SRS document for Stock Prediction Application

Presented by:

18BCS4302	**Keshav Kaushik**
18BCS4295	**Shiva Tanwar**
18BCS4297	**Divyanshu Singh Bisha**

Table of Contents

INTRODUCTION

1.1 PURPOSE:

This application is designed to provide its users with a way to predict stock prices by providing historical trends and people's reaction and comments about any particular stock.

1.2 DEFINITION, ACRONYMS and ABBREVIATIONS:

LSTM- Long Short-Term Memory
RNN- Recurrent Neural Network
NLP- Natural Language Processing
NB- Naive Bayes
UI/UX - User Interface/User Experience

1.3 PRODUCT SCOPE

The final product will provide a platform to its users to make educated guesses without the hassle of sitting down to search up the whole internet for stock related information. This will help its users to be a step ahead of others.

1.4 TECHNOLOGIES TO BE USED:

1. Flutter for developing the application and the layout and assets are saved in the form of json.
2. Historical data of stocks saved in the form of csv(fetched from yahoo finance).
3. Python and Jupyter notebook to prepare the deep learning models.
4. LSTM RNN for creating the historical data prediction model.
5. NLP(NB classifier) for sentimental analysis.
6. Firebase to run the models to run in back-end.

OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE:

The product perspective is to provide a user friendly environment. The users may use the application without any hassle of logging in as the application does not request any personal data thus maintaining privacy. The users can look at the historical trends of stock data and get the real-time reactions and comments of people in twitter to get the idea of a stock's standing and thus make educated guesses about the future stock prices.

2.2 USER CLASSES AND CHARACTERISTIC:

1. USER : The user can access the live build provided to them. They can view the present prices, historical trends and people's comments about a stock.

2.3 OPERATING ENVIRONMENT

APP

1. Android 15 or higher
2. IOS device
3. Flutter web app

2.3 SOFTWARE INTERFACES

1. APP: An application, where all the aforementioned details about stocks are displayed. The application's front-end is made using Flutter for both android users and IOS users whereas the back-end is made using Firebase and python.

2.5 EXTERNAL FORCES

ASSUMPTIONS AND DEPNDENCIES :

1. Basic knowledge of smartphones among the users.
2. The users should be acquainted with the stock market.
3. Availability of internet connection for the usage of the app.
4. Ability to read line graphs.

SYSTEM FEATURES

3.1 FUNCTIONAL REQUIREMENTS

DEFINITION :

It defines WHAT THE SYSTEM SHOULD DO. In software engineering, a functional requirement defines a system or its component. A function is nothing but inputs, its behaviour, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. Functional Requirements are also called Functional Specification. Functional software requirements help you to capture the intended behaviour of the system. This behaviour may be expressed as functions, services or tasks or which system is required to perform.

LIST OF FUNCTIONAL REQUIREMENTS FOR THE WEBSITE:

1. SELECT STOCK- The application will provide a list of stocks
   INPUT- Selecting the desired stock.
   OUTPUT- The present prices of the stock.

2. DROP DOWN MENU - A drop down is provided to the users upon entering the page after selecting their desired stock.
   INPUT- Tapping the 3-bar drop down
   OUTPUT - Three options are provided to navigate through the application.

3. PRESENT DATA - A tab povided by the application to view the present data about the stocks.
   INPUT- Tapping on PRESENT DATA tab.
   OUTPUT- The page containing the present data of the stocks is displayed.

4. HISTORICAL TRENDS – A tab will be provided by the application to view the historical trends of the selected stock.
   INPUT- Tapping on HISTORICAL TRENDS tab.
   OUTPUT- A page consisting of a graph displaying the historical trends of the selected stock.

5. PEOPLE'S COMMENTS- This option is provided to see the comments of the people about the selected stock.
   INPUT- Clicking on PEOPLE'S COMMENT option.
   OUTPUT- A page consisting of the percentage of positive and negative comments and the tweets separated by their sentiment.

6. BACK- This option is provided to go back to the stock selection page.
   INPUT- Clicking on BACK option.
   OUTPUT- Stock selection page is displayed.

3.2 NON-FUNCTIONAL REQUIREMENTS

Definition :

It defines HOW A SYSTEM SHOULD WORK WHILE PERFORMING OPERATIONS. A non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours.

LIST OF NON-FUNCTIONAL REQUIREMENTS:

1. USABILITY : The system should be designed in such a way that any naive user can easily use the system. In other words, the system should be user friendly and not complicated.
   Eg: The app and website is designed in a way that it provide best UI/UX.

2. AVAILABILITY : The application/system should be available for use whenever needed.
   Eg: Servers are running 24/7 to provide availability at any time.
3. SCALABILITY : The system should be scalable enough to accommodate changes as and when required.
   Eg: Real time changes is reflected on all the platforms running

4. RELIABILITY : The system should be reliable enough to handle the problems, i.e. it should clearly state the steps that need to be taken in case of an unlikely situation.
   Eg: Good testing results in less failure, if failure occurs so the recovery algorithm can handle it.

5. EFFICIENCY : The system should be efficient, i.e. the response time of the system should be fast.
   Eg: Everything will happen asynchronously so it can respond faster.