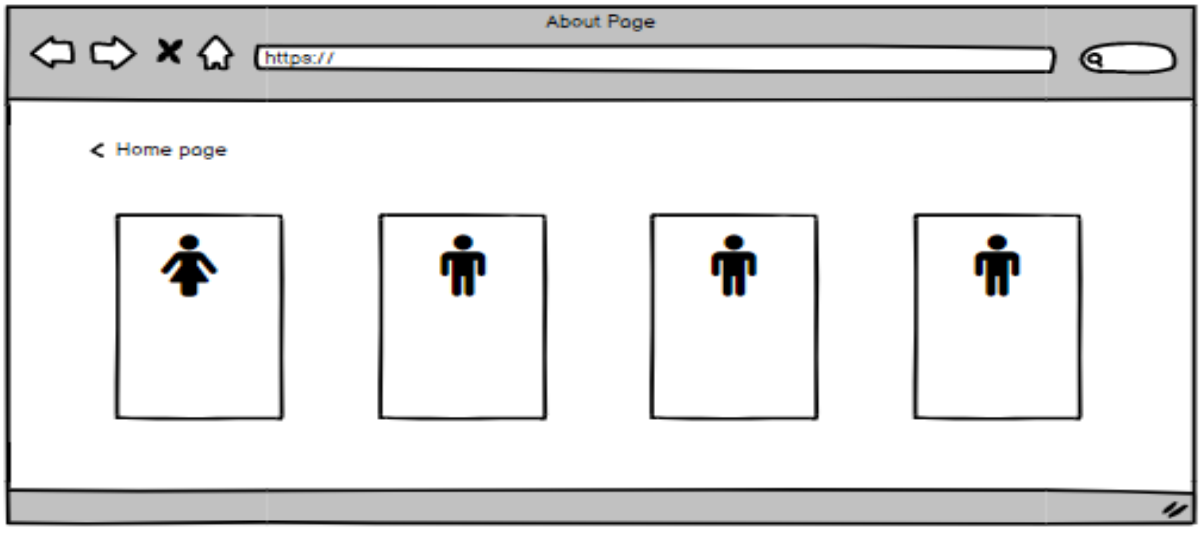


Fig. 3.5 The about us page interface design.

3.1.2 Hardware Interfaces

The hardware interfaces are -

1. A stable internet connection.
2. A power supply to power the device.
3. A network Interface Card to connect the device to the internet.

3.1.3 Software Interfaces

The only software interface is a compatible browser.

3.2 Functional Requirements

3.2.1 Functions:

3.2.1.1 Model:

- For every prediction, the data is fetched up to the last recorded day.
- The charts and prediction of the companies would be done only on the latest data stock market data.
- The data is used on the trained LSTM model to base a prediction.

3.2.1.2 User Registration:

- This page should enable the users to register by entering their details into a form, which should contain the fields of name, password, phone number and email which then would get sent into the database.
- On clicking the register button, the user credentials get added to the database.
- The user will be redirected to the login page after successful registration.
- Popup will appear on unsuccessful registration. (Using existing username/Email Id)

3.2.1.3 User Login:

- The entered email id and password will be compared to the database and validated on clicking the login button.
- The user will be redirected to the Home page after successful login.
- Popup will appear on unsuccessful login. (Wrong Email Id and password)

3.2.2 Process description

3.2.2.1 Login/Register

3.2.2.1.1 Input data entities - username, email, password

3.2.2.1.2 Algorithm or formula of process

1) Accept details

2) Check if fields are valid i.e., Validation of username, password, email

3) If fields are valid: check database for existing username

4) If username already exists in database: show error saying username already exists

5) If user doesn't exist in database: create a new entry in database with details

6) Redirect to login and let user enter login details

7) Authenticate details

3.2.2.1.3 Affected data entities – none

3.2.2.2 Authenticate

3.2.2.2.1 Input data entities - username, password

3.2.2.2.2 Algorithm or formula of process -

1) Check database for existing user

2) If user exists, and password matches- login success

3) Else reject login

3.2.2.3 View Analysis

3.2.2.3.1 Input data entities -Listed company name

3.2.2.3.2 Algorithm or formula of process

1) Accept details

2) Pass company name to the model

3) Model gets related data from yfinance data

4) Preprocess data

5) Plot data using Matplotlib and save image

6) The graph is displayed on the analysis HTML page.

3.2.2.4 View Prediction

3.2.2.4.1 Input data entities -Listed company name

3.2.2.4.2 Algorithm or formula of process

1) Accept details

2) Pass company name to the model

3) Model gets related data from yfinance data

4) Preprocess data

5) Get prediction for the next 30 days from LSTM model based on 500 days of data

6) Plot result using Matplotlib and save image

7) The graph is displayed on the analysis HTML page.

3.2.3 Data Construct Specifications

3.2.3.1 persons Table

3.2.3.1.1 Record type - contains all user details.

3.2.3.1.2 Constituent fields- username, email, password, fname, lname.

3.2.4 Data dictionary

3.2.4.1 Username

3.2.4.1.1 Name - username

3.2.4.1.2 Representation - Username

3.2.4.1.3 Units/Format - string/character

3.2.4.1.4 Precision/Accuracy - none

3.2.4.1.5 Range - A-Z, a-z, 1-9, @! $_

3.2.4.2 Email

3.2.4.2.1 Name- email

3.2.4.2.2 Representation- User’s email ID

3.2.4.2.3 Units/Format - email

3.2.4.2.4 Precision/Accuracy -100% follows email format, i.e., example@example.com

3.2.4.2.5 Range- email can be of any length.

3.2.4.3 Phone Number

3.2.4.3.1 Name - phno

3.2.4.3.2 Representation - Users Phone number

3.2.4.3.3 Units/Format - number

3.2.4.3.4 Precision/Accuracy - 100% follows 10-digit number format.

3.2.4.3.5 Range - [0-9] *10

3.2.4.4 Password

3.2.4.4.1 Name - password

3.2.4.4.2 Representation - Password

3.2.4.4.3 Units/Format - password

3.2.4.4.4 Precision/Accuracy - none

3.2.4.4.5 Range - A-Z, a-z, 1-9, @! $_

3.2.5 Data flow diagrams

The software can be represented using three data flow diagrams.

Level 0 diagram -

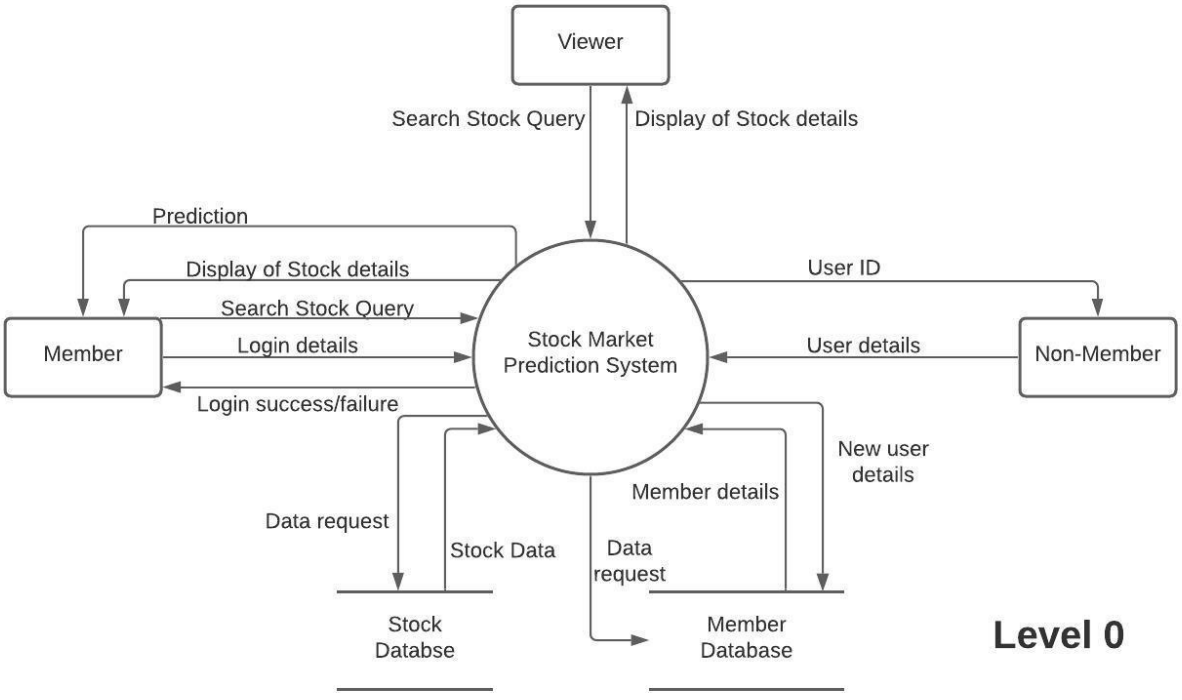


Fig. 3.6 Level 0 DFD diagram.

Level 1 diagram -

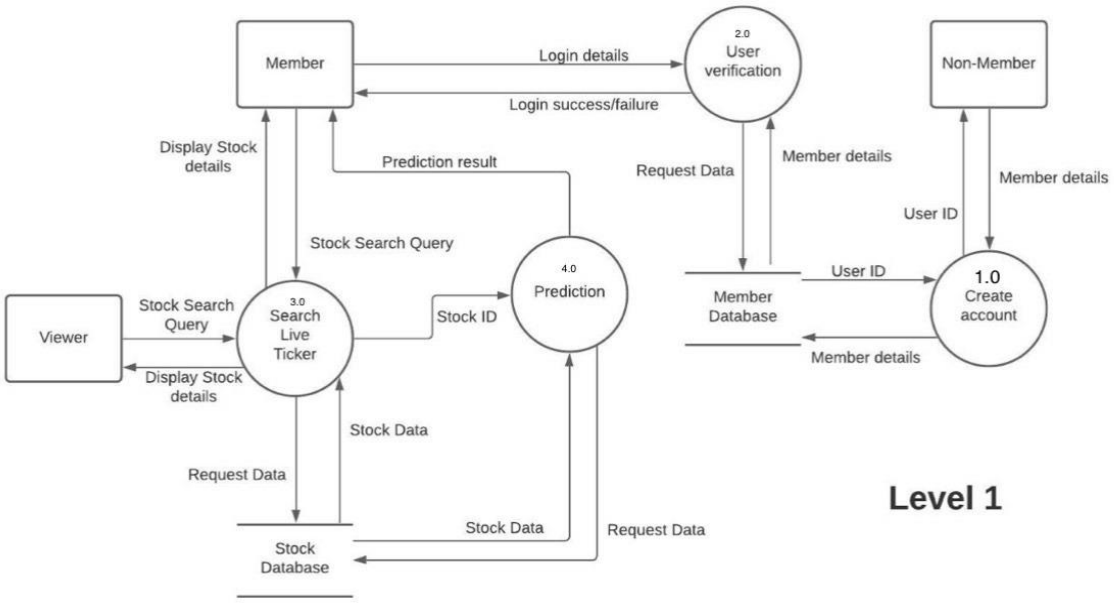


Fig. 3.7 Level 1 DFD diagram.

Level 2 diagram -

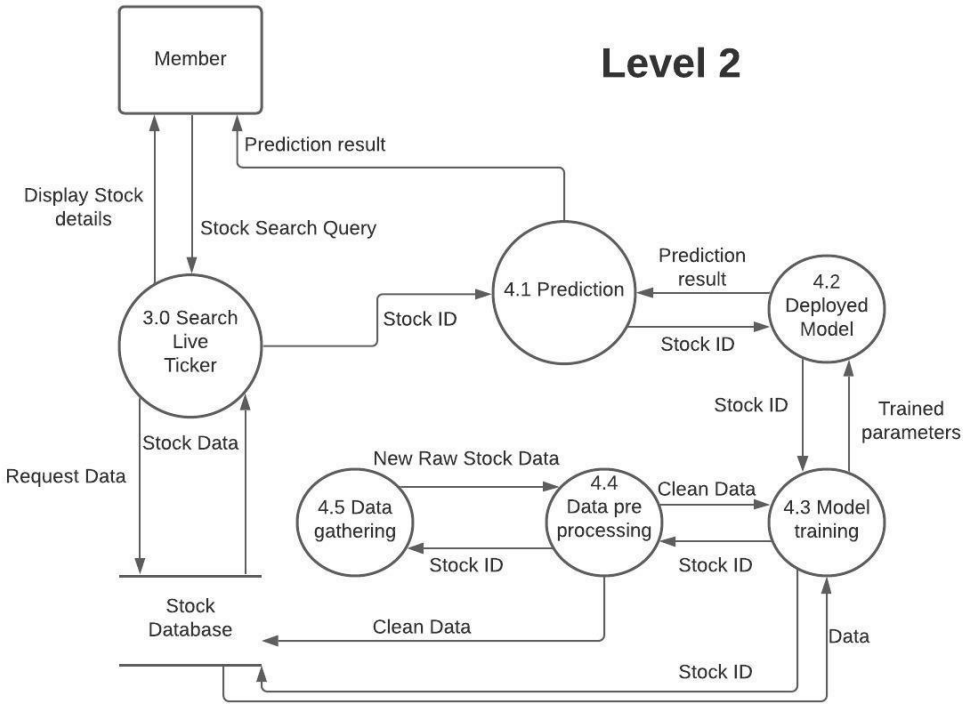


Fig. 3.8 Level 2 DFD diagram.

3.2.6 State transition diagrams

There are mainly two state transition diagrams that are associated with the software. First is related to the login and verification. Second is related to the prediction model.

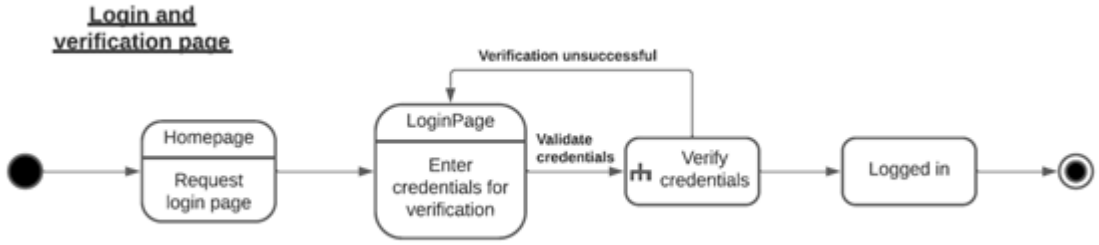


Fig. 3.9 Login page state transition diagram.

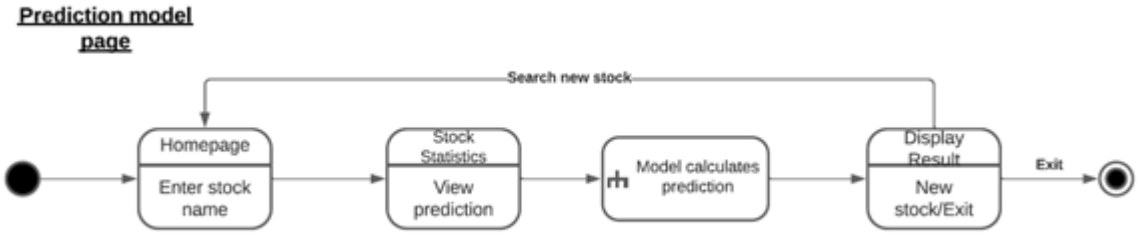


Fig. 3.10 Prediction model state transition diagram.

3.2.7 Use case diagram

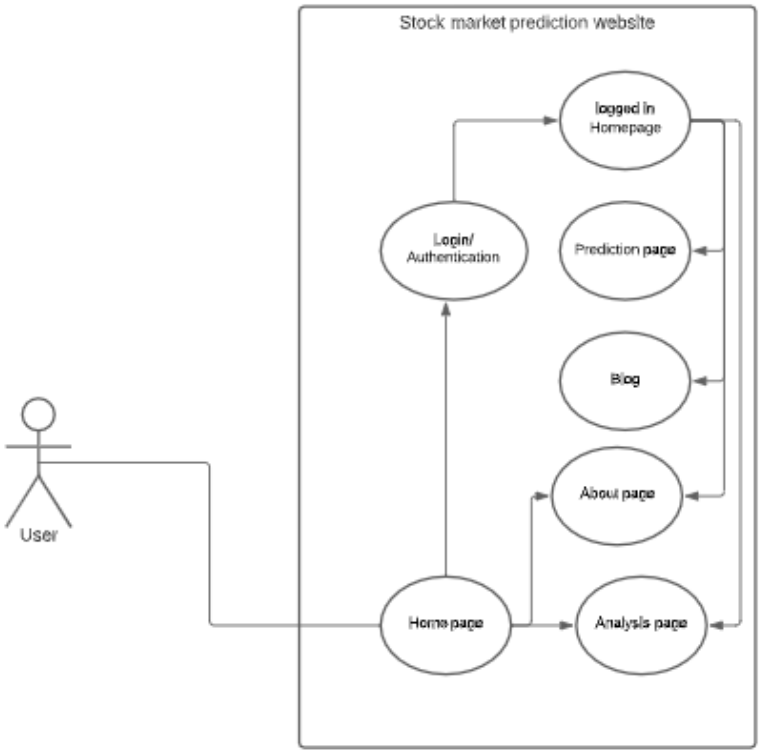


Fig. 3.11 Use case diagram

3.3 Performance Requirements

The performance requirements are -

- The responsiveness of the website must be fast, i.e., there must be sufficient number of servers to handle many users.
- The model should return the result i.e., the graph under a minute.
- Safety of the database should be provided, since it contains the user's personal information.
- Reliability of the database is a major requirement, in order to prevent any loss of data, i.e., personal information, password, number etc.

3.4 Logical Database Requirements

The database is only used to store the Login and registration credentials of the users of the application.

3.5 Design Constraints

Design constraints of the software are -

- User cannot change their details after registering.
- Same email ID, phone number and username cannot be used twice by any user.
- The software can only predict prices of NIFTY 30 companies.
- The current website is designed keeping in mind a regular sized screen and might not function on differently sized screens.
- Due to GPU limitations, the number of trained models that were deployed are only 30.

3.5.1 Standards Compliance

The project is based on yahoo finance API release and is not compatible with fix-yahoo-finance.

3.6 Software System Attributes

3.6.1 Reliability

To establish a reliable software system at the time of delivery we require a stable internet connection and browser. With respect to the MTBF requirements, the model requires training data of approximately 500 days, meaning the servers will be down once every 1.5 years. Making the MTBF 8760 hours* (MTBF calculation = number of operational hours/number of hours of failure).

** This MTBF is calculated without considering other reasons for hours of failure.*

3.6.2 Availability

The system will stay available without any disruption if the following requirements are satisfied -

- Uninterrupted internet connection

- Compatible browser support
- Compatible versions of flask, python and TensorFlow

3.6.3 Security

The following considerations are made regarding the security aspect of the software -

- In order to use sessions in flask, we set the secret key in our application settings. Secret key is a random key used to encrypt the cookies and send them to the browser. The secret key is needed to keep the client-side sessions secure.

- The primary hashing method used to store passwords in the database in a secure way was the implementation of the flask extension - Bycrpt.

- The database is stored only on one machine to maintain data integrity and to secure privacy.

3.6.4 Maintainability

Since the software is a web application low level of maintenance is required. If at all any maintenance is required, the software is divided in different modules and files making the maintenance process faster and easier. The model will be re-trained with the latest stock data from time to time to give accurate results and predictions.

3.6.5 Portability

The portability requirements of the host side are -

- A latest Python interpreter (version 3.7 +)
- Python libraries/extensions: TensorFlow, Flask, pandas, matplotlib, NumPy and sklearn.

From client side -

- The languages used for web design are HTML and CSS, which are the basics and are supported by all browsers.
- The website can be used in every operating system that has an internet browser available.

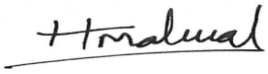
ID	Characteristic	H/M/L	1	2	3	4	5	6	7	8	9	10	11	12
1	Correctness	**H**												
2	Efficiency	**H**												
3	Flexibility	**H**												
4	Integrity/Security	**M**												
5	Interoperability	**M**												
6	Maintainability	**H**												
7	Portability	**H**												
8	Reliability	**H**												
9	Reusability	**H**												
10	Testability	**M**												
11	Usability	**H**												
12	Availability	**H**												

Table 3.1 Software attributes.

4. Change Management Process

Changes can be identified, logged, monitored and updated using the GitHub repository and its forks. GitHub is a hosting for software development and version control using Git. Customer's demands can be satisfied in a fork of the main repository and the team can decide by voting whether to implement (or pull) that in the main repository based on how the changes are affecting the functionality and if the changes are adaptable.

5. Document Approvals

			
Hersh Kenkre)	Sharvil Kotian	Hrishil Maliwal	Madiha Mansoori
B043	B045	B049	B051

Dr. Dhirendra S Mishra Professor, Computer Engineering	Manisha Tiwari Assistant Professor, Computer Engineering