

SOFTWARE ENGINEERING AND PROJECT MANAGEMENT



Batch Number: 2

- Aishwarya Lakshmi A P (RA2011026020066)
- Amruthaa S (RA2011026020073)
- Grace Hephzibah M (RA2011026020084)

TABLE OF CONTENTS

SNO	DATE	CONTENT	PAGE NO	SIGNATURE
1	08/03/2022	PROBLEM STATEMENT 1.1 Identify the software project problem statement 1.2 Create Business Class Template		
2	15/03/2022	STAKEHOLDERS & PROCESS MODELS 2.1 Identifying Stakeholders 2.2 User story 2.3 Identify the Appropriate Process Modules 2.4 Arrive at Problem Statement 2.5 Comparison between Agile and Waterfall Model		
3	22/03/2022	IDENTIFYING THE REQUIREMENTS FROM THE PROBLEM STATEMENT 3.1 Requirements 3.2 Functional Requirements 3.3 Non-Functional Requirements 3.4 Hardware Requirements 3.5 Software Requirements		
4	29/03/2022	PROJECT PLAN AND PROJECT EFFORT BASED ON RESOURCES 4.1 Project Plan 4.2 Identifying Job Roles and Responsibility		
5	05/04/2022	PROJECT EFFORT BASED ON RESOURCES 5.1 Work Breakdown structure 5.2 Risk Management		
6	12/04/2022	ESTIMATION OF PROJECT METRICS 6.1 Function Point Analysis 6.2 COCOMO Model		
7	19/04/2022	DESIGN 7.1 System Architecture		

8	26/04/2022	MODELING UML USE CASE DIAGRAMS & CAPTURING USE CASE SCENARIOS 8.1 Use Case Diagram Description 8.2 Use Case Diagram		
9	03/05/2022	ER MODELING FROM THE PROBLEM STATEMENTS 9.1 ER Diagram Description 9.2 ER Diagram		
10	10/05/2022	IDENTIFYING DOMAIN CLASSES FROM THE PROBLEM STATEMENTS 10.1 Domain Class Diagram Description 10.2 Domain Class Diagram		
11	17/05/2022	STATECHART AND COMMUNICATION MODELING 11.1 State Chart Diagram Description 11.2 State Chart Diagram 11.3 Communication Diagram Description 11.4 Communication Diagram		
12	24/05/2022	MODELING UML CLASS DIAGRAM & SEQUENCE DIAGRAMS 12.1 Class Diagram Description 12.2 Class Diagram 12.3 Sequence Diagram Description 12.4 Sequence Diagram		
13	31/05/2022	MODELING DATA FLOW DIAGRAM 13.1 Data Flow Diagram (DFD) Description 13.2 Data Flow Diagram (DFD)		
14	07/06/2022	IMPLEMENTATION 14.1 Module Implementation		
15	14/06/2022	ESTIMATION OF TEST COVERAGE METRICS & STRUCTURAL COMPLEXITY 15.1 Control Flow Graph 15.2 McCabe's Cyclomatic Complexity		

		15.3 Optimum Value of Cyclomatic Complexity		
16	21/06/2022	<i>DESIGNING TEST SUITES</i> 16.1 Software Testing 16.2 Testing Frameworks 16.3 Master Test Plan 16.4 Manual Testing		
17	28/06/2022	<i>CONCLUSION</i>		
18	28/06/2022	<i>REFERENCES</i>		

PROJECT COMPLETED

Ex 1. Problem Statement

Date	08/03/2022
Submitted By	Aishwarya Lakshmi A P - 66 Amruthaa S - 73 Grace Hephzibah M - 84
Title/ Role	Stock Market Prediction



1.1 Identify the software project problem statement

Introduction:

The stock market extensively refers to the cluster of exchanges and other platforms where the buying, selling, and issuance of shares of publicly held organisations take place. Such financial movements are conducted through standardised formal exchanges (whether physical or electronic) or via over-the-counter (OTC) marketplaces that function under a defined set of regulations.

Problems:

The stock market prediction model

- Hard to provide insights and intuitive forecasts on the future
- To capture the fluidity of the stock changes
- Creates a certain degree of short-sightedness in the entire evaluation process
- Complexity is extremely high and requires intensive prior knowledge in the field

Proposal:

The above problems can be overcome by our Stock Market Prediction Model. Instead of using pre-processed data, we can feed the model with real-time data which increases efficiency and reduces the complexities thereby providing higher accuracy.

1.2 One Page Business Case Template

The Project:

To develop a state-of-the-art artificial recurrent neural network, to better forecast the stock market.

Analyzing and predicting the regularity of stock markets has always been a key point for investors and investment firms. Stock price predictions are crucial for investors.

This project would assist anyone interested in investing in the stock market to make the right decisions and understand the future state of the market.

The History:

1. Stock Market Prediction using Simple Linear Regression

- The prediction model was run on one stock set only and not on the entire market. This creates a certain degree of short-sightedness in the entire evaluation process.
- The method of prediction using Linear Regression is comparatively less accurate when compared to other methods.
- Does not consider the Random Forest prediction model for prediction, which theoretically gives a higher accuracy when run on a small dataset.
- International Conference on Electronics, Communication and Aerospace Technology (ICECA 2017)

2. Stock Market Prediction using SVM and PCA.

- The complexity of the undertaken project is extremely high and requires intensive prior knowledge in the field.
- This paper does not provide a benchmark comparison or study, to evaluate the developed model against.
- 2015 19th International Conference on System Theory, Control and Computing (ICSTCC), October 14-16, Cheile Gradistei, Romania

3. Stock market Prediction using Artificial Neural Networks.

- This paper aims at developing an algorithm to predict stock values but does not talk about the accuracy of prediction. It gives a qualitative approach, not a quantitative one.

- It provides a semantic figure and not a visual outcome to analyse the prediction. No graphs are provided to demonstrate patterns in investment, in the market.
- 2017 International Conference on Technical Advancements in Computers and Communications.

4. Stock Market prediction using Hidden Markov Model.

- The Hidden Markov Model is only accurate in a short period of prediction. If the time period increases, then the model's performance deteriorates considerably.
- Only an empirical comparison of prediction models is done. The models are not tested on an out-of-sample dataset. So, we cannot talk about the real-time performance of the model from the given literature.
- 2013 IEEE International Conference on Granular Computing (GrC).

5. Stock Market Prediction using Sentiment Analysis: Hybrid Approach.

- No visual representation of the prediction model is provided.
- Does not implement regression. Only clustering techniques are used to predict stock values.
- Does not provide a dynamic model for comparison or prediction, but rather uses only fixed values. Hence, it's applicability in the real world is questioned, without any valid evidence to prove otherwise.
- International Conference on Computing, Communication and Automation (ICCCA2016)

6. Stock Market Prediction Model using ANN and Back Propagation.

- Requires in-depth knowledge of deep learning and neural networks, to understand and implement.
- Is a vast difference in the empirical and actual accuracy as represented by the models.
- ICCCNT'12 26th _28th July 2012, Coimbatore, India

7. Multiple Linear Regression Model to predict Audit Opinions.

- This paper only studies the Shanghai and Shenzhen Stock Markets of 2007 for a year and hence is considerably short-sighted.
- This study does not clearly classify clean audit opinions from the unclean ones.
- 2009 ISECS International Colloquium on Computing, Communication, Control, and Management

Constraints:

1. Requires a ton of understanding of the stock market. This in turn means that the user must understand the opening rate, the closing rate, volume, minimum rate and the maximum rate.
2. Frequent monitoring of the stocks is required to make sure that no loss is faced by the user.
3. Requires experience to forecast the rise or fall of the stock and depending on it, the decisions of investing or liquidating the stock.

The Approach:

The purpose of this project is to predict the returns of certain stocks. The development of this project can be widely classified into 2 parts:

1. Using LSTM neural networks to predict stock prices
2. Dashboard using Plotly dash for stock analysis

Benefits:

1. Maximize the profit by purchasing volume at the lowest price and liquidating the stock when it's at its highest.
2. Minimize the loss by selling the stock just at the right time, so users wouldn't have to face loss over their investments.
3. Processes real-time data thereby permitting the model to be authentic and comprehend live input.
4. Provides beneficial feedback on the shares of the company and foresee its elevation and decline concerning the previous situations.

Ex 2. Stakeholders And Process Models

Date	15/03/2022
Submitted By	Aishwarya Lakshmi A P – 66 Amruthaa S – 73 Grace Hephzibah M – 84
Title/ Role	Stock Market Prediction



2.1 Identifying Stakeholders

1) User:

- The service provided by the final outcome of this project will be utilized by the users.
- In this project, the end-user will be viewing and using the stock price predictions to make improvised decisions on investments.

2) Sponsor:

- The sponsor is the person or organization that provides financial aid for the project.
- Resources will be made available only with the financial support of sponsors.
- This project is currently self-sponsored.
- Further upgrades or extensions of the project could require the support of sponsors, or alternatively a crowd-sourcing initiative.

3) Program Manager:

- A program manager defines the project's objectives and strategies.
- They overlook and coordinate all tasks and activities.

4) Project Manager:

- Project managers are responsible for resource planning, organising, and taking the project forward to completion.

- Maintaining progress and required standards for the project is the main role of the project manager

5) ML Prediction System:

- The Machine Learning Prediction System uses input data to predict stock market prices.
- The final product is produced by this system.

6) Project Team:

- The project is carried out by the project team, which consists of the project manager and the team members.

2.2 User Story

1) Users of the final product:

An user here is the individual or in better words, the investor, who uses the stock price predictions provided by the final product of this project.

2) Prediction Provider:

The prediction provider gives predictions of future stock market prices. The users utilize this information to make better investments.

2.3 Identifying the Process Modules

This project is implemented using the following modules:

1) ML Prediction Model:

This module is the core of this project in which input data is fed to get stock price predictions as a result. Optimization of the final product depends on this module.

2) Dashboard:

This module is used for visualization of prediction given by the ML Prediction Model module. This improves usability and the success of the user experience is dependent on this module.

3) Deployment:

This module is responsible for presenting the final product to users. Without this module, the final product will not be accessible to interested users.

4) Real-Time Analysis

This module adds an important feature to the final product, which is the ability to extract and utilize real-time data. Target sources are identified and utilised for obtaining data in real-time.

2.4 Arriving at the Problem Statement

Problem Statement:

- Prediction accuracy issues
- Server issues
- Dashboard access issues
- Loading issues in the final product
- Issues in obtaining real-time data

2.5 Comparison between Agile and Waterfall Model

<u>Waterfall Model</u>	<u>Agile</u>
Requirement docs	Just in time, informal requirements
Occasional “customer” involvement	Frequent “customer” involvement
Start to finish project plan	Product backlog. Plan for Sprint. Details are sketchy beyond that. Priorities shift based on new data.
Tasks are assigned	Assigned tasks are a bottleneck
Potentially a large team	Team of 3-9 people
Multiple phases, eventual delivery	Working software each Sprint/iteration
Resistant to change	Change is expected
Contract says what we build, deliver	Contract is a lot closer to T&E

This project is developed and delivered by a self-organised team. Highly motivated and skilled individuals are involved in this project. As the project develops, changes and modifications to the requirements are welcome. Hence this project implements and adopts the Agile model.

Ex 3. Identifying The Requirements From The Problem Statement

Date	22/03/2022
Submitted By	Aishwarya Lakshmi A P - 66 Amruthaa S - 73 Grace Hephzibah M - 84
Title/ Role	Stock Market Prediction



3.1 Requirements:

Project requirements are conditions or tasks that must be completed to ensure the success or completion of the project. They provide a clear picture of the work that needs to be done.

3.2 Functional Requirements:

It describes the behaviour of a particular system. In this project, the dashboard is the visualization entity which is the functional requirement. The details are covered as follows

- Analysis
- Design
- Implementation
- Maintenance

3.3 Non - Functional Requirements:

- The system can load at the speed of 2.4GHz-3.6GHz
- Memory 4GB RAM
- Transferring data speed of 50 Mbps in time

3.4 Hardware Requirements:

- Processor: Pentium – IV
- Hard drive: 320 GB
- RAM: 4GB

3.5 Software Requirements:

- Operating system: Windows 10
- Programming Language: Python
- Data Visualization: Plotly Dash

Ex 4. Project Plan and Project Effort

Based On Resources

Date	29/03/2022
Submitted By	Aishwarya Lakshmi A P - 66 Amruthaa S - 73 Grace Hephzibah M - 84
Title/ Role	Stock Market Prediction



4.1 Project Plan:

Project Name:

“Stock Price Prediction”

Project Members:

Our project consists of 3 members:

1. A P Aishwarya Lakshmi - RA2011026020066
2. Amruthaa S - RA2011026020073
3. Grace Hephzibath M - RA2011026020084

Modules:

- ML Prediction Model
- Dashboard
- Deployment
- Real-Time Analysis

Scheduling:

Task	Start Date	End date
Identifying Problem Statement and Business Case Development		
Identifying Stakeholders, Process Modules and Required Modules		
Identifying the Requirements		
Cost Estimation and Setting up a Budget		
UML Diagrams		
Coding		
Testing		
Final Revisions		

Project Cost:

- Development of the project:

Resource Requirement	COST
Laptop with core i7 8th gen processor, with 8Gb RAM, running on Windows 10	Rs.80000
Hard drive: 320 GB	Rs.2000
Printing	Rs.300

- Server-End:

Resource Requirement	COST
Hosting with custom domain	Rs.30000
Streamlit	--

- Other Cost:

Employee Salary	--
Maintenance Cost	Rs.10000

4.2 Identifying Job Roles and Responsibilities:

Members	Roles and Responsibilities
A P Aishwarya Lakshmi (RA2011026020066)	<ul style="list-style-type: none"> ● TEAM LEADER: Responsibility of coordinating the team, checking for errors, updating for the current of the project, guiding the team. ● DEVELOPER: Develop a stock price prediction model, to build an interactive dashboard for stock analysis. ● ARCHITECT: Design of UML Diagrams and other blueprints.
Amruthaa S (RA2011026020073)	<ul style="list-style-type: none"> ● DEVELOPER: Develop a stock price prediction model, to build an interactive dashboard for stock analysis. ● TESTER: Responsibility for testing the project at a smaller level and reporting the errors. ● DESIGNER: Identifying areas for

	modifications in existing programs and developing the modifications.
Grace Hephzibah M (RA2011026020066)	<ul style="list-style-type: none">● DEVELOPER: Develop a stock price prediction model, to build an interactive dashboard for stock analysis.● TESTER: Responsibility for testing the project at a vast level and variety of ways.● DESIGNER: Identifying areas for modifications in existing programs and developing the modifications.

Ex 5. Project Effort Based on Resources

Date	05/04/2022
Submitted By	Aishwarya Lakshmi A P - 66 Amruthaa S - 73 Grace Hephzibah M - 84
Title/ Role	Stock Market Prediction



5.1 Work Breakdown Structure

A Work Breakdown structure is a deliverable – oriented hierarchical decomposition of the work to be executed by the project team to accomplish the project objectives and create the required deliverables. A WBS is the cornerstone of effective project planning, execution, controlling, monitoring and reporting. All the work contained within WBS is to be identified, estimated, scheduled and budgeted.

The work breakdown structure for this project can be found below:

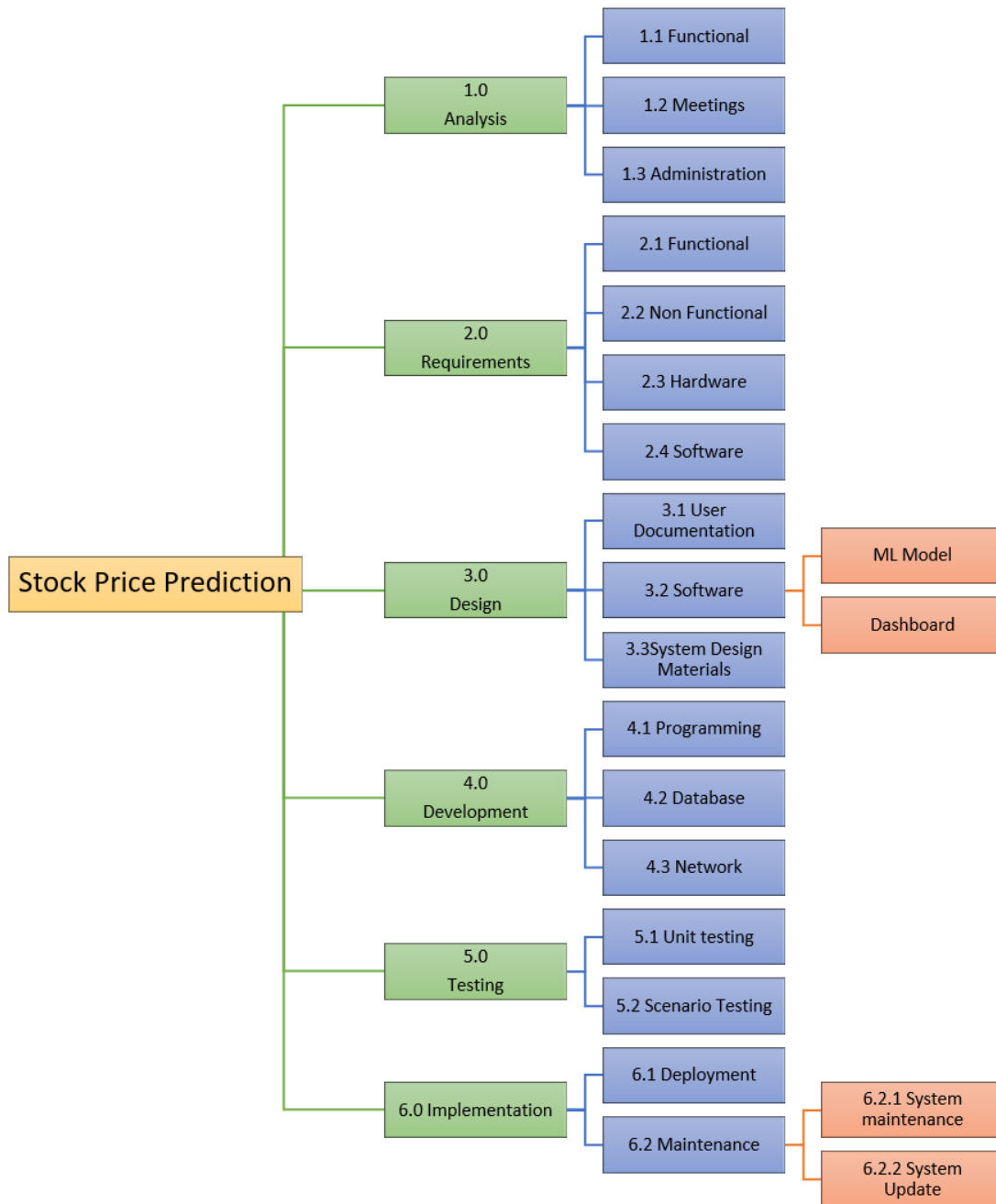


figure 5.1

5.2 Risk Management

In the modern world, risk management refers to the practice of identifying potential risks in advance by analyzing them and taking precautionary steps to curb the risk.

Risk management is the identification, evaluation, and prioritization of risks, controlling the probability or impact of unfortunate events.

Risks to be handled:

- Server downtime
- Database maintenance
- Unauthorized access to data
- Low website speed
- Improper internet connection
- Hacking and modification of data
- Product bugs

Managing Risks:

- Proper and periodic maintenance of server
- Back-up and recovery services
- Proper resource management
- Use of Captcha or other security methods to protect data
- Proper testing and validation of product for bug removals

Ex 6. Stakeholders And Process Models

Date	12/04/2022
Submitted By	Aishwarya Lakshmi A P - 66 Amruthaa S - 73 Grace Hephzibah M - 84
Title/ Role	Stock Market Prediction



6.1 Function Point Analysis

Function Point Analysis (FPA) is a method or set of rules of Functional Size Measurement. It assesses the functionality delivered to its users, based on the user's external view of the functional requirements.

Types of FPA:

- **External Input (EI):** EI processes data or control information that comes from outside the application's boundary. The EI is an elementary process.
- **External Output (EO):** EO is an elementary process that generates data or control information sent outside the application's boundary.
- **External Inquiries (EQ):** EQ is an elementary process made up of an input-output combination that results in data retrieval.
- **Internal Logical File (ILF):** A user identifiable group of logically related data or control information maintained within the boundary of the application.

- **External Interface File (EIF):** A group of users recognizable logically related data allusion to the software but maintained within the boundary of another software.

SI NO	Measurement Parameters	Examples
1.	Number of External Input (EI)	Real-Time Data Series
2.	Number of External Output (EO)	Dashboard
3.	Number of External Inquiries (EQ)	Prompts and Interrupts
4.	Number of Internal Logical File (ILF)	Directories
5.	Number of External Interface File (EIF)	Shared Routines

6.2 COCOMO MODEL

Cocomo (Constructive Cost Model) is a regression model based on LOC, i.e number of Lines of Code. It is a procedural cost estimate model for software projects and is often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time, and quality.

Different models of Cocomo have been proposed to predict the cost estimation at different levels, based on the amount of accuracy and correctness required. These characteristics pertaining to different system types are mentioned below:

1. **Organic** – A software project is said to be an organic type if the team size required is adequately small, the problem is well understood and has been solved in the past and also the team members have a nominal experience regarding the problem.
2. **Semi-detached**- A software project is said to be a Semi-detached type if the vital characteristics such as team size, experience, and knowledge of the various programming environments lie in between that of organic and Embedded.

Eg: Compilers or different Embedded Systems can be considered of Semi-Detached type.

3. **Embedded** – A software project requiring the highest level of complexity, creativity, and experience requirement falls under this category. Such software requires a larger team size than the other two models and also the developers need to be sufficiently experienced and creative to develop such complex models.

$$\text{Effort (E)} = a_b(\text{KLOC})^{b_b}$$

$$\text{Deployment (D)} = c_b(E)^{d_b}$$

$$\text{Productivity (P)} = E/D$$

Where, a_b , b_b , c_b , d_b are called coefficients

PROJECT	a_b	b_b	c_b	d_b
Organic Mode				
Semi-Detached Mode				
Embedded Mode				

Ex 7. DESIGN

Date	19/04/2022
Submitted By	Aishwarya Lakshmi A P - 66 Amruthaa S - 73 Grace Hephzibah M - 84
Title/ Role	Stock Market Prediction

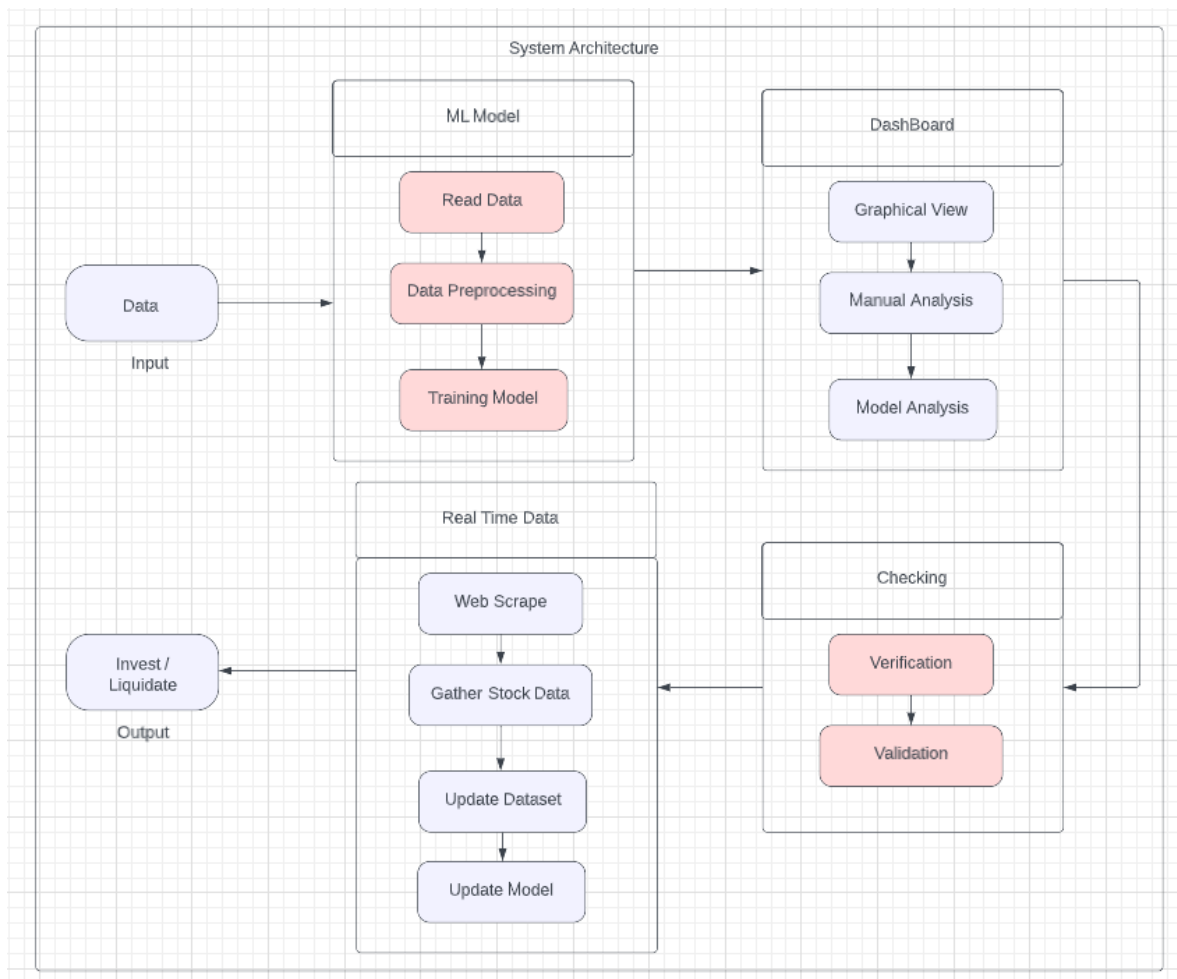


7.1 SYSTEM ARCHITECTURE

System Architecture is an abstract, conceptualization-oriented, global, and focused to achieve the mission and life cycle concepts of the system.

The purpose of system architecture is to define a comprehensive solution based on principles, concepts, and properties logically related to and consistent with each other.





System Design



Ex 8. MODELING UML USE CASE DIAGRAMS & CAPTURING USE CASE SCENARIOS

Date	26/04/2022
Submitted By	Aishwarya Lakshmi A P - 66 Amruthaa S - 73 Grace Hephzibah M - 84
Title/ Role	Stock Market Prediction



8.1 Use Case Diagram Description

USE CASE DIAGRAM: The purpose of a use case diagram in UML is to demonstrate the different ways that a user might interact with a system.

USE CASE SYMBOLS AND NOTATION: The notation for a use case diagram is pretty straightforward and doesn't involve as many types of symbols as other UML diagrams.

1] **SYSTEM:** A specific sequence of actions and interactions between actors and the system. A system may also be referred to as a scenario.

2] **USE CASES:** Horizontally shaped ovals that represent an action which accomplishes some sort of task within the system.

3] **ACTORS:** Stick figures that represent the people actually employing the use cases. It should be placed outside the system. There are two types of Actors namely:

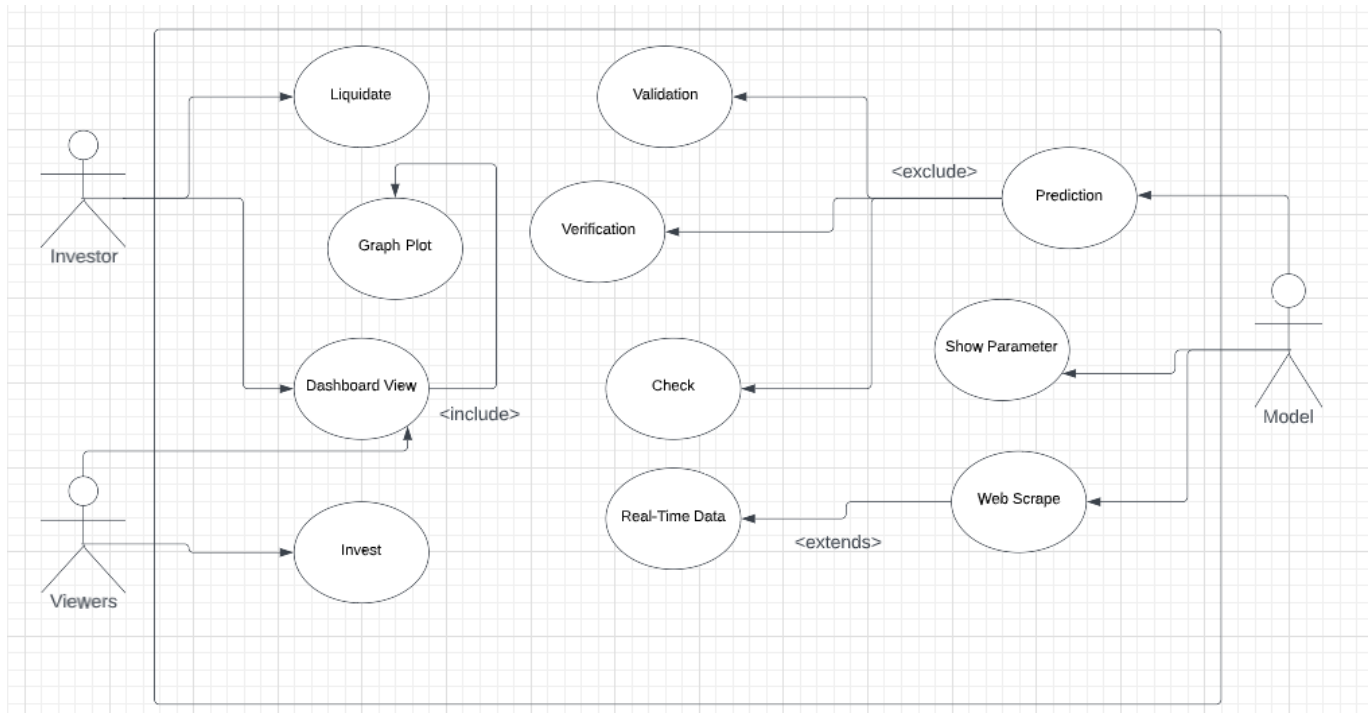
PRIMARY ACTOR: Initiates the use of the system. It should be placed on the left side of the system.

SECONDARY ACTOR: It is more reactionary and should be placed on the right side of the system.

4] RELATIONSHIPS: INCLUDE: This shows the dependency between the base and included use case (it happens every time). EXTENT: This happens only when certain criteria are met.

5] ASSOCIATION: A-line between actors and use cases. In complex diagrams, it is important to know which actors are associated with which use cases.

8.2 Use Case Diagram



Ex 9. ER Modeling From The Problem Statement

Date	03/05/2022
Submitted By	Aishwarya Lakshmi A P - 66 Amruthaa S - 73 Grace Hephzibah M - 84
Title/ Role	Stock Market Prediction



9.1 ER Modeling Description

Entity Relationship Diagram:

1. An Entity-Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system.
2. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research.

Uses of ER Diagram:

1. Database design
2. Database troubleshooting
3. Business information systems
4. Business process re-engineering (BPR)
5. Education Research

Components of ER Diagram:

ER Diagrams are composed of entities, relationships (Cardinality) and attributes. They also depict cardinality, which defines relationships in terms of numbers.

1. Entity

A definable thing—such as a person, object, concept or event—that can have data stored in it.

2. Attributes

A property or characteristic of an entity.







3. Keys

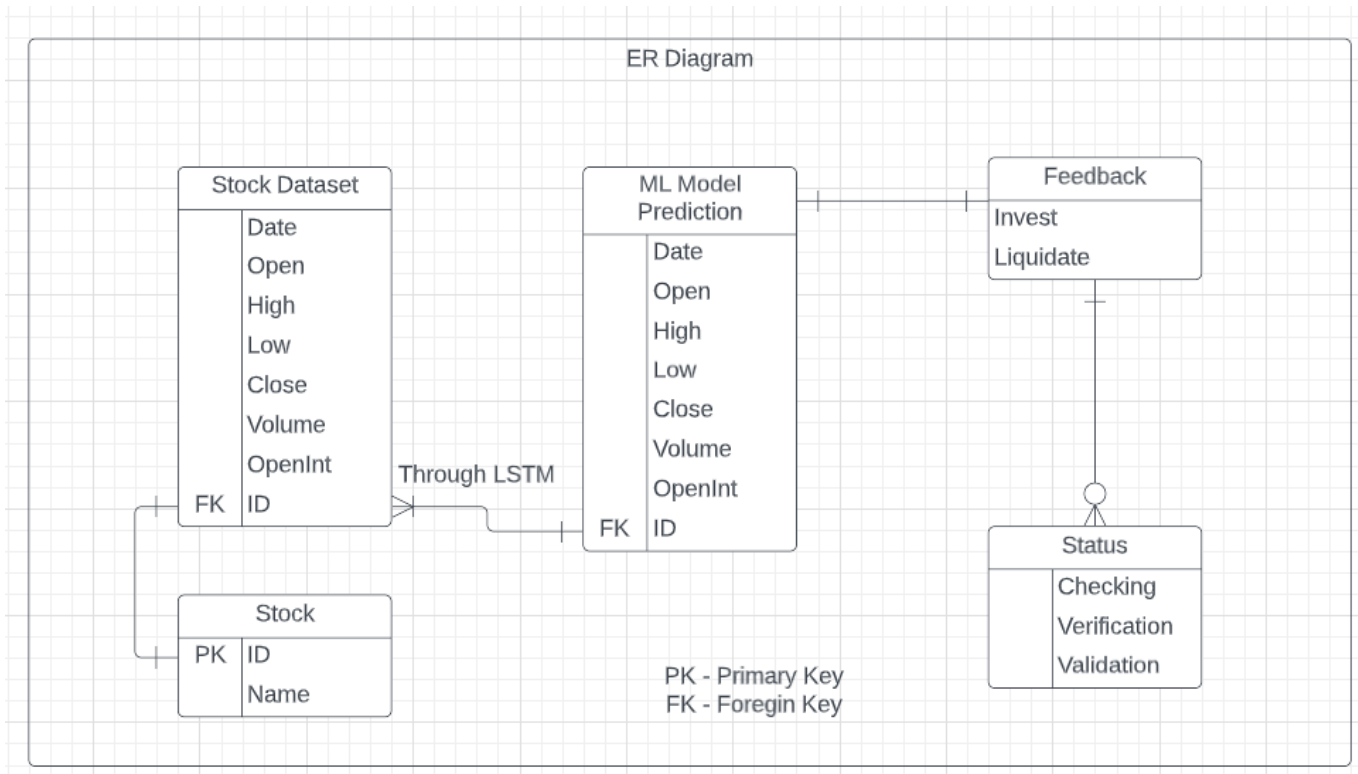
PRIMARY KEY(PK): It is unique, cannot be repeated and is never null.

FOREIGN KEY(FK): It is not unique and can be repeated.

4. Cardinality

Defines the numerical attributes of the relationship between two entities.

	One
	Many
	One (and only one)
	Zero or one
	One or many
	Zero or many



In this diagram, Stock Dataset, ML Model Prediction, Feedback, Stock and Status are the entities.

Ex 10. IDENTIFYING DOMAIN CLASSES

FROM THE PROBLEM STATEMENTS

Date	10/05/2022
Submitted By	Aishwarya Lakshmi A P - 66 Amruthaa S - 73 Grace Hephzibah M - 84
Title/ Role	Stock Market Prediction



10.1 DOMAIN CLASS DIAGRAM DESCRIPTION

Domain Class Diagram:

Domain Class Diagram provides an overview of the target system by describing the objects and classes inside the system and the relationships between them. It provides a wide variety of usages; from modelling the domain-specific data structure to the detailed design of the target system.

Domain Classes and Objects:

Objects that represent domain entities are called entities or domain objects. The classes they instantiate are called domain classes.

Identifying Stereotypes:

Typical domain class stereotypes include:

- <<thing>> = an entity that has mass and volume. Eg: Person, Report
- <<event>> = an entity that has a start time and duration. Eg: Transaction
- <<role>> = an entity that executes tasks. Eg: Customer, Doctor, Student
- <<type>> = an entity that describes other entities. Eg: Genre, ProductType, Rank

IDENTIFYING ATTRIBUTES:

A class contains definitions of all of the attributes its instances will contain.

An attribute has four attributes:

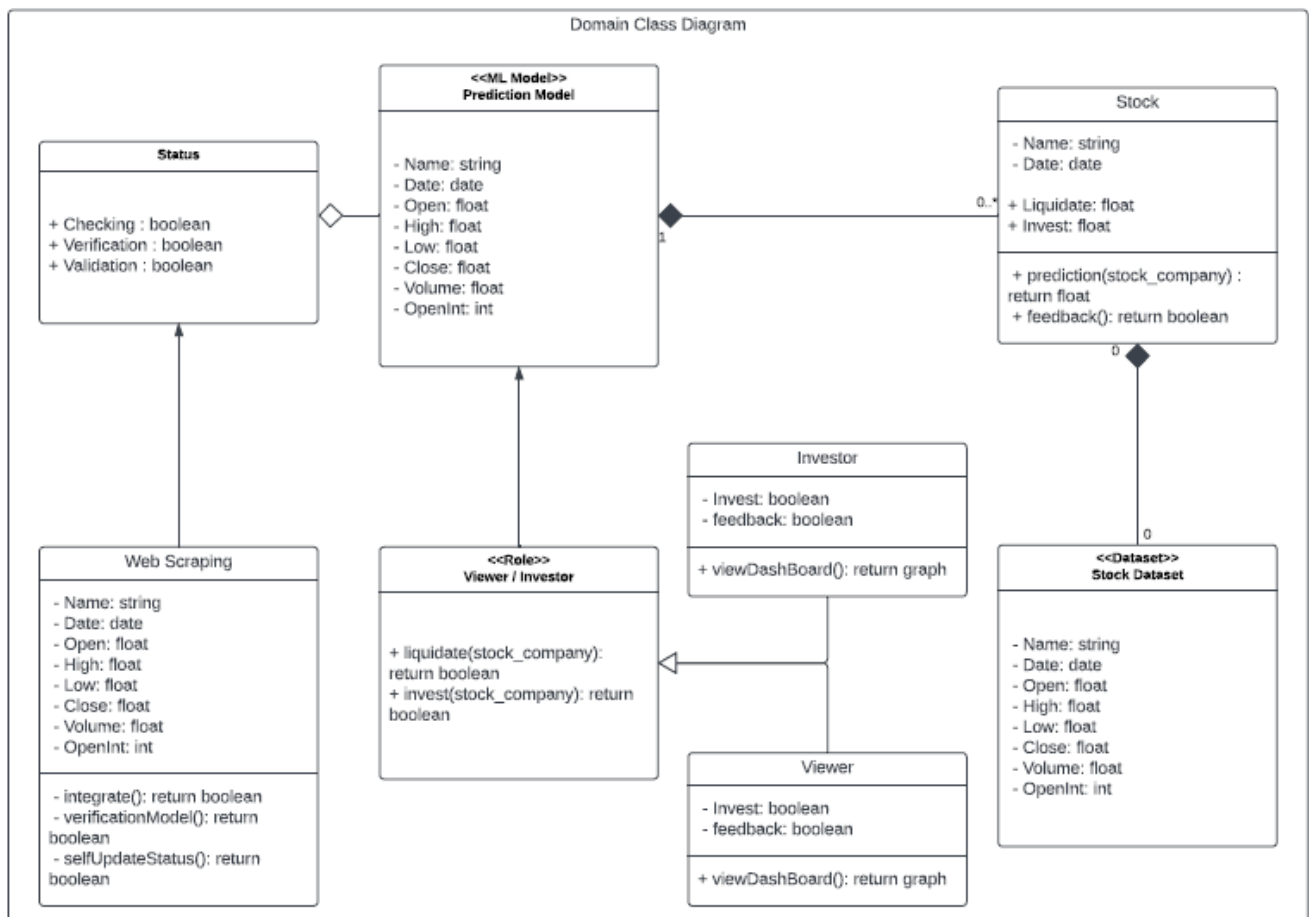
1] Name

2] Type

3] Visibility

- Private (-)
- Public (+)
- Protected (#)
- Package/Default (~)

4] Initial value (optional)



Ex 11. Statechart and Communication Modeling

Date	17/05/2022
Submitted By	Aishwarya Lakshmi A P - 66 Amruthaa S - 73 Grace Hephzibah M - 84
Title/ Role	Stock Market Prediction



11.1 Statechart Diagram Description

StateChart Diagram:

A Statechart diagram describes a state machine. The state machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events.

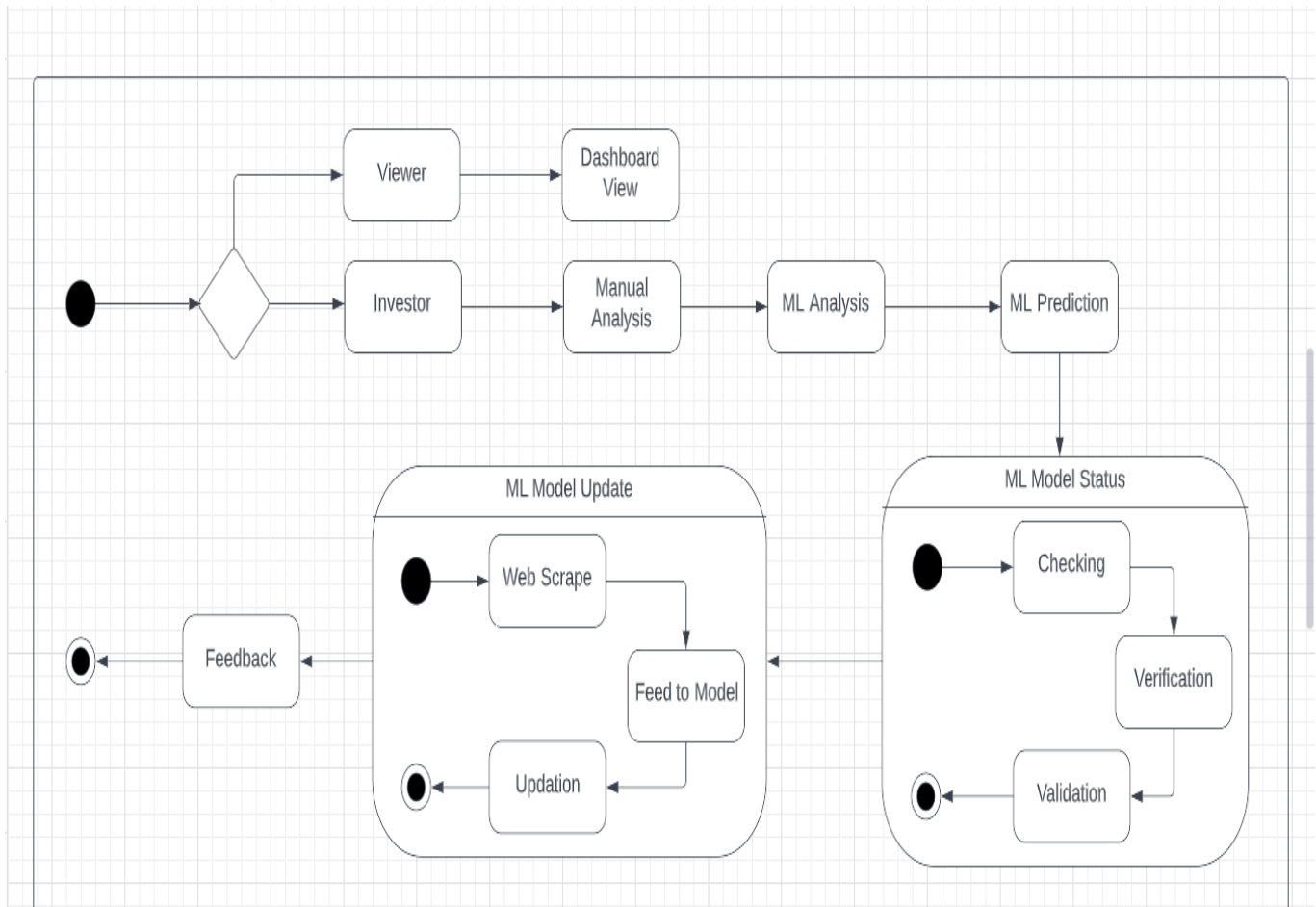
Components of a StateChart Diagram:

We can include many different shapes in a state diagram, particularly if we choose to combine it with another diagram. This list summarizes the most common shapes we may encounter.

- **Initial state** – We use a black filled circle to represent the initial state of a System or a class.
- **Transition** – We use a solid arrow to represent the transition or change of control from one state to another. The arrow is labelled with the event which causes the change in state.

- **State** – We use a rounded rectangle to represent a state. A state represents the conditions or circumstances of an object of a class at an instant in time.
- **Fork** – We use a rounded solid rectangular bar to represent a Fork notation with incoming arrows from the parent state and outgoing arrows towards the newly created states. We use the fork notation to represent a state splitting into two or more concurrent states.
- **Join** – We use a rounded solid rectangular bar to represent a Join notation with incoming arrows from the joining states and outgoing arrows towards the common goal state. We use the join notation when two or more states concurrently converge into one on the occurrence of an event or events.
- **Self transition** – We use a solid arrow pointing back to the state itself to represent a self transition. There might be scenarios when the state of the object does not change upon the occurrence of an event. We use self transitions to represent such cases.
- **Composite state** – We use a rounded rectangle to represent a composite state also. We represent a state with internal activities using a composite state.
- **Final state** – We use a filled circle within a circle notation to represent the final state in a state machine diagram.

11.2 Statechart Diagram



11.3 Communication Diagram Description

Communication Diagram:

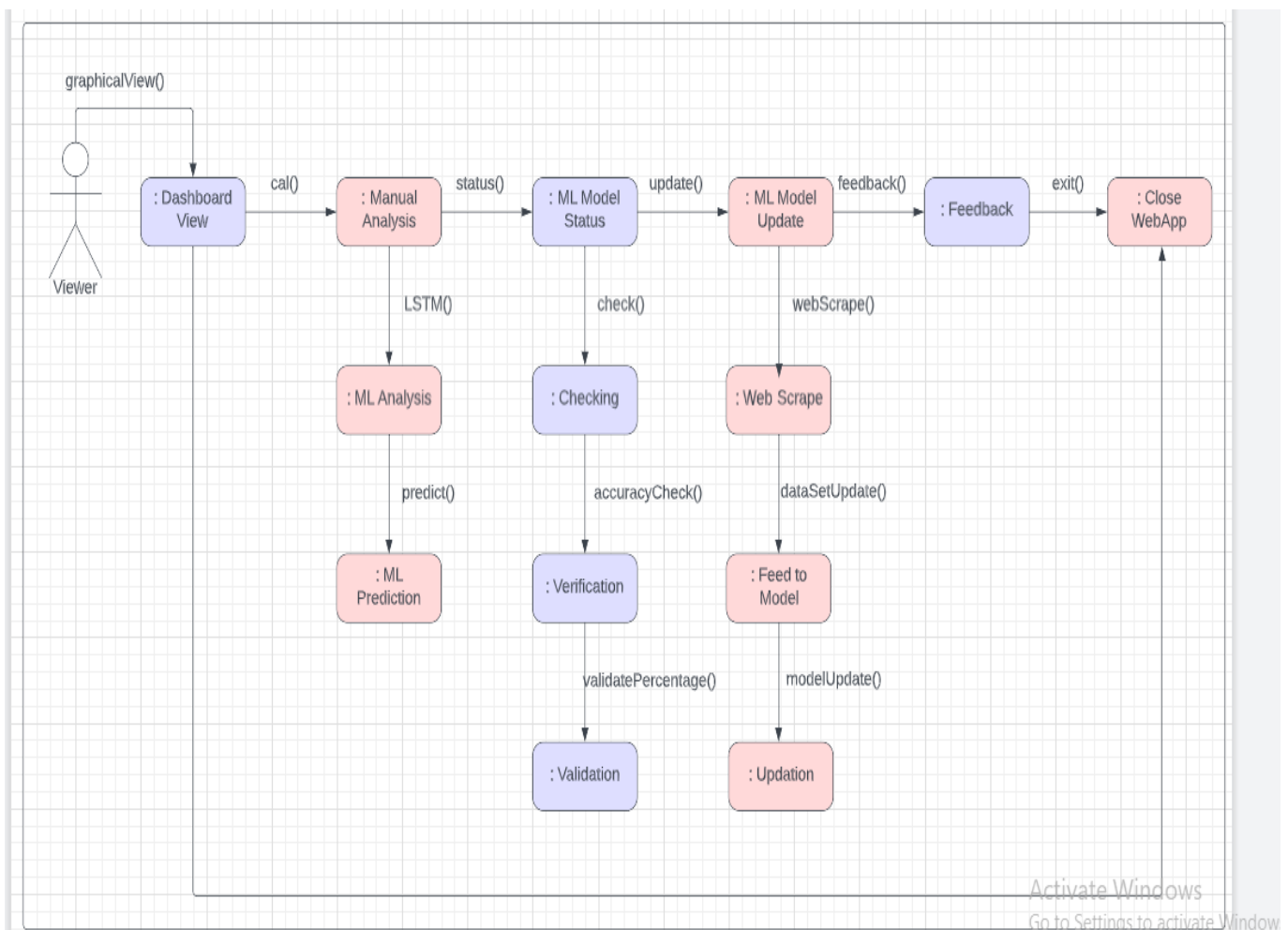
A Communication diagram models the interactions between objects or parts in terms of sequenced messages. Communication diagrams represent a combination of information taken from Class, Sequence, and Use Case Diagrams describing both the static structure and dynamic behaviour of a system.

Components of communication diagram:

- **Objects:** Objects can be classed as either a supplier or a client. Suppliers call the function that supplies the message. The client sends the message to the supplier, who receives it. It is represented by a rounded rectangle.

- **Actors:** Stick figure represents the actor. It is the instances that invoke the interaction. Each actor has a specific name and a role.
- **Links:** A straight line connecting two objects indicates a relationship between them. Two objects that are able to send messages to each other.
- **Messages:** Typically, messages will have a number and description next to them. The number determines the order in which messages should be read.

11.4 Communication Diagram



Ex 12. MODELING UML CLASS AND SEQUENCE DIAGRAM

Date	24/05/2022
Submitted By	Aishwarya Lakshmi A P - 66 Amruthaa S - 73 Grace Hephzibah M - 84
Title/ Role	Stock Market Prediction



12.1 CLASS DIAGRAM DESCRIPTION

CLASS DIAGRAM:

Class diagrams are one of the most useful types of diagrams in UML as they clearly map out the structure of a particular system by modelling its classes, attributes, operations, and relationships between objects.

COMPONENTS OF CLASS DIAGRAM:

The standard class diagram is composed of three sections:

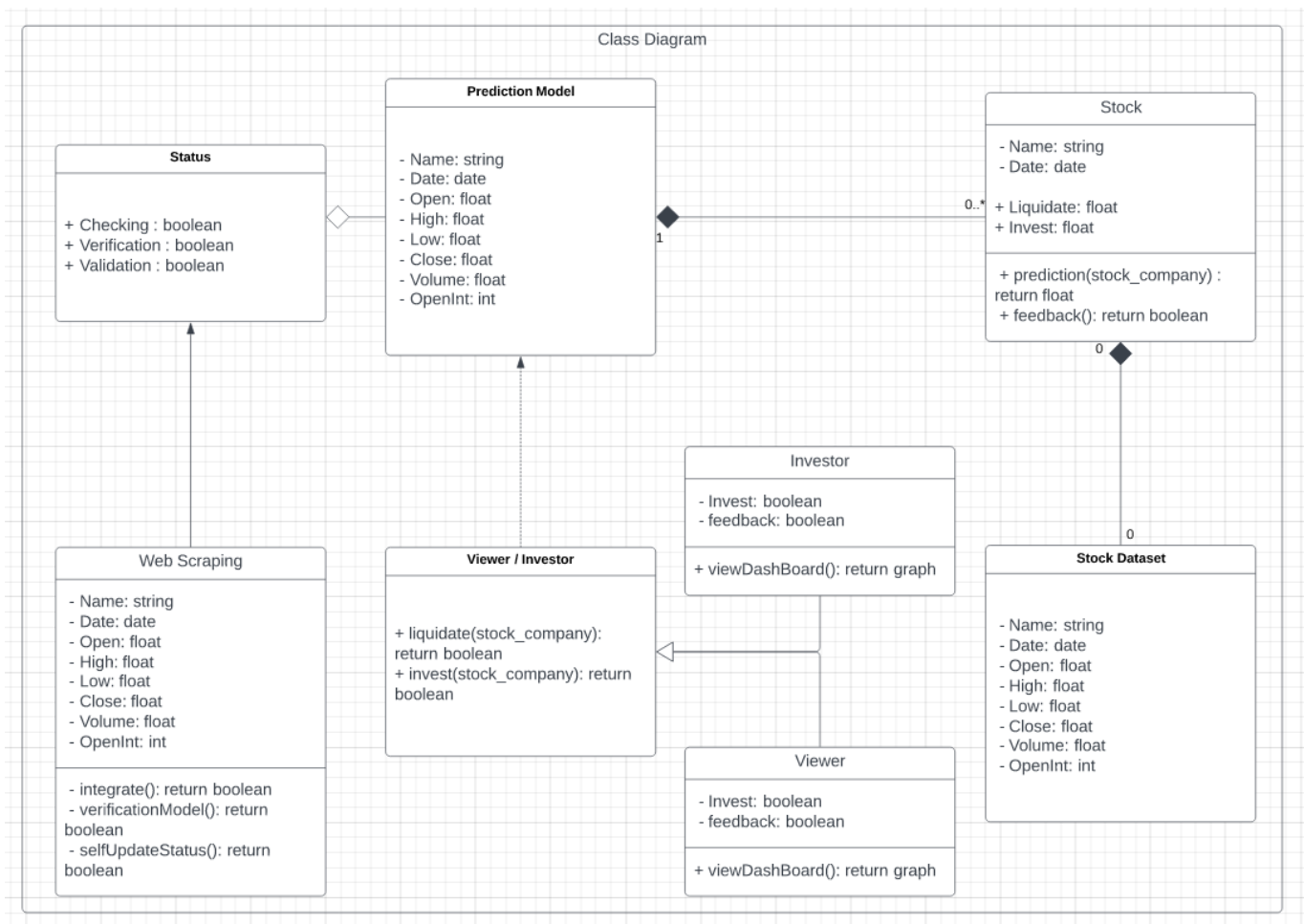
- **UPPER SECTION:** Contains the name of the class. This section is always required, whether you are talking about the classifier or an object.
- **MIDDLE SECTION:** This contains the attributes of the class. Use this section to describe the qualities of the class. This is only required when describing a specific instance of a class.
- **BOTTOM SECTION:** Includes class operations (methods). Displayed in list format, each operation takes up its own line. The operations describe how a class interacts with data.

MEMBER ACCESS SPECIFIERS:

All classes have different access levels depending on the access modifier (visibility). Here are the access levels with their corresponding symbols.

- Public (+)
- Private (-)
- Protected (#)
- Package (~)

12.2 CLASS DIAGRAM



12.3 SEQUENCE DIAGRAM DESCRIPTION

SEQUENCE DIAGRAM:

Sequence diagrams are a popular dynamic modelling solution in UML because they specifically focus on lifelines or the processes and objects that live simultaneously, and the messages exchanged between them to perform a function before the lifeline ends.

COMPONENTS IN SEQUENCE DIAGRAM:

1] ACTOR:

Stick figure represents the actor. Shows entities that interact with the external objects of the system.

2] OBJECTS:

Rectangular boxes represent the object and demonstrate how an object will behave in the context of the system.

3] ACTIVATION BOXES:

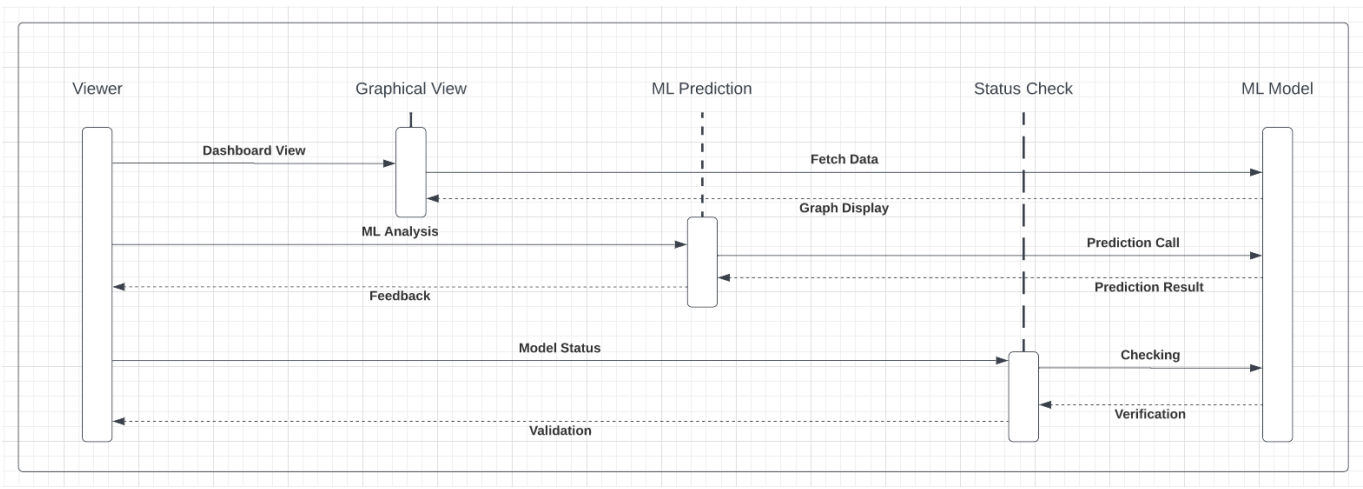
Represents the time needed for an object to complete a task. The longer the task will take, the longer the activation box becomes.

4] MESSAGE SYMBOLS:

We use the following arrows and message symbols to show how information is transmitted between objects. These symbols may reflect the start and execution of an operation or the sending and reception of a signal.

- **SYNCHRONOUS MESSAGE:** Represented by a solid line with a solid arrowhead. This symbol is used when a sender must wait for a response to a message before it continues. The diagram should show both the call and the reply.
- **ASYNCHRONOUS MESSAGE:** Represented by a solid line with a lined arrowhead. Asynchronous messages don't require a response before the sender continues. Only the call should be included in the diagram.
- **REPLY MESSAGE:** Represented by a dashed line with a lined arrowhead, these messages are replies to calls.
- **DELETE MESSAGE:** Represented by a solid line with a solid arrowhead, followed by an X. This message destroys an object.

12.4 SEQUENCE DIAGRAM



Ex 13. MODELING DATA FLOW DIAGRAM

Date	31/05/2022
Submitted By	Aishwarya Lakshmi A P - 66 Amruthaa S - 73 Grace Hephzibah M - 84
Title/ Role	Stock Market Prediction



13.1 DATA FLOW DIAGRAM DESCRIPTION

DATA FLOW DIAGRAM:

DFD graphically represents the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system.

COMPONENTS OF DATA FLOW:

The 4 basic symbols are used to represent a data-flow diagram.

1] PROCESS:

Rounded rectangle represents the process, which receives input data and produces output with different content or form. Processes can be as simple as collecting input data and saving it in the database, or they can be complex as producing a report. Every process has a name that identifies the function it performs.

2] DATA FLOW:

A data flow is a path for data to move from one part of the information system to another.

- Straight lines with incoming arrows are input data flow.
- Straight lines with outgoing arrows are output data flow.

3] DATA STORE:

A data store or data repository is used in a data-flow diagram to represent a situation when the system must retain data because one or more processes need to use the stored data later.

Data can be written into the data store, depicted by an outgoing arrow. Data can be read from a data store, depicted by an incoming arrow.

4] EXTERNAL ENTITY:

A rectangle represents an external entity. They are components outside of the boundaries of the information systems that either supply or receive data but does not process data. They represent how the information system interacts with the outside world.

External entities also are called **terminators** because they are data origins or final destinations.

An external entity must be connected to a process through a data flow.

13.2 DATA FLOW DIAGRAM

