Software Design Report

for

Stocks Trend Prediction

Version 1.0

Prepared by

|  |  |  |
| --- | --- | --- |
| Hassan Ahmed Khan | 201907008 | hassanahmedkhan2018@gmail.com |

|  |  |
| --- | --- |
| Internal: | Engr. Kashif Jamil |
| External: | - |
| Disicipline: | BS – Software Engineering |

Table of Contents

**88**

**88**

**88**

88

88

89

89

**89**

89

89

**89**

89

89

89

**90**

**90**

90

**91**

**91**

**92**

92

92

**93**

93

1. **Introduction………………………………………………………………………………**
2. **Product Features…………………………………………………………………………**
3. **User Interface……………………………………………………………………………**
   1. Component 1: Predictions……………………………………………………………
   2. Component 2: Stocks Information…………………………………………………….
   3. Component 3: Time Series Predictions………………………………………………..
   4. Component 4: Stocks News…………………………………………………………….
4. **Interfaces to External Hardware and Software…………………………………………**
   1. Hardware Interface…………………………………………………………………….
   2. Software Interface…………………………………………………………………….
5. **Non – Functional Requirement…………………………………………………………..**
   1. Performance Requirements…………………………………………………………..
   2. Safety Requirements………………………………………………………………….
   3. Security Requirements…………………………………………………………………
6. **Data Storage……………………………………………………………………………..**
7. **High Level Designs………………………………………………………………………**
   1. Class Diagram………………………………………………………………………….
8. **Design Verification……………………………………………………………………….**
9. **Gantt Chart………………………………………………………………………………**
10. **CPM Model………………………………………………………………………………**
    1. CPM Chart……………………………………………………………………….
    2. Critical Path Diagram……………………………………………………………
11. **Estimation…………………………………………………………………………………**
    1. The Intermediate Cocomo………………………………………………………..
12. **Introduction**

The system architecture designed for the mobile application is a client-server architecture. The client-side is the user interface that the users interact with while the server-side is the combination of the database and the stock price prediction model.

The performance of a few stock price prediction models was examined in order to select the best performance model to predict the stock prices. All the historical stock prices of the stocks were retrieved from the YahooFinance.com website through an API which retrieve the data from the website to python file. This can ensure that the data can run automatically through the model without manually download the .csv files from the website.

1. **Product Features**

* Maximum price received from during the selected days and years.
* Minimum price received from during the selected days and years.
* Average price received from during the selected days and years.
* Volume received from during the selected days and years.
* Daily Opening and Closing price during the selected days and years.
* Investors of the searched ticker.
* Balance sheet of the searched ticker.
* Institutional shareholders of the searched ticker.
* Stocks value of the searched ticker.
* Getting the news about searched ticker.
* Predictions of the searched tickers.
* Time Series Analysis of the searched ticker.

1. **User Interface**

**3.1 Component 1: Predictions**

|  |  |  |
| --- | --- | --- |
| **Input** | **Actions** | **Outputs** |
| .csv file from yfinance | CNN-LSTM Models | Predictions of secarched ticker |

**3. 2 Component 2: Stocks Information**

|  |  |  |
| --- | --- | --- |
| **Input** | **Actions** | **Outputs** |
| Input/Output date/  Input Tickers | Stock\_data.history | Getting informations about stocks |

**3.3 Component 3: Time Series Predictions**

|  |  |  |
| --- | --- | --- |
| **Input** | **Actions** | **Outputs** |
| .csv file from yfinance | FbProphet | Time Series Predictions of secarched ticker |

**3.4 Component 4: Stocks News**

|  |  |  |
| --- | --- | --- |
| **Input** | **Actions** | **Outputs** |
| Stocks Name (ticker) | Gnews models | Getting the news about stocks |

1. **Interfaces to External Hardware and Software**

**4.1 Hardware Interface**

Depending on executing the system our hardware should be well maintained and able to execute the program in a well upgraded hardware.

**4.2 Software Interface**

CNN-LSTM models are powerful, especially for retaining a long-term memory, by design in a pure machine learning interest. In my opinion, the model has observed certain patterns in the data, thus giving it the ability to correctly predict the stocks movements most of the time.

1. **Non-Functional Requirements**

**5.1 Performance Requirements**

Performance of overall system is very efficient and well optimize. From the time taken to capture and process it everything is well organized. While processing the stock price it will take time depending on the hardware we will be executing this on.

**5.2 Safety Requirements**

This syetm does not contain any critical data. Still it provide. The databases that are accessed are locally executed. In case of any update in libraries used can lead to system failure.

**5.3 Security Requirements**

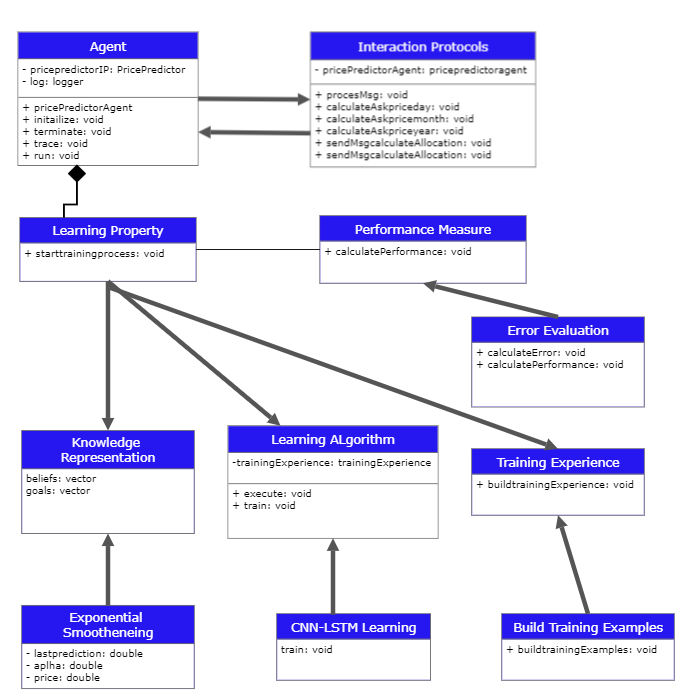
All the libraries used are certified and standard. As it is demo based program after that is released.

1. **Data Storage**

Although the application has many data objects, it does not have any data storage. All the objects and their related data are handled by the models. So the developers need not think about data storage. For this reason, data model is redundant for this project.

1. **High Level Designs**

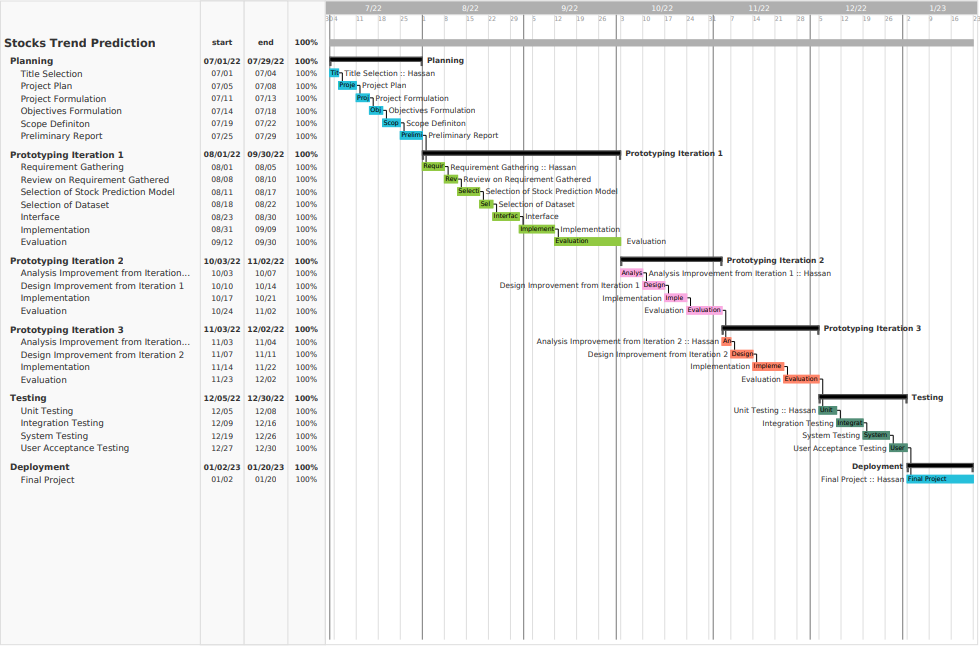
**7.1 Class Diagram:**



1. **Design Verification**

Detail design can be checked with different tabs of ipad, different resolution of screens and different engines.

1. **Gantt chart**



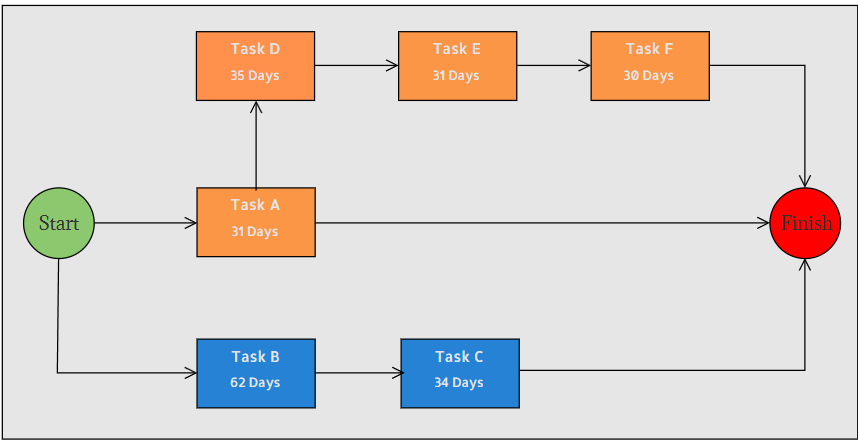
**Gantt Chart**

1. **CPM Model**

**10.1 CPM Chart**

|  |  |  |
| --- | --- | --- |
| **Task ID** | **Task** | **Duration (days)** |
| A | Planning Phase | 31 |
| B | Prototyping Iteration 1 | 62 |
| C | Prototyping Iteration 2 | 34 |
| D | Prototyping Iteration 3 | 35 |
| E | Testing | 31 |
| F | Deployment | 30 |
|  | **Total days:** | **223** |

**10.2 Critical Path Diagram**

****

1. **Estimation**

***11.1 The Intermediate COCOMO:***

Effort (E) = a\*(KLOC)b \*EAF  MM  
Scheduled Time (D) = c\*(E)d  Months(M)

Where,

* E = Total effort required for the project in Man-Months (MM).
* D = Total time required for project development in Months (M).
* KLOC = The size of the code for the project in Kilo lines of code.
* a, b, c, d = The constant parameters for the software project.

EAF = It is an Effort Adjustment Factor, which is calculated by multiplying the parameter values of different cost driver parameters. For ideal, the value is 1.

---------------------------------------------------------------------------------------------------

Estimated size of the project is: 1.0 KLOC

The complexity of the project is high having parameter: 1.15 (as per cocomo cost driver parameter)

Developer having normal experience in programming: 1 (as per cocomo cost driver parameter

---------------------------------------------------------------------------------------------------

**EAF** = 1.15\*1 = **1.15**

**Effort (E)** = a\*(KLOC)b \*EAF = 3.0\*(1.0)1.12 \*1.15 = **3.45 MM**

**Scheduled Time (D**) = c\*(E)d  = 2.5\*(3.45)0.35= **3.85Months(M)**

**Productivity of Software** = KLOC/E = 1.0/3.45 = **0.289 KLOC/MM = 289 LOC/MM**

**---------------------------------------------------------------------------------------------------**