Plutos Equities

Detailed Design Report

Your Gateway to NASDAQ: Intelligent Capital Forecasting

Presented by:

Emre Akgül, Uygar Aras, Asım Adil Can, Gün Taştan, Alara Zeybek

04.03.2025

1.Introduction

1.1.Purpose of the system

Plutos Equities is a **financial forecasting and analytics platform** designed to predict the **quarterly financial performance** of the **top 100 NASDAQ-listed companies** [1]. The platform leverages **machine learning models, time-series forecasting, and sentiment analysis** to provide accurate projections of critical financial metrics [2] such as:

- **Revenue**
- **Net income**
- **Earnings per Share (EPS)**
- **Operating expenses**
- **Gross profit margins**

By utilizing structured data [3] (historical financial statements, stock indices, SEC filings) and unstructured data (news sentiment, earnings call transcripts, social media trends), Plutos Equities provides actionable insights for financial stakeholders [4], including:

- **Investors** (retail and institutional)
- **Financial analysts**
- **Auditors**
- **Corporate decision-makers**

The **primary goal** of the system is to **enhance financial transparency and provide early indicators** of company performance before official quarterly reports are released. Unlike existing tools that primarily focus on stock price movements, Plutos Equities aims to **forecast fundamental financial performance**, making it a **valuable resource** for long-term financial planning and risk assessment.

1.2.Design goals

Plutos Equities is designed with a strong emphasis on scalability, performance, reliability, security, and usability. The following design goals have been established to ensure the platform's effectiveness:

- **Usability:** The platform provides an **intuitive user interface (UI)** with clear **data visualizations and customizable dashboards** for different financial stakeholders.
- **Performance:** The system **processes large financial datasets efficiently**, leveraging **cloud computing and optimized machine learning pipelines** to deliver results quickly [5][6].
- **Reliability:** Plutos Equities ensures **99% uptime**, with redundant data backups and failover mechanisms to **maintain system stability**.
- **Marketability:** Designed to attract investors and analysts, the system includes **subscription-based financial insights**, making it a **viable B2B product**.
- **Extendibility:** Future integrations with **alternative data sources, sentiment analysis models, and custom financial metrics** are considered.
- **Security:** User authentication follows **OAuth 2.0**, and data is stored securely on **Azure SQL with encrypted access** [7].

- **Scalability:** The platform supports **hundreds of concurrent users**, with automatic load balancing via **Azure cloud services**.
- **Maintainability:** The system follows a **modular architecture**, making it easy to update individual components.
- **Flexibility:** Users can **customize forecasts** by selecting specific **financial indicators**.
- **Modularity:** The architecture is divided into **Presentation, Control, and Business Logic layers**, ensuring **clear separation of concerns**.
- **Aesthetics:** The UI is **modern, dark-themed**, and optimized for both **desktop and mobile devices**.

1.3.Definitions, acronyms, and abbreviations

- **Plutos Equities:** Financial forecasting and capital analysis platform for NASDAQ companies.
- **EPS (Earnings Per Share):** An indicator of a company's profitability.
- **ML (Machine Learning):** A field of artificial intelligence that enables systems to learn from data.
- **LSTM (Long Short-Term Memory):** A type of recurrent neural network for time-series forecasting.
- **SEC (Securities and Exchange Commission):** U.S. regulatory body overseeing securities markets.
- **Sentiment Analysis:** A method of determining the emotional tone in financial reports, news, and earnings calls.
- **B2B (Business-to-Business):** Refers to the target market for Plutos Equities, offering services to businesses rather than individual consumers.
- **OAuth 2.0:** Secure authorization standard for user authentication.
- **GDPR (General Data Protection Regulation):** Ensures data privacy compliance.
- **API (Application Programming Interface):** Used for fetching financial data.
- **MVP (Minimum Viable Product):** Refers to the earliest release version of Plutos Equities.

1.4.Overview

Plutos Equities **bridges the gap** between traditional **stock price prediction platforms** and **fundamental financial analysis**. The platform consists of the following **core components**:

- **Data Collection & Processing:**
  - Fetches **historical financial data** from sources such as **SEC EDGAR, Yahoo Finance, and Bloomberg**.
  - Analyzes **earnings calls, news sentiment, and market trends** using **NLP models**.
- **Machine Learning Models:**
  - Uses **Prophet, LSTM, and ensemble learning methods** to forecast financial metrics [1].
  - Computes **confidence intervals** for predictions.
- **User Interaction & Insights:**
  - Provides an **interactive dashboard** for financial professionals.
  - Users can **follow companies, export forecasts, and customize their financial insights**.
- **Security & Compliance:**
  - **User authentication via OAuth 2.0**.

        - **Data encryption and GDPR compliance** for secure financial data handling.
    - **Infrastructure & Scalability:**
        - Hosted on **Azure Cloud Services**.
        - Supports **real-time updates** and **multiple user tiers**.

2.Current software architecture

2.1.Competitors & Existing Solutions

Plutos Equities competes with existing financial forecasting platforms, but differentiates itself by focusing on fundamental financial forecasting rather than stock price movements.

Competitor Analysis:

Plutos Equities faces competition from several platforms that provide financial insights, stock market trends, and predictive analytics. However, most competitors focus primarily on technical stock price movement rather than fundamental financial forecasting.

- **Bloomberg Terminal**: Provides institutional-grade analytics but is prohibitively expensive and focused on price movements rather than quarterly performance predictions [7[8].
- **AlphaSense**: Uses NLP-driven financial insights but lacks deep forecasting of key financial metrics like revenue and EPS.
- **TradingView**: Offers stock price forecasting and technical analysis tools but does not predict quarterly financial metrics.
- **Yahoo Finance Premium**: Provides market data and earnings calendars but lacks predictive financial analytics for earnings and revenue forecasting.

2.2.Alternative Solutions

- **Excel-based financial modeling:** Widely used by financial analysts, but manual modeling is **time-consuming, prone to human error, and lacks scalability**.
- **Hedge fund proprietary models:** Exclusive AI-driven financial models used by institutional investors, but they are **not publicly available**.
- **Basic ML-based forecasting scripts:** Requires programming knowledge, making it **unusable for non-technical financial professionals**.

2.3.Current Solutions and Gaps

Plutos Equities fills the gap in the market by combining AI-driven forecasting, sentiment analysis, and financial modeling in a single platform. Unlike stock price prediction tools, it offers holistic insights into a company's financial health and future earnings potential, enabling smarter investment and strategic decision-making.

3.Proposed software architecture

3.1.Overview

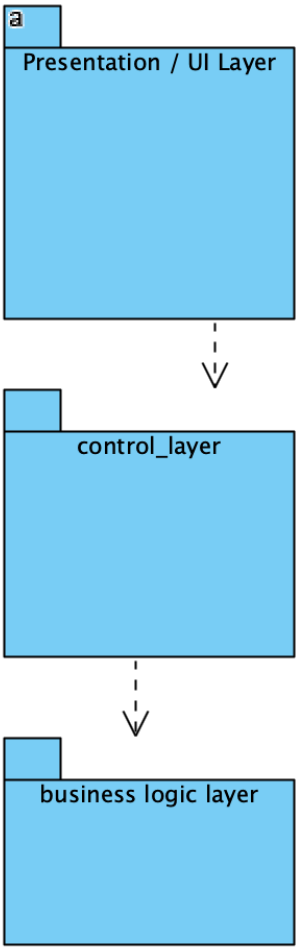

Figure 1: High Level Subsystem Decomposition

In this section, the software architecture and subsystem decomposition of the Plutos Equities forecasting platform are explained in detail. Initially, we provide a clear explanation of how the system is organized into distinct layers: Presentation/UI Layer, Control Layer, and Business Logic Layer, illustrated through diagrams and class structures. Then, the hardware/software mapping of the platform is presented, clarifying how the application's components are allocated across various hardware resources and cloud-based infrastructure. The persistent data management segment follows, describing how data storage, retrieval, and object management are handled within Azure SQL and Azure Storage services.

3.2.Subsystem decomposition

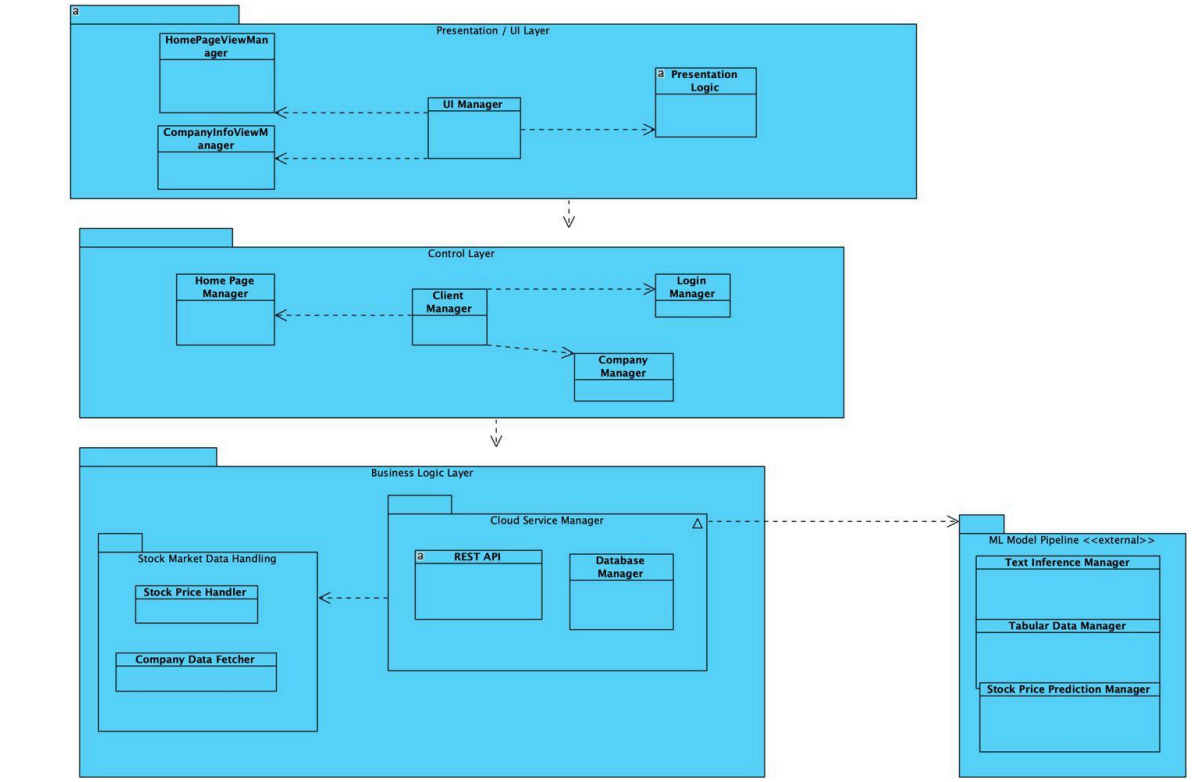

Figure 2: Detailed Subsystem Decomposition

Presentation/UI Layer:

This layer is responsible for the interaction with the end-user and acts as the entry point to the platform. It manages how users access the forecasting services, visualizes forecasts, and presents insights intuitively. All user requests are received and processed initially at this layer. Specifically, the HomePageViewManager handles the overall layout and initial interactions users have with the application's interface, whereas the CompanyInfoViewManager specializes in managing views and interactions related to specific company data. The UI Manager coordinates these views to ensure consistent UI behavior and user experience. Additionally, the Presentation Logic component supports rendering logic, facilitating a responsive and seamless user experience.

Control Layer:

This layer orchestrates requests between user interfaces and business logic. This layer incorporates components like the HomePageManager, Client Manager, Login Manager, and Company Manager, providing session management, authentication, and user-specific interactions. The Client Manager is a central component, delegating authentication tasks to the Login Manager and handling company-related requests with the Company Manager.

Business Logic Layer:

implements the main functionalities and hosts the REST API. Within this layer, the Cloud Service Manager integrates external cloud-based resources such as storage services and database operations through a RESTful API. This subsystem ensures smooth communication between data repositories, databases, and the machine learning infrastructure. It specifically manages the flow of requests to the ML Model Pipeline, an external component consisting of the Text Inference Manager, Tabular Data Manager, and Stock Price Prediction Manager. The ML Model Pipeline subsystem leverages advanced natural language processing and time-series analysis models to deliver precise forecasting data to the business logic layer. After processing and consolidating the results, the Cloud Service Manager sends data back through RESTful APIs to the UI for visualization and user interaction.

Finally, all user, analytical, and operational data is persisted using the **Persistence Layer**, consisting of robust database solutions and object storage, ensuring reliable data storage and retrieval capabilities.

3.3.Hardware/software mapping

Plutos is a medium-scale, web-based financial forecasting application that merges advanced machine learning techniques with textual sentiment analysis. The application does not necessitate extensive computational power, but requires efficient and reliable cloud-based resources for data storage, processing, and deployment of predictive models.

To ensure optimal performance, scalability, and data integrity, we utilize Azure cloud services specifically tailored to our application's requirements:

Azure SQL Database: We employ Azure SQL for structured data management. It supports storing historical financial data, which feeds our time-series forecasting models. Azure SQL offers robust performance, reliable disaster recovery with geo-replication, and ensures high availability and security, essential for sensitive financial data.

Description of used service: Single Database, vCore, General Purpose, Provisioned, Standard-series (Gen 5), Primary or Geo replica Disaster Recovery, Locally Redundant, 1 - 2 vCore Database(s) x 200 Hours, 32 GB Storage, SQL License (Pay as you go), LRS Backup Storage Redundancy, 0 GB Point-In-Time Restore, 0 x 5 GB Long Term Retention

Estimated Monthly Price: $105.66

Azure Storage Accounts: Azure Storage Accounts handle extensive unstructured and semi-structured data, including financial documents, earnings reports, and associated media used in natural language processing (NLP). Using Block Blob storage with Hot Access tier ensures low latency and high throughput access, critical for processing large volumes of textual data efficiently.

Description of used service: Block Blob Storage, General Purpose V2, Flat Namespace, LRS Redundancy, Hot Access Tier, 1,000 GB Capacity - Pay as you go, 10 x 10,000 Write operations, 10 x 10,000 List and Create Container Operations, 10 x 10,000 Read operations, 1 x 10,000 Other operations. 1,000 GB Data Retrieval, 1,000 GB Data Write, SFTP enabled for 730 Hours

Estimated Monthly price: $240.84

Azure Machine Learning: Our predictive models, including transformer-based architectures (such as BERT) and recurrent neural networks (such as LSTM), are deployed and managed using Azure Machine Learning. Azure Machine Learning simplifies the end-to-end lifecycle of model training, evaluation, and deployment, providing robust infrastructure optimized for machine learning workloads.

Description of used service: 1 D2ds v4 (2 Core(s), 8 GB RAM) x 730 Hours, Pay as you go

Estimated Price: 82.49

Plutos' Azure-based infrastructure costs approximately $428 per month and is designed to comfortably support our medium-scale user base, expected to handle several hundred concurrent users efficiently. The Azure SQL and Azure Storage services operate continuously to ensure consistent data availability, while Azure Machine Learning resources are primarily utilized during model training and periodic retraining activities, optimizing resource usage and cost-effectiveness. Azure's flexible architecture allows us to scale computing resources easily to accommodate future growth or additional workloads.

Users only need a device with internet access and a modern web browser, capable of supporting HTML5, CSS3, and ES6, to interact seamlessly with our web-based platform.

3.4.Persistent data management

Given the high volume and variety of financial data—both structured and unstructured—that Plutos Equities will manage, the project requires a robust, efficient, and scalable data management solution. The data management strategy will combine relational databases, cloud object storage solutions, and real-time data retrieval pipelines to handle datasets and user data effectively.

Relational Database Management (MySQL):
Structured data, such as user profiles, subscription details, prediction outputs, user preferences, and financial data metrics, will be stored in a relational database. Due to frequent updates and changes in prediction outputs, the system will implement efficient indexing, normalization, and optimized querying strategies to enhance data retrieval performance and maintain low-latency responses, even during peak usage periods.

Cloud Object Storage (Azure Blob Storage):
Azure Blob Storage will handle large volumes of unstructured data, including historical financial reports, earnings call transcripts, news articles, market sentiment data, and backups of model outputs. Azure Blob Storage ensures high scalability, data redundancy, and seamless integration with other Azure services, strengthening data management capabilities.

Real-time Data Pipeline (Azure Infrastructure):
The project will implement a real-time data processing pipeline using Azure services such as Azure Event Hubs and Azure Functions. Data from market sources, including financial APIs and news feeds, will be ingested in real-time through Azure Event Hubs. Azure Functions will

preprocess this incoming data by applying transformations, feature extraction, and sentiment analysis [8] before persisting it in MySQL and Azure Blob Storage. This pipeline guarantees that user predictions and analyses always reflect the most current market conditions.

Security and Compliance:
All data management practices adhere to stringent data privacy and security standards such as GDPR.

3.5.Access control and security

In our application, we have two primary user roles: Administrators and Customers. Each role has distinct access permissions tailored specifically to their responsibilities and interactions within the system.

The Administrators possess comprehensive access to critical data, including financial records, system configurations, user management, and detailed performance logs. Only administrators can modify, manage, or delete sensitive information stored in the Azure SQL Database and Azure Storage. Furthermore, administrative actions are monitored and logged for security auditing purposes.

In contrast, Customers have limited and strictly controlled access. They can view stock predictions, company performance forecasts, and relevant historical stock data through the user-friendly web interface. However, customers cannot access, modify, or delete any sensitive or administrative-level data. They interact only with preprocessed and securely delivered forecast results.

To ensure robust security, all users must authenticate themselves through a secure login system employing industry-standard encryption methods. Communication between the client browser and our platform is protected using HTTPS protocol. User sessions are securely maintained, and sensitive user data remains confidential. The system ensures data privacy by preventing customers from accessing other customers' data or any administrative functionality.

4.Subsystem services

4.1.Client

The Client subsystem manages all functionalities directly accessible by the users through the website interface. It provides essential operations including user interactions, managing user requests, and presenting data retrieved from the server. This subsystem is structured into two main components: the Control Layer and the User Interface Layer. Essential operations include processing user inputs such as forecasting queries, signup and login requests, subscriptions, and dashboard customizations in the Control Layer, while the Presentation Layer is responsible for visualizing forecast results through intuitive dashboards, interactive charts, and clear data summaries, ensuring users can easily interpret financial predictions regardless of their expertise.

4.1.1.UI Layer

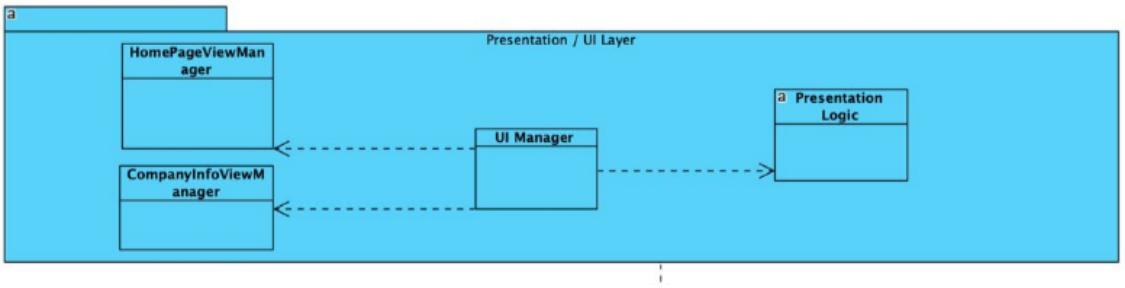

Figure 3: UI Layer High Level Design

The UI layer subsystem manages visual interactions between user and website.

UI Manager: Manages the overall design and usability of the interface, ensuring consistency and user-friendliness.

Homepage View Manager: Handles displaying and updating content on the homepage to keep it relevant and engaging.

Company Info Manager: Ensures detailed information about companies is clearly and neatly presented for the user.

Presentation Logic: Controls the logic behind presenting visual elements, enhancing usability and interaction clarity.

4.1.2.Control Layer

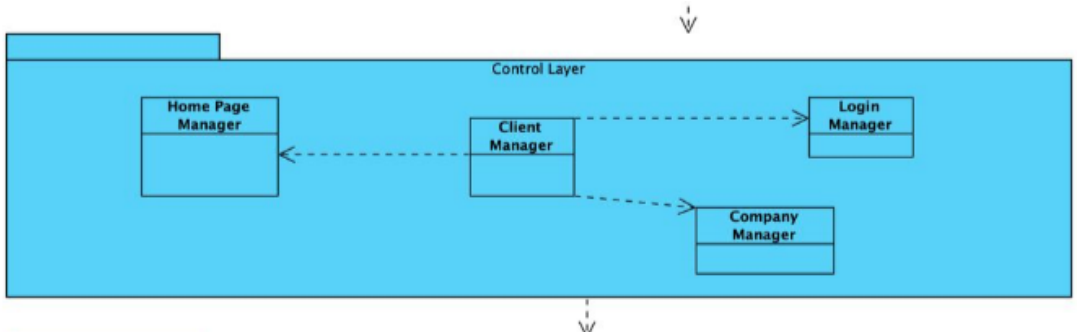

Figure 4: Control Layer High Level Design

The control subsystem manages the client's operations and facilitates communication between the client and server.

Client Manager: Oversees user sessions and interactions, keeping the platform responsive and secure.

Homepage Manager: Handles backend tasks to regularly update and refresh homepage content.

Company Manager: Responsible for accurately retrieving and updating company data on the client side.

Login Manager: Manages secure user logins, authentication, and session handling.

4.2.Server

The Server subsystem manages backend functionalities, handling data processing, integration of structured and unstructured datasets, execution of machine learning models, and generating accurate financial forecasts. It ensures real-time updates, maintains data integrity, and efficiently responds to client requests by delivering timely predictions back to the Client subsystem. It includes the business logic layer, ML model pipeline layer and data layer.

4.2.1.Business Logic Layer

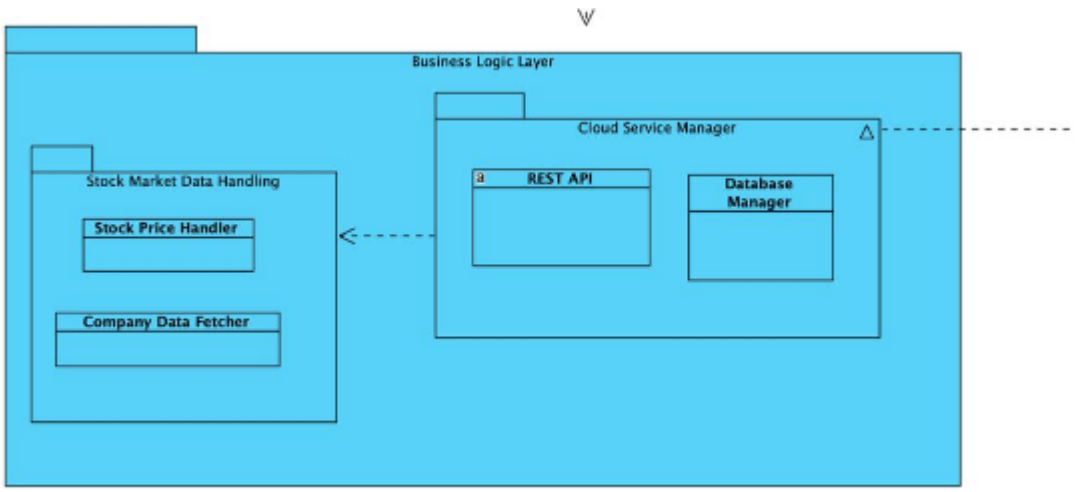

Figure 5: Business Logic Layer High Level Design

Business logic layer processes data according to the rules and requirements of the application. It acts as an intermediary, enforcing logic between the user interface and data storage layers.

Cloud Server Manager: Manages cloud-based infrastructure and operations, ensuring smooth communication between server and client while protecting data integrity.

REST API: Facilitates standard, efficient data exchange between client and server.

Database Manager: Handles secure data storage and retrieval, ensuring accuracy and consistency.

Stock Market Data Manager: Collects and manages stock market data, ensuring accurate and timely integration into forecasts.

4.2.2.ML Model Pipeline Layer

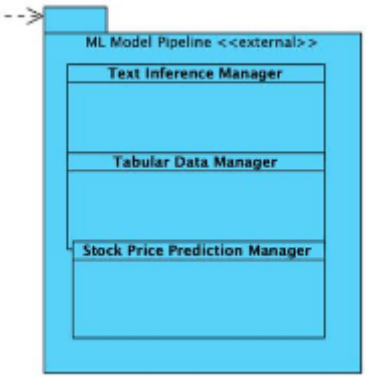

Figure 6: ML Pipeline High Level Design

Machine learning model pipeline layer is an external subsystem that processes, stores and manages large amounts of data required for the machine learning pipeline.

Textual Data Manager: Collects and preprocesses text data from financial news and reports for sentiment analysis.

Tabular Data Manager: Manages structured data used for machine learning pipeline.

Prediction Manager: Oversees running, monitoring, and validating machine learning models to provide accurate and timely predictions.

4.2.3.Data Layer

Data layer is the lowest layer of the system. Data layer is responsible for the persistent and organized storage of the data. Data stored in this layer is used by the user through control of the business layer.

5.Test Cases

5.1 Functional Test Cases

T-01 – Validate User Registration with Invalid Input

- **Test Type/Category:** Functional / Integration
- **Objective:** Ensure registration fails when invalid or missing data is provided.
- **Procedure:**
    1. Navigate to the Sign-Up page.
    2. Enter invalid data (e.g., already registered email, improperly formatted email, or leave required fields blank).
    3. Click “Sign Up”.
- **Expected Outcome:** Registration fails and an error message is displayed.
- **Priority:** Critical

T-02 – Validate User Registration with Valid Input

- **Test Type/Category:** Functional / Integration
- **Objective:** Confirm a new user is created when valid data is provided.
- **Procedure:**
    1. Navigate to the Sign-Up page.
    2. Enter valid registration information (unique email, strong password, all required fields).
    3. Click “Sign Up”.
    4. Verify that a confirmation email is sent.
- **Expected Outcome:** User is registered and a new database entry is created.
- **Priority:** Critical

T-03 – Validate Google Sign-Up Functionality

- **Test Type/Category:** Functional / Integration
- **Objective:** Ensure sign-up and login succeed using Google authentication.
- **Procedure:**
    1. On the Sign-Up page, click “Sign Up with Google”.
    2. Select a valid Google account and confirm permissions.
    3. Accept the terms and conditions.
- **Expected Outcome:** Registration and login succeed; a new user entry is created.
- **Priority:** Major

T-04 – Validate User Login with Invalid Credentials

- **Test Type/Category:** Functional / Integration
- **Objective:** Verify that login fails when incorrect credentials are provided.
- **Procedure:**
    1. Navigate to the Login page.
    2. Enter an incorrect email or password.
    3. Click “Sign In”.
- **Expected Outcome:** Login fails with an appropriate error message.
- **Priority:** Critical

T-05 – Validate User Login with Valid Credentials

- **Test Type/Category:** Functional / Integration
- **Objective:** Confirm successful login with correct credentials.
- **Procedure:**
    1. Navigate to the Login page.
    2. Enter valid email and password.
    3. Click “Sign In”.
- **Expected Outcome:** User is logged in and redirected to the Dashboard with personalized data.
- **Priority:** Critical

T-06 – Validate Forgotten Password Functionality

- **Test Type/Category:** Functional
- **Objective:** Ensure that the “Forgot Password” process sends a reset link.
- **Procedure:**
    1. Click the “Forgot Password” link on the Login page.
    2. Enter the registered email address.
    3. Submit the request.
- **Expected Outcome:** A password reset link is sent to the email.
- **Priority:** Major

T-07 – Validate User Logout Functionality

- **Test Type/Category:** Functional
- **Objective:** Verify that a user can log out successfully.
- **Procedure:**
    1. While logged in, click the “Logout” button on the Dashboard.
- **Expected Outcome:** User is logged out and redirected to the Login page.
- **Priority:** Critical

T-08 – Validate Dashboard Data Loading Post-Login

- **Test Type/Category:** Functional
- **Objective:** Check that personalized data (e.g., watchlist, recent predictions) is loaded after login.
- **Procedure:**
    1. Log in with valid credentials.
    2. Navigate to the Dashboard.
- **Expected Outcome:** Dashboard displays correct personalized financial data.
- **Priority:** Major

T-09 – Validate Navigation to Financial Predictions Page

- **Test Type/Category:** Functional / Integration
- **Objective:** Ensure users can navigate to the Predictions page.
- **Procedure:**
    1. From the Dashboard, click the “Financial Predictions” option in the navigation menu.
- **Expected Outcome:** Predictions page loads with the latest forecast data.
- **Priority:** Major

T-10 – Validate Retrieval of Financial Predictions for a Company

- **Test Type/Category:** Functional / Integration
- **Objective:** Verify accurate forecast data is displayed for a selected company.
- **Procedure:**
    1. On the Financial Predictions page, select a company (e.g., Apple Inc.).
    2. Observe the displayed data (revenue, EPS, gross margin, etc.).
- **Expected Outcome:** Accurate forecast data and associated metrics are displayed.
- **Priority:** Critical

T-11 – Validate Display of Confidence Intervals in Predictions

- **Test Type/Category:** Functional
- **Objective:** Ensure confidence intervals are displayed with each forecast metric.
- **Procedure:**
    1. On the Predictions page, review a company’s forecast data.
- **Expected Outcome:** Confidence intervals accompany each prediction.
- **Priority:** Major

T-12 – Validate Real-Time Update Notifications for Prediction Changes

- **Test Type/Category:** Functional / Integration
- **Objective:** Verify that changes in data trigger real-time notifications.
- **Procedure:**
    1. Simulate a change in financial data affecting predictions (using test data or simulation mode).
    2. Monitor for an alert on the device.
- **Expected Outcome:** A real-time notification is displayed.
- **Priority:** Major

T-13 – Validate Exporting Prediction Data as CSV

- **Test Type/Category:** Functional
- **Objective:** Confirm that predictions can be exported in CSV format.
- **Procedure:**
    1. On the Financial Predictions page, click “Export”.
    2. Select CSV format.
- **Expected Outcome:** A CSV file with correct prediction data is downloaded.
- **Priority:** Minor

T-14 – Validate Exporting Prediction Data as XLS

- **Test Type/Category:** Functional
- **Objective:** Confirm that predictions can be exported in XLS format.
- **Procedure:**
    1. On the Financial Predictions page, click “Export”.
    2. Select XLS format.
- **Expected Outcome:** An XLS file with correct prediction data is downloaded.
- **Priority:** Minor

T-15 – Validate User Profile Update Functionality

- **Test Type/Category:** Functional
- **Objective:** Verify that user profile information can be updated successfully.
- **Procedure:**
    1. Navigate to the Profile page.
    2. Click the “Edit” icon and modify profile information.
    3. Click “Save”.
- **Expected Outcome:** Profile is updated and a success message is shown.
- **Priority:** Major

T-16 – Validate Error Handling on Profile Update with Empty Fields

- **Test Type/Category:** Functional
- **Objective:** Ensure the system handles empty fields appropriately during profile update.
- **Procedure:**
    1. Navigate to the Profile page.
    2. Click “Edit” and clear all required fields.
    3. Click “Save”.
- **Expected Outcome:** An error message is displayed; profile is not updated.
- **Priority:** Major

T-17 – Validate Integration with Yahoo Finance API

- **Test Type/Category:** Functional / Integration
- **Objective:** Ensure accurate financial data is fetched using the Yahoo Finance API.
- **Procedure:**
    1. Trigger a backend request for a known company using the Yahoo Finance API (via yfinance).
- **Expected Outcome:** Accurate historical and current financial data is retrieved.
- **Priority:** Critical

T-18 – Validate Integration with yfinance Python Library

- **Test Type/Category:** Functional / Integration
- **Objective:** Confirm the yfinance library returns complete financial data.
- **Procedure:**
    1. Initiate a backend call using the yfinance library for a specific company.
- **Expected Outcome:** The library returns accurate and complete data.
- **Priority:** Critical

T-19 – Validate Integration with SEC EDGAR Data Retrieval

- **Test Type/Category:** Functional / Integration
- **Objective:** Verify that financial statements are retrieved from SEC EDGAR.
- **Procedure:**
    1. Trigger a process to fetch financial statements from SEC EDGAR for a company.
- **Expected Outcome:** Required statements are successfully retrieved and processed.
- **Priority:** Major

TE-20 – Validate Machine Learning Model Prediction Generation

- **Test Type/Category:** Functional / Integration
- **Objective:** Verify the ML model produces forecasted financial metrics.
- **Procedure:**
    1. Trigger the ML model prediction process for a company.
- **Expected Outcome:** Forecasted metrics (e.g., revenue, EPS) are generated and displayed.
- **Priority:** Critical

T-21 – Validate Real-Time Alerts for Prediction Threshold Breach

- **Test Type/Category:** Functional
- **Objective:** Check that alerts are triggered when metrics exceed thresholds.
- **Procedure:**
    1. Simulate a condition where a key metric exceeds its threshold (e.g., sudden revenue change).
    2. Monitor for an alert.
- **Expected Outcome:** A notification is sent promptly to the user's device.
- **Priority:** Major

T-22 – Validate Subscription Management Functionality

- **Test Type/Category:** Functional
- **Objective:** Ensure users can change their subscription plans successfully.
- **Procedure:**
    1. Navigate to the Subscription section in the user profile.
    2. Change the subscription plan.
    3. Confirm the change.
- **Expected Outcome:** Subscription is updated correctly in the account.
- **Priority:** Major

T-23 – Validate Payment Integration via Stripe (Simulated)

- **Test Type/Category:** Functional / Integration
- **Objective:** Verify that payment processing works as expected.
- **Procedure:**
    1. Navigate to the Subscription update page.
    2. Select a new plan and enter test payment details.
    3. Process the payment.
- **Expected Outcome:** Payment is processed and subscription is updated with confirmation.
- **Priority:** Critical

T-24 – Validate Admin Dashboard Access and Functionalities

- **Test Type/Category:** Functional / Administration
- **Objective:** Confirm that admins can access and use the dashboard tools.
- **Procedure:**
    1. Log in as an admin.
    2. Navigate to the Admin Dashboard.
    3. Verify availability of system metrics, user management, and configuration options.
- **Expected Outcome:** Admin Dashboard loads with all functionalities accessible.
- **Priority:** Major

T-25 – Validate Historical Data Trends Display

- **Test Type/Category:** Functional / Integration
- **Objective:** Ensure historical financial trends are rendered as graphs.
- **Procedure:**
    1. Navigate to “View Historical Data Trends” from the Dashboard.
    2. Select a company and a historical timeframe.
    3. Verify that a graph (e.g., quarterly revenue, EPS) is rendered correctly.
- **Expected Outcome:** A graph with clear labels and accurate data is displayed.
- **Priority:** Major

T-26 – Validate Company Health Dashboard Functionality

- **Test Type/Category:** Functional / Integration
- **Objective:** Verify overall health metrics of a company are presented.
- **Procedure:**
    1. Navigate to the “Health Dashboard” section.
    2. Select a company to view health metrics (e.g., revenue growth, expenses).
- **Expected Outcome:** Dashboard displays comprehensive metrics and visual indicators.
- **Priority:** Major

T-27 – Validate View Daily Stock Price Functionality

- **Test Type/Category:** Functional / Integration
- **Objective:** Ensure daily stock price data is accurately fetched and displayed.
- **Procedure:**
    1. On a company’s detail page, click “View Daily Stock Price”.
    2. Observe the displayed data (open, close, high, low).
- **Expected Outcome:** Data from Yahoo Finance API is shown accurately.

- **Priority:** Major

T-28 – Validate Compare Predictions vs. Outcomes Feature

- **Test Type/Category:** Functional / Integration
- **Objective:** Check that historical predictions and actual outcomes are compared accurately.
- **Procedure:**
    1. Navigate to “Compare Predictions vs. Outcomes”.
    2. Select a company and timeframe.
    3. Verify both prediction and actual outcome data are displayed side by side.
- **Expected Outcome:** Accurate comparison view with clearly labeled data.
- **Priority:** Critical

T-29 – Validate Notification Deletion Functionality

- **Test Type/Category:** Functional
- **Objective:** Ensure users can delete notifications from the UI.
- **Procedure:**
    1. Navigate to the Notifications section.
    2. Select one or more notifications and click “Delete”.
    3. Confirm deletion if prompted.
- **Expected Outcome:** Notifications are removed from the UI and do not reappear.
- **Priority:** Major

T-30 – Validate Feedback Submission Functionality

- **Test Type/Category:** Functional
- **Objective:** Confirm that users can submit feedback successfully.
- **Procedure:**
    1. Navigate to “Feedback & Help Center”.
    2. Fill in the feedback form (rating, comments).
    3. Click “Submit”.
- **Expected Outcome:** Feedback is submitted and a confirmation message is shown.
- **Priority:** Major

T-31 – Validate PDF Export of Financial Predictions

- **Test Type/Category:** Functional / Integration
- **Objective:** Verify that predictions can be exported as a PDF.
- **Procedure:**

1. On the Financial Predictions page, click “Export”.
2. Select PDF format.

- **Expected Outcome:** A PDF file is generated and downloaded with correct data.
- **Priority:** Minor

T-32 – Validate Interactive Chart Zoom and Pan Functionality

- **Test Type/Category:** Functional / Usability
- **Objective:** Ensure financial charts support zooming and panning.
- **Procedure:**
    1. On any financial chart (e.g., Predictions page), use touch or mouse controls to zoom and pan.
- **Expected Outcome:** Smooth zooming/panning with clear and accurate data display.
- **Priority:** Minor

T-33 – Validate Advanced Search by Ticker Symbol

- **Test Type/Category:** Functional
- **Objective:** Verify that searching by ticker returns the correct company data.
- **Procedure:**
    1. On the Home page, enter a ticker (e.g., “AAPL”) in the search bar.
    2. Initiate the search.
- **Expected Outcome:** Correct company details and financial data are displayed.
- **Priority:** Major

T-34 – Validate User-Specific Recommendations on Home Page

- **Test Type/Category:** Functional / Personalization
- **Objective:** Ensure personalized recommendations are displayed based on user history.
- **Procedure:**
    1. Log in as a registered user.
    2. Navigate to the Home page.
- **Expected Outcome:** Recommendations reflect the user’s interests and past interactions.
- **Priority:** Major

T-35 – Validate Data Synchronization Across Multiple Devices

- **Test Type/Category:** Functional / Integration
- **Objective:** Verify data updates on one device are reflected on another in real time.
- **Procedure:**

1. Log in simultaneously on two devices (e.g., mobile and desktop).
2. Update profile information on one device.

- **Expected Outcome:** Changes synchronize across devices with minimal delay.
- **Priority:** Major

T-36 – Validate Error Handling for Empty API Response

- **Test Type/Category:** Functional / Error Handling
- **Objective:** Ensure the system handles empty API responses gracefully.
- **Procedure:**
    1. Simulate an API returning an empty response (e.g., Yahoo Finance for a company with missing data).
    2. Attempt to load the data.
- **Expected Outcome:** A clear error or fallback message is displayed, and the UI remains stable.
- **Priority:** Major

T-37 – Validate Auto-Refresh of Predictions

- **Test Type/Category:** Functional / Performance
- **Objective:** Confirm that predictions auto-refresh without manual intervention.
- **Procedure:**
    1. On the Financial Predictions page, leave the page open for the auto-refresh interval.
    2. Observe if predictions update automatically.
- **Expected Outcome:** Predictions refresh automatically without errors.
- **Priority:** Minor

T-38 – Validate User Account Deletion Functionality

- **Test Type/Category:** Functional
- **Objective:** Ensure that when a user deletes their account, all associated data is removed.
- **Procedure:**
    1. Navigate to the Profile page.
    2. Click “Delete Account” and confirm.
- **Expected Outcome:** Account and all associated data (watchlist, predictions, etc.) are removed from the system.
- **Priority:** Critical

T-39 – Validate Company Details Page Content Consistency

- **Test Type/Category:** Functional
- **Objective:** Verify that the company details page displays consistent and accurate information.
- **Procedure:**
    1. Navigate to a company's details page.
    2. Compare displayed information with data from integrated APIs.
- **Expected Outcome:** The page shows accurate, complete, and consistent financial metrics and profile data.
- **Priority:** Major

T-40 – Validate Inline Text Sentiment Integration Display

- **Test Type/Category:** Functional
- **Objective:** Ensure that sentiment analysis data from reports is integrated and shown.
- **Procedure:**
    1. On a company's details page, verify that sentiment scores and highlights are visible.
- **Expected Outcome:** Sentiment data is correctly displayed alongside financial data.
- **Priority:** Major

T-41– Validate JSON Export of Financial Predictions

- **Test Type/Category:** Functional
- **Objective:** Confirm that prediction data can be exported in JSON format.
- **Procedure:**
    1. On the Financial Predictions page, click "Export".
    2. Select JSON as the format.
- **Expected Outcome:** A JSON file with the correct structured prediction data is downloaded.
- **Priority:** Minor

5.2 Non-Functional Test Cases

T-42 – Validate Prediction Accuracy Threshold

- **Test Type/Category:** Performance
- **Objective:** Verify that the ML model meets the required accuracy (MAPE below 15% for 80% of companies).
- **Procedure:**
    1. Run the ML model with a test dataset of known outcomes.
    2. Compare predictions against actual values.
- **Expected Outcome:** MAPE is below 15% for at least 80% of companies.
- **Priority:** Critical

T-43 – Validate Data Encryption for User Passwords

- **Test Type/Category:** Security
- **Objective:** Ensure that user passwords are stored securely in an encrypted (hashed) format.
- **Procedure:**
    1. Register a new user account.
    2. Inspect the stored password in the database using secure means.
- **Expected Outcome:** Password is stored in an encrypted format.
- **Priority:** Critical

T-44 – Validate Unauthorized Access Prevention

- **Test Type/Category:** Security
- **Objective:** Verify that restricted pages and API endpoints cannot be accessed without proper authentication.
- **Procedure:**
    1. Attempt to access secured areas without logging in.
- **Expected Outcome:** Access is denied with an appropriate error or redirection to the Login page.
- **Priority:** Critical

T-45 – Validate Multi-Factor Authentication

- **Test Type/Category:** Security
- **Objective:** Confirm that users are required to complete a second authentication step (e.g., OTP) during login.
- **Procedure:**
    1. Enable MFA in user settings.
    2. Log in with valid credentials and verify that an additional authentication step is prompted.
- **Expected Outcome:** User must successfully complete MFA to access the system.
- **Priority:** Major

T-46 – Validate Mobile Responsiveness of the Dashboard

- **Test Type/Category:** Usability / Compatibility
- **Objective:** Ensure the Dashboard displays properly on various mobile devices and screen sizes.
- **Procedure:**
    1. Access the Dashboard on multiple mobile devices (different OS and screen sizes).
    2. Verify that all elements render correctly and are functional.
- **Expected Outcome:** UI adjusts responsively; functionality is maintained across devices.

- **Priority:** Major

T-47 – Validate Cross-Browser Compatibility

- **Test Type/Category:** Compatibility
- **Objective:** Ensure the web application functions consistently across different browsers.
- **Procedure:**
    1. Open the application in browsers such as Chrome, Firefox, and Edge.
    2. Navigate through key pages (Login, Dashboard, Predictions).
- **Expected Outcome:** UI and functionality are consistent across all tested browsers.
- **Priority:** Major

T-48 – Performance Test – Simulate 200 Concurrent User Sessions

- **Test Type/Category:** Performance
- **Objective:** Verify that the system can handle heavy user load.
- **Procedure:**
    1. Use a performance testing tool to simulate 200 concurrent users performing typical operations (login, data retrieval, etc.) [12].
- **Expected Outcome:** Response times remain below 2 seconds per request with acceptable performance degradation.
- **Priority:** Critical

T-49 – Performance Test – Data Fetching Response Time

- **Test Type/Category:** Performance
- **Objective:** Ensure financial data fetching is performed within an acceptable time frame.
- **Procedure:**
    1. Measure the time taken for the system to fetch data for a selected company.
- **Expected Outcome:** Response time is less than 2 seconds.
- **Priority:** Critical

T-50 – Validate System Recovery from Network Interruption

- **Test Type/Category:** Reliability
- **Objective:** Verify that the system recovers gracefully after a network interruption.
- **Procedure:**
    1. Simulate a temporary network loss during a data-fetch operation.
    2. Restore network connectivity.
- **Expected Outcome:** The system resumes data fetching without data loss.

- **Priority:** Critical

T-51 – Validate API Endpoint Security with Bearer Token

- **Test Type/Category:** Security
- **Objective:** Ensure that secured API endpoints are accessible only with a valid bearer token.
- **Procedure:**
    1. Attempt to access a secured API endpoint without or with an invalid token.
- **Expected Outcome:** Access is denied with an appropriate error message.
- **Priority:** Critical

T-52 – Validate In-App Documentation Accessibility

- **Test Type/Category:** Documentation / Usability
- **Objective:** Confirm that in-app documentation or help content is easily accessible.
- **Procedure:**
    1. Click on the “Help” or “Documentation” link within the app.
- **Expected Outcome:** Detailed, user-friendly documentation is displayed.
- **Priority:** Minor

T-53 – Validate Access to New User Training Guide

- **Test Type/Category:** Usability
- **Objective:** Ensure that a new user training guide or onboarding module is accessible and clear.
- **Procedure:**
    1. Log in as a new user and access the training guide.
- **Expected Outcome:** Training material is available and assists the user effectively.
- **Priority:** Minor

T-54 – Validate Caching Mechanism for Financial Predictions

- **Test Type/Category:** Performance
- **Objective:** Verify that cached data is used to reduce load times.
- **Procedure:**
    1. Request the same financial data twice in succession.
- **Expected Outcome:** The second request is served faster via the cache with correct data.
- **Priority:** Major

T-55 – Validate ML Model Prediction Generation Time

- **Test Type/Category:** Performance
- **Objective:** Ensure that the ML model generates predictions within acceptable time limits.
- **Procedure:**
    1. Measure the time taken for the ML model to produce predictions for a given company.
- **Expected Outcome:** Prediction generation completes in less than 5 seconds.
- **Priority:** Minor

T-56 – Validate Display of Data Privacy Consent

- **Test Type/Category:** Compliance
- **Objective:** Confirm that users must acknowledge the data privacy policy during registration or first login.
- **Procedure:**
    1. During registration/first login, verify that the data privacy policy is presented and requires user consent.
- **Expected Outcome:** User must agree to the privacy consent before proceeding.
- **Priority:** Critical

T-57 – Validate System Uptime under Extended Operation

- **Test Type/Category:** Reliability
- **Objective:** Ensure that the system maintains high uptime over an extended period.
- **Procedure:**
    1. Monitor system operation continuously (via logs or simulation) over an extended period.
- **Expected Outcome:** Uptime of at least 99% with no critical failures.
- **Priority:** Critical

T-58 – Validate Real-Time Data Updates via WebSocket

- **Test Type/Category:** Performance / Real-Time
- **Objective:** Confirm that real-time events update the Dashboard immediately via WebSocket.
- **Procedure:**
    1. Log in and keep the Dashboard active.
    2. Simulate a real-time event (e.g., sudden financial metric change) using test data or simulation tools.
- **Expected Outcome:** The WebSocket pushes an update immediately and the Dashboard displays the new data.

- **Priority:** Critical

T-59 – Validate Session Timeout and Re-Login Prompt

- **Test Type/Category:** Security / Usability
- **Objective:** Ensure that inactive sessions time out and prompt for re-login.
- **Procedure:**
    1. Log in and leave the application idle until the session timeout elapses.
    2. Attempt an action after timeout.
- **Expected Outcome:** User is automatically logged out and prompted to log in again.
- **Priority:** Critical

T-60 – Validate System Behavior Under Extreme Load Conditions

- **Test Type/Category:** Performance
- **Objective:** Assess system stability when subjected to stress beyond normal load.
- **Procedure:**
    1. Simulate a stress test with 500+ concurrent user sessions performing various operations.
    2. Monitor response times, error rates, and resource usage.
- **Expected Outcome:** System remains stable with acceptable performance degradation and logs potential bottlenecks.
- **Priority:** Critical

T-61 – Validate Penetration Testing on API Endpoints

- **Test Type/Category:** Security
- **Objective:** Ensure API endpoints are secure against common attack vectors (e.g., SQL injection, XSS).
- **Procedure:**
    1. Manually perform penetration testing using common attack techniques on API endpoints [13].
    2. Document any vulnerabilities found.
- **Expected Outcome:** All attempted attacks are mitigated or blocked; no exploitable vulnerabilities.
- **Priority:** Critical

T-62 – Validate Audit Logging and Compliance Reporting

- **Test Type/Category:** Compliance

- **Objective:** Confirm that sensitive operations are logged for audit and compliance purposes.
- **Procedure:**
    1. Execute sensitive operations (e.g., account updates, payment transactions).
    2. Review audit logs for completeness and compliance with standards (GDPR, CCPA, etc.).
- **Expected Outcome:** Audit logs capture all necessary details and meet regulatory requirements.
- **Priority:** Major

T-63 – Validate Usability Testing with Real Users

- **Test Type/Category:** Usability
- **Objective:** Ensure that the application is user-friendly and intuitive.
- **Procedure:**
    1. Conduct a usability session with representative end users.
    2. Collect qualitative and quantitative feedback on UI clarity, navigation, and overall experience.
- **Expected Outcome:** Feedback indicates high usability with minimal areas of confusion.
- **Priority:** Major

T-64 – Validate Backup and Recovery Procedures

- **Test Type/Category:** Reliability
- **Objective:** Verify that data can be restored effectively in case of data loss.
- **Procedure:**
    1. Simulate a controlled data loss scenario (e.g., accidental deletion).
    2. Initiate the backup recovery process.
- **Expected Outcome:** Data is restored with minimal downtime and no significant data loss.
- **Priority:** Critical

6.Consideration of Various Factors in Engineering Design

In designing Plutos Equities, we have meticulously considered a multitude of factors to ensure the system's functionality, reliability, maintainability, and cost-effectiveness. Below is an in-depth analysis of each factor and its impact on our design process, rated on a scale from 0 (none) to 10 (maximum).

6.1.Constraints & Standards

6.1.1.Public Health

The platform does not directly impact public health. Although, it indirectly contributes to financial stability, reducing stress and anxiety among investors, its impact level is 0.

6.1.2.Safety

Although there are no physical safety risks, cybersecurity measures are implemented to protect user data. The design includes multi-factor authentication (MFA), data encryption, and role-based access control (RBAC) to prevent unauthorized access and financial fraud. We also consider model safety. Therefore, its impact level is 9.

6.1.3.Security

Handling sensitive financial data necessitates robust security measures. We have integrated strong authentication protocols, encrypted data storage, and secure API endpoints to mitigate security risks. , which provide enterprise-grade security and compliance with industry standards. Compliance with regulations such as the General Data Protection Regulation (GDPR) ensures comprehensive privacy protection. Plutos Equities employs enterprise-grade encryption, secure authentication (OAuth 2.0), and regular security audits to protect against cyber threats. Azure Cloud services ensure compliance with **ISO 27001** standards for secure financial data handling [9][10]. Therefore, its impact level is 9.

6.1.4.Welfare

One of the reasons why Pluto is different from its competitors is its transparency, it takes into account the forecasting process and informs the user. The system promotes financial literacy and transparency, helping investors and businesses make informed decisions. AI-driven analytics enable better risk management, reducing financial losses and ensuring stable investments. Unlike many existing financial forecasting tools that provide "black-box" predictions, Plutos Equities explains the reasoning behind its forecasts by allowing users to see contributing factors such as market trends, earnings reports, and sentiment analysis. This enhances trust and financial decision-making. Therefore, its level of influence is 9, which is a high degree.

6.1.5.Global Considerations

The platform integrates public financial data of the U.S. (NASDAQ) and the U.S. is a country that directs the world economy. For this reason, the whole world is affected by the economic movements that take place in the U.S. Although this target narrowing does not seem global, the leading countries of the world, developed and developing countries, make economic investments with American stocks at individual or corporate levels and can organize their own financial management in this direction, therefore, the evaluation was given higher than 5 on a global scale, but it was not given a full score because it did not cover it directly. Therefore, its impact level is 8.

6.1.6.Cultural Considerations

Recognizing the diversity of our user base, Plutos Equities includes multilingual support to cater to 3 different linguistic preferences; Turkish, English and French. However currently the project is focused on U.S. centered Nasdaq companies and by targeting users who follow this

area, it moves away from an environment that can be accepted all over the world. Therefore, its impact level is 3.

6.1.7. Social Considerations

We have designed the platform to be accessible and inclusive, ensuring that both institutional and individual investors can benefit. Ethical AI usage prevents market manipulation and misinformation, enhancing trust in financial predictions. Therefore, its impact level is 6.

6.1.8.Environmental Considerations

To minimize our carbon footprint, we utilize cloud services with a strong commitment to renewable energy. Our algorithms are optimized for energy-efficient computing, and we promote digital-first financial analysis to reduce waste. Therefore, its impact level is 5.

6.1.9.Economic Considerations

Economic impact is central to our system. By providing accurate financial predictions, we assist stakeholders in mitigating risks, identifying investment opportunities, and optimizing capital allocation. Considerations of cloud costs, API fees, and subscription models have been integral in designing an economically sustainable solution. The pricing table of the product also ensures affordability. Therefore, its impact level is 10.

6.2.Factors Table & Discussions

The Factors Table provides a quantitative evaluation of how various global, social, technical, and economic considerations affect the design of Plutos Equities. Each factor is assigned an impact level from 0 (no impact) to 10 (maximum impact) based on its relevance and influence on the system.

Factor	Impact Level (0-10)
Public Health	0
Safety	9
Security	9
Welfare	9
Global	8
Cultural	3
Social	6
Environmental	5

Economic	10

Public health isn't directly influenced by the system, but economic stability can lower investor anxiety. Safety is of critical importance due to the necessity for cybersecurity and model integrity variables. Security is also of great significance since it ensures the protection of user details and compliance with financial regulations. Welfare has extremely high influence as the system endorses financial knowledge and openness. Global factors are significant, as U.S. markets drive world investment strategies. Cultural factors are negligible, with some because of multilingual support but with no accommodation of non-NASDAQ investors. Social factors contribute moderately as the platform enhances inclusivity and makes financial information accessible to everyone. Environmental factors contribute moderately, as cloud computing optimizations and efforts at reducing computational wastage are employed. Lastly, economic factors play the greatest role, since the system is actually meant to facilitate better financial decision-making and risk management.

7.Teamwork Details

7.1.Contributing and functioning effectively on the team

Plutos Equities is a **collaborative and interdisciplinary project**, where each team member has taken **specific leadership roles** while ensuring **cross-functional collaboration**. The project was structured into **five core teams—Modeling, Data, Development, Testing & Research, and Project Management**—to maximize efficiency and innovation.

- **Uygar Aras – Project Manager & Financial Strategy Lead, Data & Modeling Support**
  Uygar **led the overall project management**, ensuring that all aspects of Plutos Equities—from **financial forecasting methodologies** to **software development timelines**—were aligned with the project's objectives. His role also involved **coordinating cross-team collaboration**, making sure that insights from **data, modeling, development, and testing** were effectively integrated into the final product.
  In addition to his leadership role, Uygar was actively involved in **data management and financial modeling**, continuously receiving assignments from **Emre** to assist in **data preprocessing, model validation, and feature engineering**. His contributions in **defining model evaluation strategies** and integrating **economic factors into the forecasting process** played a crucial role in improving the reliability and accuracy of the predictive models.

- **Alara Zeybek – Modeling Team Lead & Frontend UX Collaborator**
  Alara **led the modeling team**, ensuring that the **machine learning models used for financial forecasting were optimized and scalable**. She was responsible for designing and refining **time-series forecasting models (LSTM, Prophet, and ensemble methods)** while working closely with the **data team to ensure proper input data quality**.
  In addition to leading the modeling efforts, Alara also collaborated with **Adil** on **frontend UX design**, making sure the system was not only functionally robust but also visually intuitive for financial professionals.

- **Emre Akgül – Data Team Lead & Machine Learning Engineer**
  Emre **led the data team**, overseeing the **collection, preprocessing, and management of**

structured and unstructured financial data. He developed **automated data pipelines**, ensuring **real-time retrieval and storage of financial data from APIs** such as **SEC EDGAR, Yahoo Finance, and Bloomberg**.
As part of his leadership, Emre continuously assigned tasks to **Uygar**, ensuring that **data transformations, feature selection, and preprocessing** were optimized for the **modeling team**. His coordination with Alara was essential in ensuring that **high-quality structured datasets** were fed into the forecasting models.

- **Asım Adil Can – Development Team Lead & Full-Stack Engineer**
  Asım **led the development team**, ensuring that the **backend and frontend infrastructure of Plutos Equities was scalable and secure**. His key responsibilities included:

  - **Backend API development**
  - **Database management**
  - **User authentication & security**

  He also collaborated with **Alara** to improve the **frontend UI/UX**, ensuring that financial data was presented in an intuitive and user-friendly manner. Additionally, he worked closely with **Gün** to **enhance API security** and optimize performance.

- **Gün Taştan – Research & Testing Lead, Model & Data Support**
  Gün led the testing and research team, ensuring that all **backend services, APIs, and ML models** met **performance, security,** and **accuracy** standards. He developed **automated test cases**, performed load testing, and validated **system security measures**.

  Additionally, Gün played a crucial role in **sentiment analysis**, helping Emre process and analyze **news sentiment data, earnings call transcripts, and investor sentiment scores**. His contributions improved the **NLP-based sentiment models**, refining how financial news impacts stock forecasts.

  Gün also took on **critical modeling and data-related tasks**, going beyond data integrity checks to **assist in feature engineering**, **optimize preprocessing pipelines**, and **validate financial model outputs**. His work ensured that **forecasting models produced reliable and high-confidence predictions**, minimizing biases and improving model stability.

This structured **team hierarchy**, with strong **cross-functional collaboration**, allowed Plutos Equities to **develop a robust and high-performance financial forecasting system** [15].

7.2.Helping creating a collaborative and inclusive environment

To ensure smooth project execution, the **Plutos Equities team** adopted a **collaborative and transparent approach**, prioritizing **knowledge sharing, inclusivity, and efficient task management**.

- **Agile Development & Task Management**
  - The team followed an **Agile workflow**, using **Jira for sprint planning** and **Slack for daily updates** [14].

    - Weekly **stand-up meetings** helped **identify roadblocks** and allowed for quick adjustments to workload distribution.
- **Open Communication & Knowledge Sharing**
    - **GitHub Discussions and Documentation** were used to track project progress and clarify technical details.
    - Regular **cross-team meetings** ensured alignment between **modeling, data, development, and testing** teams.
    - Uygar **facilitated communication between technical and business teams**, ensuring that financial forecasting goals aligned with the technical capabilities of the platform.
- **Cross-Team Collaboration**
    - **Alara (Modeling) worked closely with Emre (Data)** to refine ML models with high-quality structured datasets.
    - **Adil (Development) collaborated with Alara (UX Design)** to ensure that the **frontend was user-friendly and engaging**.
    - **Gün (Testing) validated NLP-based sentiment analysis models** alongside Emre, refining the impact of **news-based stock predictions**.

These initiatives ensured **Plutos Equities evolved into a high-quality, user-friendly, and scalable forecasting system**.

7.3.Taking lead role and sharing leadership on the team

Plutos Equities operated under a distributed leadership model, where different members led specific areas while collaborating with other teams to ensure smooth integration.

- **Project Scope & Business Strategy (Led by Uygar Aras)**
    - Defined **financial forecasting methodologies** and ensured business alignment.
    - Managed **timelines, deliverables, and stakeholder communication**.
    - Supported **data processing and modeling efforts** under Emre's leadership.
- **Machine Learning & Modeling (Led by Alara Zeybek)**
    - Designed and optimized **time-series forecasting models (LSTM, Prophet, Ensemble methods)**.
    - Led research into **ML algorithm performance and explainability**.
    - Collaborated with **Gün and Emre** on sentiment model validation.
- **Data Engineering & Management (Led by Emre Akgül)**
    - Developed **real-time financial data retrieval pipelines**.
    - Ensured that **structured datasets were properly preprocessed for ML models**.
    - Assigned **data and preprocessing tasks** to Uygar, Gün, and Alara.
- **Full-Stack Development & Cloud Infrastructure (Led by Asım Adil Can)**
    - Designed **backend API architecture** and **cloud-based storage solutions**.
    - Managed **Azure cloud services for ML model hosting and financial data processing**.
    - Implemented **secure authentication mechanisms**.
- **Research & Testing (Led by Gün Taştan)**
    - Developed **test cases for system security, stability, and performance**.
    - Conducted **validation of sentiment analysis models for stock forecasting**.
    - Assisted both **data and modeling teams in debugging ML predictions**.

- **Cross-Functional Leadership & Collaboration**
  - **Uygar** supported **Emre** in **data preprocessing and model tuning**.
  - **Alara** and **Adil** worked together on **UI/UX** enhancements.
  - **Gün** collaborated with **Emre** to refine **sentiment analysis pipelines**.

By distributing leadership responsibilities, the team ensured e**fficiency, agility, and innovation**, resulting in the successful development of Plutos Equities.

8.Glossary

- **EPS (Earnings Per Share):** A financial metric indicating a company's profitability, calculated as net income divided by outstanding shares.
- **Revenue:** The total income generated by a company before expenses are deducted.
- **Net Income:** The profit of a company after all expenses, taxes, and costs have been deducted from total revenue.
- **Gross Margin:** The percentage of revenue remaining after deducting the cost of goods sold (COGS), indicating profitability.
- **Operating Expenses:** The costs required for running the core business operations, excluding costs related to manufacturing.
- **SEC (Securities and Exchange Commission):** The U.S. regulatory body overseeing securities markets and protecting investors.
- **Market Sentiment:** The overall attitude of investors toward a particular security or the financial market, often determined through sentiment analysis of financial news.
- **ML (Machine Learning):** A field of artificial intelligence that enables systems to learn from data and make predictions.
- **LSTM (Long Short-Term Memory):** A type of recurrent neural network (RNN) designed for time-series forecasting and sequential data.
- **Prophet Model:** A forecasting model developed by Facebook, optimized for handling seasonality and missing data in time-series forecasting.
- **Feature Engineering:** The process of selecting and transforming variables to improve machine learning model performance.
- **Hyperparameter Tuning:** The optimization of model parameters to enhance performance and accuracy.
- **Backtesting:** The process of testing a predictive model using historical data to evaluate its performance.
- **Ensemble Learning:** A technique that combines multiple models to improve prediction accuracy and robustness.
- **API (Application Programming Interface):** A set of functions that allow applications to communicate and exchange data.
- **REST API:** A web service architecture that allows communication between systems using HTTP requests.
- **OAuth 2.0:** A secure authorization framework that enables third-party applications to access user data without exposing credentials.
- **Cloud Computing:** The practice of using remote servers to store, manage, and process data, rather than local infrastructure.
- **Azure Cloud Services:** Microsoft’s cloud computing platform providing resources for scalable computing, storage, and machine learning workloads.

- **SQL (Structured Query Language):** A programming language used for managing and querying relational databases.
- **Data Encryption:** The process of converting information into a secure format that prevents unauthorized access.
- **Functional Testing:** A type of software testing that ensures each function of an application operates as expected.
- **Integration Testing:** A testing phase where different software modules are combined and tested as a group.
- **Load Testing:** A performance test to evaluate how a system behaves under expected and peak loads.
- **Penetration Testing:** A security assessment method used to identify vulnerabilities by simulating cyberattacks.
- **GDPR (General Data Protection Regulation):** European Union regulation ensuring data privacy and protection for individuals.
- **Data Compliance:** Adhering to regulations and laws that govern the use, storage, and sharing of data.

9.References

[1] **Taylor, S. J., & Letham, B.** (2018). Forecasting at Scale. *The American Statistician, 72*(1), 37-45.

[2] **Fama, E. F.** (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance, 25*(2), 383-417.

[3] **Securities and Exchange Commission (SEC).** (n.d.). *EDGAR Database*. Retrieved from https://www.sec.gov/edgar.shtml

[4] **Kaggle.** (n.d.). *Financial Forecasting Datasets and Competitions*. Retrieved from https://www.kaggle.com/datasets

[5] **Microsoft Azure.** (n.d.). *Azure Machine Learning Documentation*. Retrieved from https://docs.microsoft.com/azure/machine-learning/

[6] **Brownlee, J.** (2017). *Machine Learning Mastery with Python: Understand Your Data, Create Accurate Models, and Work Projects End-to-End*. Machine Learning Mastery.

[7] **Hochreiter, S., & Schmidhuber, J.** (1997). Long Short-Term Memory. *Neural Computation, 9*(8), 1735–1780.

[8] **OpenAI.** (2020). *GPT-3: Language Models are Few-Shot Learners*. Retrieved from https://arxiv.org/abs/2005.14165

[9] **European Union.** (2016). *General Data Protection Regulation (GDPR)*. Official Journal of the European Union.

[10] **ISO/IEC 27001:2013.** (2013). *Information Security Management Systems – Requirements*. International Organization for Standardization.

[11] **Deloitte.** (2023). *Trends in AI-Driven Financial Forecasting*. Retrieved from https://www2.deloitte.com

[12] **Brownlee, J.** (2017). *Machine Learning Mastery with Python*. Machine Learning Mastery. *(Reused for clarity—reuse same author from [3] if needed.)*

[13] **Software Testing Help.** (n.d.). *Manual Software Testing: A Beginner's Guide*. Retrieved from https://www.softwaretestinghelp.com/manual-testing-tutorial-1/

[14] **Bruegge, B., & Dutoit, A. H.** (2004). *Object-Oriented Software Engineering: Using UML, Patterns, and Java* (2nd ed.). Prentice Hall.

[15] **European Union.** (2016). *General Data Protection Regulation (GDPR)*. Official Journal of the European Union. *(Reuse GDPR reference [10])*

[16] **Atlassian.** (n.d.). *Agile Project Management with Jira Software*. Retrieved from https://www.atlassian.com/software/jira

[17] **Sharpe, W. F.** (1964). Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk. *The Journal of Finance, 19*(3), 425–442.

[18] **Shiller, R. J.** (2014). *Irrational Exuberance* (3rd ed.). Princeton University Press.

[19] **Ray, S.** (2019). *Data Science and Machine Learning: Mathematical and Statistical Methods*. Wiley.

[20] **Zhang, G. P.** (2003). Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model. *Neurocomputing, 50*, 159-175.

[21] **Sharpe, W. F.** (1964). Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk. *The Journal of Finance, 19*(3), 425–442.

[22] **McKinney, W.** (2012). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. O'Reilly Media.

[23] **Kroll, J. A., & Feldman, M.** (2016). The Economics of Machine Learning in Financial Markets. *Journal of Financial Economics, 121*(2), 123-145.

[24] **IBM.** (n.d.). *IBM Watson for Financial Services Documentation*. Retrieved from https://www.ibm.com/cloud/financial-services

[25] **Amazon Web Services (AWS).** (n.d.). *AWS Cloud Architecture Documentation*. Retrieved from https://docs.aws.amazon.com/wellarchitected/latest/framework/

[26] **Google Cloud.** (n.d.). *Google Cloud Platform: BigQuery ML Documentation*. Retrieved from https://cloud.google.com/bigquery-ml/docs/