AI Investment Advisory

Software Requirement Specification (SRS)

(Provide GitHub repository URL here)

BY

Team Lead: ***Syed Hussain Imam/19773***

Member(s):

Ahmed Abdul Ghafoor 19735

Huzaifa Farooq 19766

Mohammad Razi Moosa 18663

PROJECTS COMMITTEE (PC)

SUPERVISOR: ***Tahir Syed***

CO-SUPERVISOR: ***-***

Member(s):

SUBMITTED TO

DrTahir Syed

PROJECTS Manager – FYP

ON

DATE (27/01/2023)

Table of Contents

Software Requirement Specification (SRS)

1. Problem Statement

Stock market analysis and prediction has been a very important area of study in the realm of finance. Current investors, primarily individual investors rely heavily on publicly available news sources and sentiment. Making sound investment decisions proves to be an extremely difficult task due the amount of information needed to consistently beat the market. The current widely used practices for stock market prediction are:

- Historical price information
- Sentiment analysis of news and general public opinion
- Macro and Micro economic factors such as
    - Interest rates
    - Exchange rates
    - Industry specific information:
        - Growth rates
        - Consumer prices
        - Income statements
        - Dividend yields

Efficient market hypothesis states that all available information is accounted for in the current stock price, Hence it is impossible to predict future prices without any new information. This is a debated issue and stock markets around the world provide significant evidence that markets are not fully efficient.

To take advantage of this we will be creating a stock advisory platform that will take advantage of artificial intelligence techniques. We will take the technical analysis approach where we take into account psychological factors that affect future prices and trends. The crowd psychology of collective panic, fear, pessimism, confidence, excessive optimism and greed.

Alot of popular techniques aim to quantity is factor namely:

- Moving Averages (MA)
- Auto Regressive Integrated Moving Average (ARIMA)
- Artificial Intelligence techniques

Over the course of this project we will be studying past researches of how accurately different approaches have attempted to predict stock price and their limitations. Alongside building the advisory platform we will be reviewing these approaches.

The Advisory platform will be a publicly available web application in the beginning but will move to a subscription model for individual and institutional investors in the future. The platform will provide precise stock information of all stocks in the KSE-100 regular index, the app will visualize the predictions of our model and provide advice for investments to users. The users will be able to filter, select, and follow the stocks of their choice.

The level of individual investor engagement is very low in the Pakistan Stock Exchange (PSX) compared to other countries. This platform will also try to promote individuals to become more active in the PSX by providing important information and advice that is difficult for new users to understand or find in currently available data portals.

2. System Requirement

2.1. List of Functional Requirements

Functional Requirement No.	Functional Requirement Description
FR01	System can authenticate users through user accounts (Username/Email + Password)
FR02	System can process new user registration
FR03	System can display stock price data for all stocks in KSE-100 index
FR04	System allows users to create a portfolio
FR05	System allows users to add/remove stocks from their portfolio
FR06	System shows stock price prediction for all stocks in KSE-100 index
FR07	System shows current and predicts future portfolio values
FR08	System should retrieve daily stock price data from data sources

2.2. List of Non-functional Requirements

1.1.1 Start-Up Time

Users will be expected to use this application on the go thus the application should load and be usable within 3 seconds.

1.1.2 Filter Response Time

The application should display values and models according to the filtered search within 5 seconds after the user triggers the filter operation to maintain a flowing user experience.

1.1.3 Confirmation Time

The user should be notified of successful order placement within 5 seconds of triggering the operation of adding the stock to their portfolio .

1.1.4 Tracking Stocks

The tracking details of the user stocks should be updated every specific interval of time.

1.2 Safety Requirements

Safety requirements consist of the after effects of any damage or possible loss that can culminate because of the usage of the product itself. The software would have the ability to work normally in case of any discrepancy that may be caused by any user command.

1.2.1 Server – Receiver

Data stored in the server will be protected from power loss. Data in transit (Server -> Receiver and vice versa) can be lost, in such a situation, the data will be sent again.

1.2.2 Data Privacy

Correct protocols to ensure encryption should be followed.

1.2.3 Account Sharing

Account sharing will be discouraged. Account sharing can lead to outcomes unintended by the owner of the account.

1.3 Security Requirements

This requirement focuses on preventing any kind of data loss and provides protection against harmful viruses and unauthorized access.

1.3.1 Authentication

To aid the system to identify the user and to limit uninvited malicious user activity. Email, password and phone numbers will be used to create user accounts. A unique code will be sent to the registered phone number and email to ensure authenticity.

1.3.2 Confidentiality

User data will be encrypted and access to private user data will be limited.

1.3.3 Data Integrity

Correct format of addresses, email and phone numbers will be enforced to ensure data integrity.

1.4 Software Quality Attributes

1.4.1 Maintainability

The application design shall aid in bug fixing, future optimization and adding of new functionality in the future. The software would rapidly respond to any command given to it and in case of a failure; it would restore without any hassle.

1.4.2 Availability

Users should be able to see and add stocks to their profile. Denial of Service attacks are not foreseen and will not be treated as a cause for concern. The software would be optimized so that it takes the minimum amount of time to cater any user command without crashing.

1.4.3 Extendibility

The application shall be designed in a way that supports the implementation of optional functional requirements at a later date, making the application with a more modular approach.

1.4.4 Reusability

Components developed for this application can be reused in isolation and integrated with other applications.

1.4.5 Portability

The application will be able to run on web browsers, iOS and Android devices provided it meets the minimum requirements. It will run on mobile phones, making it easy to access on the go.

1.4.6 Reliability

The application will ensure all prices of all stocks are accurate and will try to most probably give an accurate prediction. The software would be designed so that it performs well and consistently without failure. It would work efficiently without utilizing too much resources and the system would be scalable.

1.5 Business Rules

1.5.1 User

- **User Authentication**

User will be required to enter his identification details, address, email, and password.

- If the Submit button is pressed on the user register form page and all mandatory fields are present only then the user will have an account created.

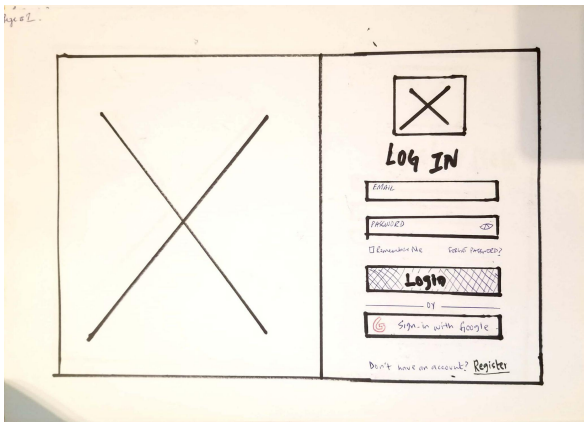
- If any fields are filled incorrectly or left unfilled, the user will be prompted to fill the required fields as needed and until these are fixed, no user account will be created.

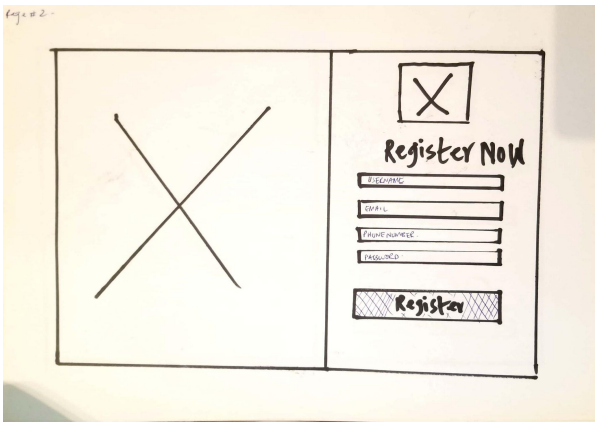
- **Cancellation**

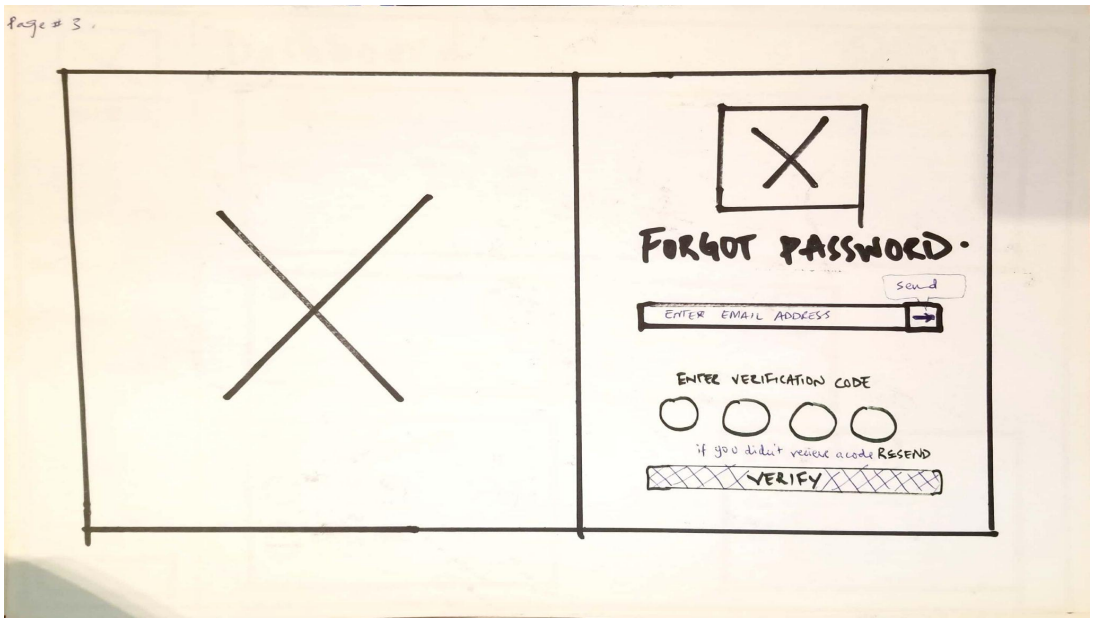
Users will be able to cancel the stock that they have added.

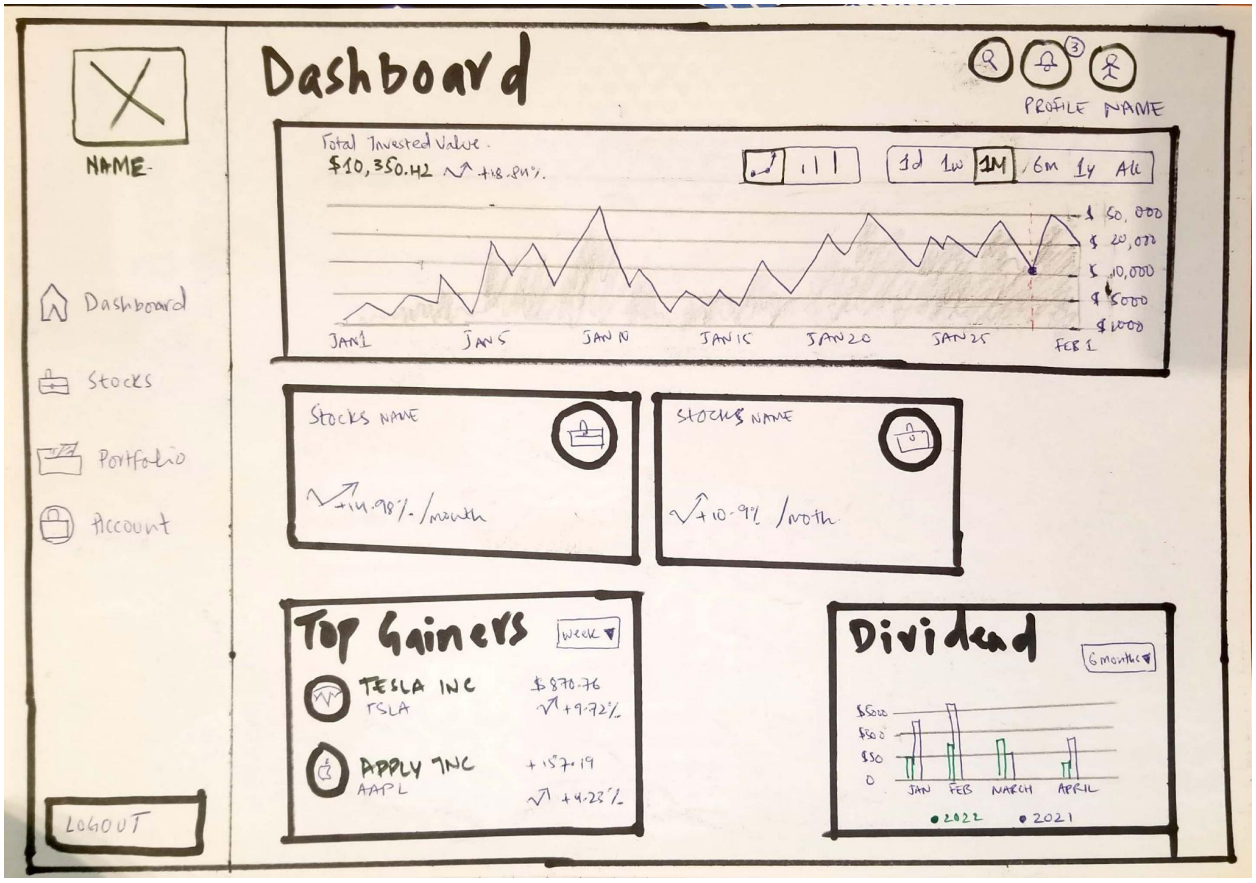
2.3. User Interface Requirements

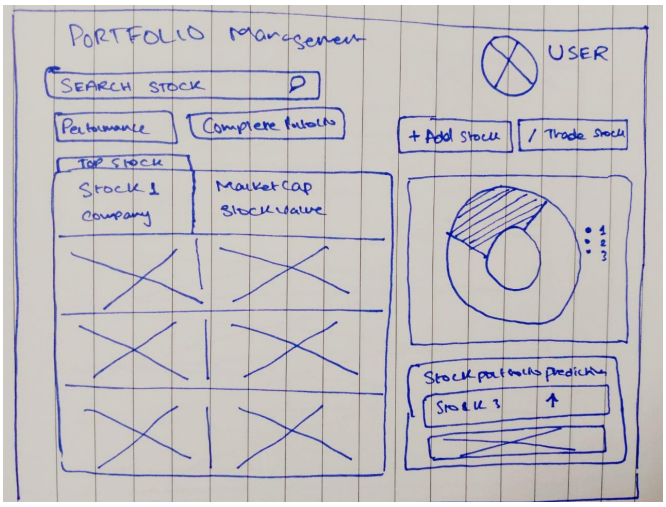
Use graphics to specify these. Neatly handwritten are also acceptable.

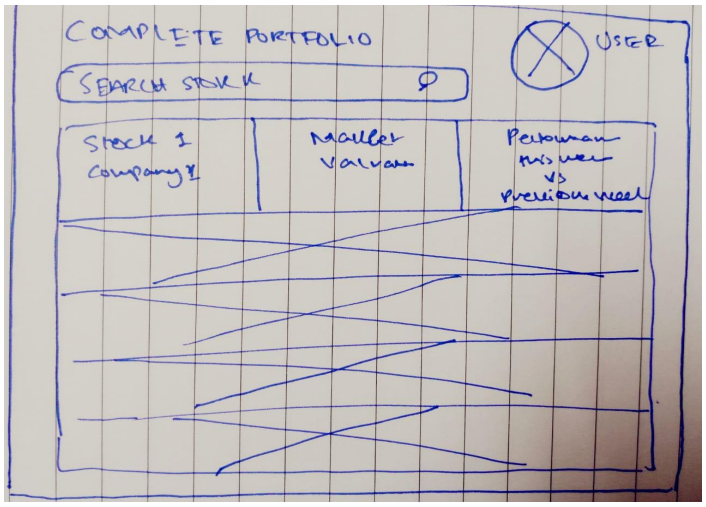







Dashboard
PROFILE NAME
NAME-
Dashboard
Stocks
Portfolio
Account
LOGOUT
Total Invested Value
$10,350.42 +18.84%
1d 1w 1M 6m 1y All
$ 60,000
$ 20,000
$ 10,000
$ 5000
$ 1000
JAN 1
JAN 5
JAN 10
JAN 15
JAN 20
JAN 25
FEB 1
Stocks NAME
+14.98%/month
Stocks NAME
+10.9% /moth
Top Gainers
week
TESLA INC
TSLA
$ 870.76
+9.72%
APPLY INC
AAPL
+157.19
+4.23%
Dividend
6 months
$600
$500
$50
0
JAN
FEB
MARCH
APRIL
2022
2021


PORTFOLIO Management
USER
SEARCH STOCK
Performance
Complete Portfolio
+ Add Stock
/ Trade Stock
Top Stock
Stock 1
Company
Market Cap
Stock Value
1
2
3
Stock portfolio prediction
Stock 3


COMPLETE PORTFOLIO
USER
SEARCH STOCK
Stock 1
Company 1
Market
Valuation
Performance
vs
previous week

3. Functional Requirements Specification

Provide use cases based on 1 and 2 above.

3.1. Stakeholders

- Brokers
- Clients

3.2. Actors and Goals

- The actors involved are the mainly the users and along that database / server

3.3. Use Cases

Account Management

Use Case Name	**Register**
Related Requirements	*A sign up form will be displayed which the user will be required to fill in with their full name, email address, contact number, password and address to sign up.* *The email address will be verified.* *The password will be verified to see if it conforms to the requirements.*

	The account creation will be canceled if the user clicks out of the form at any time before the completion of the form. *If the information entered is valid, the account details will be stored.*
Goal in Context	A new user creates an account and becomes a registered user.
Preconditions	None
Successful End Condition	The data entered by the user is stored in the user account. A confirmation email is sent to the user to the appropriate email address.
Failed End Condition	The user entered invalid data or chose to cancel the account creation. In either case, the account will not be created.
Primary Actors	Guest
Secondary Actors	Database / Server
Trigger	The user requests the system to create an account
Included Cases	Register Verification

Main Flow	Step	Action
	1	The User enters the required User Account information values and requests that the system saves the entered values.
	2	The system validates the entered User Account information.
	3	The values for the User Account information are stored in the User's account. The system notifies the User that the account has been created.
	4	A confirmation email is sent to the user on the appropriate email address

Use Case Name	Log In
Related Requirements	*A login form will be displayed to the user to be filled at the time of login.* *The email and password entered will be verified.* *If an incorrect email or password is entered, an error message will be displayed.* *A login button will be visible to the user.*

	The login request will be canceled if the user cancels the request at any time.	
Goal in Context	The user signs into their account	
Preconditions	User account must exist	
Successful End Condition	The User is authenticated, and the system displays all features available for the role the user is associated with as defined in his/her user account.	
Failed End Condition	This can occur because the User repeatedly entered invalid sign in information. The User has been notified of the reason why he/she was not signed in. The User is not authenticated and remains in the guest role.	
Primary Actors	User	
Secondary Actors	Database / Server	
Trigger	This use case starts when the User accesses the sign in feature of the system.	
Included Cases	Login verification	
Main Flow	**Step**	**Action**

	1	The system prompts the User for his/her username and password.
	2	The User enters his/her username and password.
	3	The system validates the entered information, making sure that the entered username and password are valid for one user account in the system, and that the required password is entered for the entered username
	4	The User is signed in. The system displays a message indicating that the user is signed in.

Use Case Name	**Log Out**
Related Requirements	*A logout button will be available to the user once the user has logged in.* *The logout request will be canceled if the user cancels the request at any time.*
Goal in Context	The user signs out of their account

Preconditions	The user is signed in	
Successful End Condition	The user successfully signs out of their account and enters the anonymous user role.	
Failed End Condition	The user stays signed into their account	
Primary Actors	User	
Secondary Actors		
Trigger	This use case starts when the User accesses the sign out feature of the system.	
Included Cases		
Main Flow	**Step**	**Action**
	1	The user requests the system to logout.
	2	The User is signed out. The system displays a message indicating that the user has successfully signed out.

Use Case Name	**Delete Account**
Related Requirements	*A delete account button will be available to the user once the user has logged in.* *At the time of deletion, the user will be sent a confirmation email.* *After deletion, any user information in the database will be removed.* *The deletion request will be canceled if the user cancels the request at any time.*
Goal in Context	The user deletes their account
Preconditions	User must be logged into the account
Successful End Condition	The user account is deleted
Failed End Condition	The user account is not deleted
Primary Actors	User
Secondary Actors	Database/Server
Trigger	The user requests the system to delete their account

Included Cases	Add item	
Main Flow	**Step**	**Action**
	1	User clicks on the button to delete account
	2	The user is asked to enter their email and password.
	3	The user information is verified
	4	A confirmation message pops up on the screen to confirm the users request
	5	The account is deleted

Use Case Name	**Forgot Password**
Related Requirements	*The forgot password button will prompt the user to enter a valid email address* *A reset password link will be sent to the user's email address if the address is verified.* *The new password entered will be updated in the database.*

	A 'forgot password' option will be visible to the user at the time of login.	
Goal in Context	The user resets his/ her password if they have forgotten their password	
Preconditions		
Successful End Condition	The user resets their account password	
Failed End Condition	The user is not able to reset their password	
Primary Actors	User	
Secondary Actors		
Trigger	The user requests click on the forgot password button	
Included Cases		
Main Flow	**Step**	**Action**
	1	User clicks on the forgot password button

	2	User is asked to enter their email address
	3	The email address is verified to see whether an account with the given email exists
	4	The user enters the new password
	5	The new password is set and the user can now access their account using the new password

USE CASE NAME	EDIT PROFILE
Related Requirements	*Once the user has logged in, they will be able to edit their profile at any time.* *If the user cancels the edit profile request, the information will not be updated.* *The edited user information will be updated in the database.*
Goal In Context	The user should be able to change his profile details such as username, password, email or address etc.

Preconditions		User needs to be logged in
Successful End Condition		The user is successfully able to edit his account details
Failed End Condition		An error is shown while changing any details like if a password is already taken
Primary Actors		User
Secondary Actors		Verification Database/Server
Trigger		A user clicks the sidebar on home page and then clicks on profile where he selects edit
Included Cases		-
MAIN FLOW	**STEP**	**ACTION**
	1	A user has logged in and is on the home page
	2	The user clicks the side bar and selects profile
	3	He then selects edit in profile and changes his details
	4	The new details are stored in the server and a details changed message is displayed

Stock-Dashboard Management

Use Case Name	View stock prices (Dashboard - Main Page)
Related Requirements	System will fetch data from the PSX site. System will display future trends based on time series. System will display the current stock listing at KSE-100 along with daily price trends. System will display top performing stocks in user's portfolio.
Goal in Context	The user will be able to view a bird eye view of stock market and will be able to assess own portfolio.
Preconditions	Users must be logged in. To view a portfolio , users must first make a purchase via the app.
Successful End Condition	User able to view portfolio and analyze stock market trend.
Failed End Condition	Users are not able to view market trends or portfolios.
Primary Actors	User
Secondary Actors	-

Trigger	User logs into their account and lands on the home screen.	
Included Cases		
Main Flow	**Step**	**Action**
	1	User logs into the app.
	2	System fetches current data of KSE and refreshes portfolio data accordingly.
	3	System displays data onto the user dashboard.
	4	User views the on-site generated time-series graphs and user portfolio.
	5	User interacts successfully with the dashboard.

Use Case Name	Add a stock to portfolio
Related Requirements	System allows users to add/remove stocks from their portfolio System will display top performing stocks in user's portfolio.

Goal in Context	To enable a user to add a stock to their portfolio	
Preconditions	The user must be logged in.	
Successful End Condition	The user is able to add a stock to their portfolio	
Failed End Condition	The user is not able to add the stock to their portfolio	
Primary Actors	User	
Secondary Actors	Database	
Trigger	User logs into their account, lands on the home screen and clicks on the add stock button	
Included Cases		
Main Flow	**Step**	**Action**
	1	User logs into the app.
	2	User clicks on the Add stock button
	3	User selects a stock from the list and enters the details

	4	User clicks on the add button
	5	User successfully added a stock to his portfolio

Use Case Name	Remove a stock from portfolio
Related Requirements	System allows users to add/remove stocks from their portfolio System will display top performing stocks in user's portfolio.
Goal in Context	To enable a user to remove a stock from their portfolio
Preconditions	The user must be logged in.
Successful End Condition	The user is able to remove a stock from their portfolio
Failed End Condition	The user is not able to remove the stock from their portfolio
Primary Actors	User
Secondary Actors	Database

Trigger	User logs into their account, lands on the home screen and clicks on the remove stock button	
Included Cases		
Main Flow	**Step**	**Action**
	1	User logs into the app.
	2	User clicks on the Remove stock button
	3	User selects a stock from the list of stocks already in his portfolio
	4	User clicks on the remove button
	5	User successfully removed stock from his portfolio

4. User Interface Specification

For each use case specify the user interface. All reports generated by the system are also part of user interface specification.

Use case: Login and Register and Edit Profile

- Simple registration form with email and password. SSO widgets i.e Gmail, Facebook , GitHub will be below.
- Signup form will have simple data input form i.e Name, Email , Password , Confirm Password.
- Simple form displayed to edit the form fields previously given by user.
- Greeting message to be displayed on loading time.

Use case: Forgot Password, Log out , and Delete account.

- Simple email confirmation will be sent to the user with a link to generate / reset password as user clicks on it.
- Simple confirmation message will be displayed to user when logging out.
- To delete the account, the user will be required to provide password again and upon confirmation account will be deleted.

Use case: Main Dashboard

- The main dashboard will have a clean and modern design, with a navigation bar at the top and a main content area below. The navigation bar will have links to different sections of the application, such as a portfolio overview, and a news feed. The main content area will display a grid of stock cards, each displaying the name, symbol, current price, and change in price for a specific stock. The user will be able to sort and filter the stock cards by different criteria, such as sector or performance.