**A Web-based Financial Markets Data Analysis Application for Educational Use**

By

Kariuki Celestine

116533

A Project Report Submitted to the Strathmore Institute of Management and Technology in partial fulfilment of the requirements for the award of Diploma in Business Information Technology.

Diploma in Business Information Technology

**Strathmore University**

July 2024

# Declaration and Approval

I, **Kariuki Celestine**, declare that this project has not been submitted to any other University for the award of a Diploma in Business Information Technology.

Student Name: **Kariuki Celestine**

Sign: \_\_\_CelestineKariuki\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_15/08/2024\_\_\_\_\_\_\_\_

Supervisor’s Name: **Frederick Barasa**

Sign: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Abstract

Facing significant challenges due to the lack of affordable, user-friendly financial markets data analysis tools, the educational sector struggles with teaching and research in finance, business, economics, data science, and statistics. Students, professors, and researchers encounter obstacles with inadequate access to real-time financial data, economic indicators, and foreign exchange rates. Existing solutions are often prohibitively expensive or insufficiently detailed, leaving a notable gap in effective learning and research tools.

To address this, I have developed a user-friendly web application to provide real-time access to financial market data through intuitive text commands, buttons, and APIs from established financial data providers. This application, built with Python, includes features such as data downloading, searching, basic analysis, and report generation. Utilizing the Object-Oriented Analysis and Design (OOAD) approach has ensured modularity and scalability, while the Agile System Development Methodology has supported iterative development and responsiveness to user needs.

The implementation phase of the document includes detailed system testing, with the backend designed for efficient data management and integration with multiple financial data APIs. The frontend, developed using Python’s Streamlit library, enhances user interaction. The application has demonstrated its value as a practical tool for financial data analysis in educational contexts. Future enhancements will focus on transitioning to a graphical user interface, expanding data sources, and integrating AI for advanced analytics, aiming to further improve its functionality and accessibility.

# Table of Contents

[Declaration and Approval ii](#_Toc174577060)

[Abstract iii](#_Toc174577061)

[Table of Contents iv](#_Toc174577062)

[List of Tables viii](#_Toc174577063)

[List of Figures ix](#_Toc174577064)

[Chapter 1: Introduction 1](#_Toc174577065)

[1.1 Background 1](#_Toc174577066)

[1.2 Problem Statement 2](#_Toc174577067)

[1.3 General Aim 3](#_Toc174577068)

[1.4 Specific Objectives 3](#_Toc174577069)

[1.5 Research Questions 4](#_Toc174577070)

[1.6 Justification 4](#_Toc174577071)

[1.7 Scope 5](#_Toc174577072)

[1.8 Limitations 5](#_Toc174577073)

[1.8 Delimitation 6](#_Toc174577074)

[Chapter 2: Literature Review 7](#_Toc174577075)

[2.1 Introduction 7](#_Toc174577076)

[2.2 Challenges students and educators face in acquiring financial literacy 7](#_Toc174577077)

[2.2.1 Barriers to effective financial markets data analysis 7](#_Toc174577078)

[2.2.2 Importance of user-friendly interfaces 7](#_Toc174577079)

[2.2.3 User experience design for financial literacy education 8](#_Toc174577080)

[2.3 Data Acquisition and Integration 8](#_Toc174577081)

[2.3.1 Challenges in Accessing Financial Data for Educational Purposes 8](#_Toc174577082)

[2.3.2 Ensuring Data Security and Accuracy 8](#_Toc174577083)

[2.3.3 Existing Data Sources for Financial Education 8](#_Toc174577084)

[2.4 Benchmarking Existing Solutions 9](#_Toc174577085)

[2.4.1 Bloomberg Terminal 9](#_Toc174577086)

[2.4.2 Thomas Reuters’ Refinitiv Eikon 11](#_Toc174577087)

[2.4.3 FactSet 13](#_Toc174577088)

[2.4.4 Why my solution is more superior to others for educational use 17](#_Toc174577089)

[2.5 Conceptual Framework 17](#_Toc174577090)

[2.5.1 Introduction 17](#_Toc174577091)

[2.5.2 Literature-Based Framework 17](#_Toc174577092)

[2.5.3 Stakeholder Roles: 18](#_Toc174577093)

[2.5.4 Boundaries and Emerging Technologies 19](#_Toc174577094)

[Chapter 3: Research Methodology 21](#_Toc174577095)

[3.1 Introduction 21](#_Toc174577096)

[3.2 Research Design 21](#_Toc174577097)

[3.3 Agile Development System Development Methodology 22](#_Toc174577098)

[3.3.1 Ideation and Requirements Analysis Phase 22](#_Toc174577099)

[3.3.2 Design Phase 23](#_Toc174577100)

[3.3.3 Development Phase 23](#_Toc174577101)

[3.3.4 Testing Phase 24](#_Toc174577102)

[3.4 System Implementation 24](#_Toc174577103)

[3.5 System Architecture and Modules 25](#_Toc174577104)

[3.5.1 Data Acquisition Module 25](#_Toc174577105)

[3.5.2 Data Cleaning and Pre-processing Module 25](#_Toc174577106)

[3.5.3 Data Processing and Analysis Module 25](#_Toc174577107)

[3.5.4 Report Generation Module 25](#_Toc174577108)

[3.6 System Deliverables and Milestones 26](#_Toc174577109)

[3.6.1 Design Diagrams 26](#_Toc174577110)

[3.6.2 Database Connectivity 26](#_Toc174577111)

[3.6.3 User Interface Completion 27](#_Toc174577112)

[3.6.4 System Modules Development 27](#_Toc174577113)

[3.6.5 System Testing 27](#_Toc174577114)

[3.6.6 Final Documentation 27](#_Toc174577115)

[3.6.7 System Completion and Demo 27](#_Toc174577116)

[Chapter 4: System Analysis and Design 29](#_Toc174577117)

[4.1 Introduction 29](#_Toc174577118)

[4.2 System Analysis 29](#_Toc174577119)

[4.2.1 Functional Requirements 29](#_Toc174577120)

[4.2.2 Non-Functional Requirements 30](#_Toc174577121)

[4.3 System Design 30](#_Toc174577122)

[4.4 System Analysis 31](#_Toc174577123)

[4.5 System Analysis Diagrams 31](#_Toc174577124)

[4.5.1 Use Case UML Diagram 31](#_Toc174577125)

[4.5.2 Class UML Diagram 33](#_Toc174577126)

[4.5.2 Entity Relationship Diagram 34](#_Toc174577127)

[4.5.4 Sequence Diagram 36](#_Toc174577128)

[4.5.5 Wireframes 38](#_Toc174577129)

[4.5.6 System Architecture 42](#_Toc174577130)

[4.5.7 Component UML Diagram(Specific To The Web Application) 43](#_Toc174577131)

[Chapter 5: System Implementation Testing 45](#_Toc174577132)

[5.1 Introduction 45](#_Toc174577133)

[5.2 System Implementation 45](#_Toc174577134)

[5.2.1 Systems Backend 45](#_Toc174577135)

[5.2.2 System’s Frontend 47](#_Toc174577136)

[5.3 System Manual 49](#_Toc174577137)

[5.3.1 Generating Tables 49](#_Toc174577138)

[5.3.2 Generating Excel Reports 50](#_Toc174577139)

[5.4 System Testing 53](#_Toc174577140)

[5.4.1 Black Box Testing 53](#_Toc174577141)

[5.4.2 Functionality Testing 53](#_Toc174577142)

[5.4.3 Unit Testing 54](#_Toc174577143)

[5.5 Test Cases 55](#_Toc174577144)

[5.6 Test Results 56](#_Toc174577145)

[Chapter 6: Conclusions and Recommendations for Future Work 57](#_Toc174577146)

[6.1 Conclusions 57](#_Toc174577147)

[6.2 Recommendations 57](#_Toc174577148)

[6.3 Future Work 58](#_Toc174577149)

[References 59](#_Toc174577150)

[Appendix A: Gantt Chart 1](#_Toc174577151)

[Appendix B: Marking Guide 3](#_Toc174577152)

# List of Tables

[Table 1: Comparison between FactSet, Bloomberg and Thomas Reuter's Eikon 14](#_Toc174577419)

[Table 2:Test Cases 55](#_Toc174577420)

[Table 3:Test Results 56](#_Toc174577421)

# List of Figures

[Figure 1: View of The Bloomberg Desktop Terminal 9](#_Toc174575004)

[Figure 2: View of The Thomas Reuters Eikon Desktop Terminal Web Overview 11](#_Toc174575005)

[Figure 3:View of FactSet Terminal 13](#_Toc174575006)

[Figure 4: Conceptual Framework of the 'Infomark Web App' 18](#_Toc174575007)

[Figure 5: Agile Development Methodology using the Scrum Process 21](#_Toc174575008)

[Figure 6: Use Case UML Diagram for 31](#_Toc174575009)

[Figure 7:Class UML Diagram for 33](#_Toc174575010)

[Figure 8: ER Diagram for Infomark 35](#_Toc174575011)

[Figure 9: Sequence UML Diagram 37](#_Toc174575012)

[Figure 10: Infomark Terminal Web App Home Page 38](#_Toc174575013)

[Figure 11:Web App Home Page 39](#_Toc174575014)

[Figure 12:Terminal Table Display 40](#_Toc174575015)

[Figure 13: Terminal Graph Display 41](#_Toc174575016)

[Figure 14: Simple System Architecture Diagram 42](#_Toc174575017)

[Figure 15: Component UML Diagram(A type of System Architecture Diagram) 43](#_Toc174575018)

[Figure 16: Infomark Login/Register Page 46](#_Toc174575019)

[Figure 17: Apple's Historical Open, Close, High and Low Prices generated 47](#_Toc174575020)

[Figure 18: Searching for Company Ticker Symbols from the SEC 47](#_Toc174575021)

[Figure 19: Setting Date Range for Asset Data 48](#_Toc174575022)

[Figure 20: Table Generated For Apple's Historical Stock Price 49](#_Toc174575023)

[Figure 21:Infomark Web App Exports Table As CSV 50](#_Toc174575024)

[Figure 22: Apple Historical Prices Exported To CSV 51](#_Toc174575025)

[Figure 23:Running Unit Tests on Infomark Using Pytest On Visual Studio Code 53](#_Toc174575026)

[Figure 24: Gantt Chart for project 1](#_Toc174575027)

# Chapter 1: Introduction

## Background

Influencing a broad spectrum of disciplines including finance, business, economics, data science, and statistics, the global financial markets are integral to the world economy (Investopedia, n.d.). Evolving from a niche skill into a fundamental competency required across these fields, the ability to understand and analyse financial data has become crucial. Highlighting this shift, a 2021 study from the CFA Institute finds that 85% of employers worldwide consider financial literacy a crucial criterion when hiring recent graduates (CFA Institute, 2021). Despite the growing recognition of financial skills' importance, a substantial gap persists in accessing high-quality, comprehensive, and user-friendly financial data resources, particularly within educational settings.

Impeded by the lack of accessible financial data, effective teaching, learning, and research face significant challenges. A 2023 study by the OECD reports that 68% of educators globally experience difficulties integrating real-time financial data into their curricula, primarily due to limited access or high costs associated with traditional data providers (OECD, 2023). Exacerbating this issue for students in low- and middle-income countries, a 2021 World Bank report highlights that 72% of educational institutions lack access to affordable financial data solutions (World Bank, 2021). Creating an uneven playing field, these disparities obstruct educational opportunities and potentially restrict future career prospects for students in these regions.

Addressing this critical issue, my project aims to bridge this gap by developing a web-based financial data application specifically designed for educational purposes. Providing real-time access to extensive financial market data through an intuitive, user-friendly interface, the web application empowers students and educators with the tools needed for in-depth financial analysis, coursework completion, and research. The web application seeks to democratise access to financial data, removing barriers of cost and technical complexity that have traditionally hindered users from engaging with such data effectively.

Extending beyond merely facilitating data collection, the potential impact of the web application is significant. Noting a shift in employer expectations, the Association to Advance Collegiate Schools of Business reports that companies increasingly seek graduates who can critically analyse financial data and generate actionable insights (Association to Advance Collegiate Schools of Business, 2023). Supporting educators in incorporating real-time data into their teaching, the web application I have developed nurtures essential critical thinking skills, preparing students for careers in finance and business while equipping them to navigate an increasingly data-rich environment.

Representing a significant investment in the future of financial education, the web application fosters a generation of financially literate individuals capable of contributing meaningfully to a more inclusive and prosperous world. Envisioned as a transformative solution to persistent challenges in financial data accessibility within educational contexts, the web application seeks to enhance educational outcomes, support research, and prepare students for successful careers in a complex global financial landscape.

## 1.2 Problem Statement

In Kenya, a notable disconnect existing between theoretical financial education and its practical application impedes students' preparedness for real-world financial challenges. Exacerbating this issue, the lack of accessible, user-friendly tools hindering effective analysis of real-time financial data persists. Research by Chege, and Mwaura (Chege & Mwaura, 2020) underscores the scarcity of practical financial tools in Kenya, limiting students' ability to apply theoretical knowledge and impeding their financial literacy and practical skills. Students encountering significant difficulties in accessing up-to-date financial data for coursework and practical exercises struggle with integrating theoretical concepts and undermining their career readiness (Mwangi, 2018). The latter did equally happen to myself and my classmates during the course of our Business Statistics, and Business Finance & Economics courses offered by Strathmore’s DBIT course. Educators similarly struggling with integrating real-time financial data into their teaching face prohibitive costs and complex interfaces of existing platforms, compounded by resource constraints in Kenyan educational institutions (Chege & Mwaura, 2020).

The far-reaching consequences of these challenges impact both immediate educational outcomes and long-term career prospects. Graduates emerging with insufficient financial literacy to navigate complex financial situations face adverse effects on their future career opportunities and financial stability (Lusardi & Mitchell, 2014). Moreover, high costs and complexities of current data platforms create barriers for students, particularly in resource-constrained settings, perpetuating educational inequalities (Atkinson & Messy, 2012). My web application seeks to mitigate these issues by providing a user-friendly and affordable platform tailored for educational use in Kenya and beyond, enhancing financial literacy, supporting a competent workforce, and improving overall educational outcomes.

1.3 General Aim

To develop an affordable and user-friendly web application that provides comprehensive real-time financial markets data analysis tooling to enhance educational outcomes for students and educators in Kenya.

## 1.4 Specific Objectives

1. To investigate student and educator needs and design an intuitive student-friendly and educator-friendly experience, ensuring alignment with project goals and research findings.
2. To identify, acquire, and integrate external financial data APIs, establishing robust data feeds for the web application.
3. To perform a comparative analysis of existing financial markets data analysis tools, evaluating their effectiveness and relevance to identify gaps that the web application will address.
4. To develop and implement the web application software system, incorporating all designed functionalities and ensuring it meets specified requirements.
5. To test and refine the web application based on feedback and against its set requirements, ensuring it meets user needs and expectations.

## 1.5 Research Questions

1. What are the primary needs of students and educators in Kenya regarding financial markets data analysis tools, and how can the web application be designed to meet these needs?
2. Which external financial markets data sources are most suitable for integration into the web application, and what methods should be employed to ensure effective incorporation?
3. How do current financial markets data analysis tools compare in functionality, cost, and user experience, and what gaps can the web application fill?
4. What are the essential functionalities required for the web application, and how can these be developed and implemented effectively?
5. What criteria should be used to evaluate user acceptance of the web application, and how can user feedback be utilised to enhance the system?

## 1.6 Justification

Research highlights a significant gap between theoretical financial knowledge and practical application, with many graduates struggling to analyse real-world data effectively (Huston, 2004). This gap impedes their ability to manage financial challenges, affecting their well-being and limiting their future contributions (OECD, 2023). My web application addresses this issue by providing a user-friendly, affordable tool that facilitates hands-on learning and helps students apply theoretical concepts confidently (Fullerton & Taylor, 2017).

Educators, facing difficulties with the complexity and cost of existing financial data tools, find the web application to be a streamlined, cost-effective solution. By fostering an active learning environment, it supports dynamic exploration of financial concepts and develops students' interest in financial literacy. Unlike most financial markets data analysis tools out there, my application is web-based and thus more user-friendly for non-technical finance and statistics students and educators, ensuring that they only focus on developing their finance and statistical analysis skills. I developed using Python because Python is the most popular language in the financial technology industry (Stack Overflow, 2023).

## 1.7 Scope

This project focuses on developing a functional prototype of the web application within three months. It includes conducting user research, integrating essential financial data sources, and implementing basic functionalities for data acquisition and analysis. The scope involves creating a preliminary user interface and carrying out initial user testing to gather feedback. The project excludes the integration of additional data sources, advanced analytical features, a fully refined user interface, and extensive security measures, which will be addressed in future development phases to maintain a focused and manageable initial release.

## 1.8 Limitations

The development of the web application faces several limitations. Initially, the project has been focused on integrating financial data from established markets such as the US, EU, and major emerging economies like China, India, and Brazil. The challenge of acquiring comprehensive and affordable data for frontier economies, including Kenya, will be addressed in future iterations, because it requires special and carefully mediated partnership agreements with providers in these countries who strictly put this data under strict proprietary licenses.

The initial version emphasises core functionalities, such as data retrieval and basic visualisation, while advanced features like complex statistical analysis and portfolio management will be reserved for subsequent updates. The user interface supports essential navigation and functionality, with further refinements planned based on user feedback and accessibility standards. Basic security measures are implemented initially, with advanced security considerations deferred to future development as the platform evolves. These limitations reflect the project’s current focus and provide a foundation for iterative improvements.

1.8 Delimitation

To address the limitations outlined in section 1.7, the web application implements several strategic measures. Initially, the project focuses on integrating financial data from established markets and emerging economies to ensure a solid foundation and effective user testing within the given timeframe. Future plans involve exploring partnerships with data providers, such as the Nairobi Securities Exchange (NSE), to acquire Kenya-specific data as a subsequent enhancement. Additionally, a tiered subscription model is developed to support the costs associated with expanding data coverage, including a "Paid Educational Tier" for institutions. This approach allows the platform to gradually incorporate more comprehensive data sets and functionalities, addressing limitations while providing a strong basis for future growth and adaptation.

# Chapter 2: Literature Review

## 2.1 Introduction

This chapter examines existing research related to the limitations of financial literacy education and explores how technology can bridge the gap. It directly connects to the problem statement (Section 1.2) by highlighting the identified challenges and how the web-based financial markets data analysis application for educational use can serve as a solution. Lastly, it presents a conceptual framework of the system.

2.2 Challenges students and educators face in acquiring financial literacy

2.2.1 Barriers to effective financial markets data analysis

Effective financial markets data analysis results in critical improvements to financial literacy. Students and educators face significant obstacles, including the complexity of financial data and a lack of adequate analytical skills. Research by Lusardi and Mitchell (Lusardi & Mitchell, 2014) shows that the overwhelming volume of data and the intricacy of financial instruments hinder users' ability to perform accurate analysis. However, simplified data presentation and enhanced analytical tools lead to overcoming these challenges and improving educational outcomes (Hastings, Madrian, & Skimmyhorn, 2013) .

2.2.2 Importance of user-friendly interfaces

Usability is paramount for successful technology adoption, especially in educational settings. Chen and C. Tsai (Chen & C, 2017) emphasize that user-friendly interfaces are essential for ensuring that educational tools are accessible and effective for learners with varying technical skill levels. This also aligns with the findings of Morris and Venkatesh (Morris & Venkatesh, 2000) .

2.2.3 User experience design for financial literacy education

Interactive and engaging learning experiences are crucial for financial literacy education. Hastings and Weinstein (Hastings & Weinstein, 2013) demonstrate that effective user experience design can significantly enhance learner motivation and knowledge retention. Gibson and O’Hara, in their work (Gibson & O'Hara, 2016), further support this by detailing design principles that promote a positive user experience in educational technologies, highlighting their importance in fostering effective financial literacy.

2.3 Data Acquisition and Integration

2.3.1 Challenges in Accessing Financial Data for Educational Purposes

Accessing comprehensive and reliable financial data is a persistent challenge for educational institutions, often due to high costs and technical barriers (Genc, 2019) . Addressing these challenges requires exploring alternative data sources and forming strategic partnerships to ensure that the web application can provide accurate and relevant financial data for educational purposes.

2**.3.2 Ensuring Data Security and Accuracy**

Ensuring the security and accuracy of financial data is critical for maintaining the integrity of educational tools. Kshetri, in his work (Kshetri, 2017), discusses the importance of robust data security measures in financial technologies to protect sensitive information. Janssen and Kuk, in their work (Janssen & Kuk, 2016), emphasize that maintaining high data quality and accuracy is essential for effective financial analysis and positive educational outcomes.

### 2.3.3 Existing Data Sources for Financial Education

Several financial data providers offer Application Programming Interfaces (APIs) for programmatic access to financial data. Existing solutions from organizations like Alpha Vantage, Financial Modelling Prep, NASDAQ, Federal Reserve of New York, Intrinio, Yahoo Finance and IEX Cloud do inform data acquisition strategies for my web application.

## 2.4 Benchmarking Existing Solutions

2.4.1 Bloomberg Terminal



Figure : View of The Bloomberg Desktop Terminal

The Bloomberg Terminal, developed by Bloomberg Finance L.P., is widely acknowledged as the premier tool for professional financial markets data analysis (Bloomberg Finance L.P. , 2024). It provides an extensive array of features crucial for in-depth financial research, including real-time and historical market data on various financial instruments such as equities, fixed income, currencies, commodities, and derivatives (Bloomberg Finance L.P. , 2024). The Bloomberg Terminal is equipped with sophisticated charting and analytics tools, facilitating both advanced technical and fundamental market analysis (Huang & Kurov, 2017). Additionally, it offers access to comprehensive news feeds, company filings, and other critical financial information sources, along with a secure messaging platform designed to enhance communication within the financial sector (Boehmer & Wu, 2021).

Despite its industry-leading capabilities, the Bloomberg Terminal presents several limitations when considered for educational purposes. The prohibitive cost of subscription fees significantly limits access for many educational institutions and students, thereby restricting its broader educational application (Jiang, 2019). Furthermore, the terminal’s complexity, tailored predominantly for experienced financial professionals, results in a steep learning curve for novices (Kwak & Lee, 2018) , most of whom are active students, educators, and recent graduates. Additionally, the overwhelming volume of data and functionalities available can be daunting for users who are new to financial analysis, potentially hindering their learning experience (Cui, 2020).

### 2.4.2 Thomas Reuters’ Refinitiv Eikon

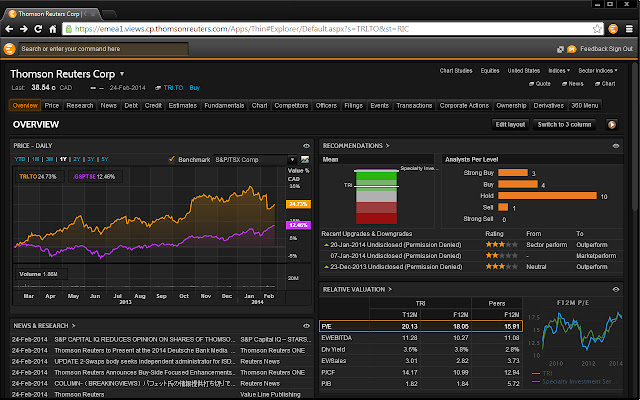


Figure : View of The Thomas Reuters Eikon Desktop Terminal Web Overview

Thomson Reuters’ Refinitiv Eikon, now part of the London Stock Exchange Group plc, stands as a notable competitor to the Bloomberg Terminal, particularly distinguished by its emphasis on fundamental analysis and news integration (London Stock Exchange Group plc, 2024) . The software provides a comprehensive suite of features, including real-time and historical market data spanning multiple asset classes. Its advanced analytics tools are designed to support fundamental analysis, offering insights into company financials, analyst ratings, and detailed industry reports (Thomson Reuters, 2024). Additionally, Refinitiv Eikon integrates news feeds from Reuters and other reputable sources, enhancing its utility for up-to-date financial information. The platform also includes collaboration features, facilitating communication and research sharing within teams (Davis & Wu, 2020).

However, Refinitiv Eikon faces limitations in educational contexts similar to those encountered with Bloomberg Terminal. The cost of subscription remains prohibitively high for many educational institutions, constraining its accessibility (Smith T. , 2019). Although Eikon’s interface is somewhat less complex than Bloomberg’s, it still presents a considerable learning curve for novice users (Jones & Brown, 2021). Furthermore, the software’s strong focus on fundamental analysis may not fully align with all financial literacy curricula, potentially limiting its applicability in diverse educational settings (Lee & Chen, 2018).

2.4.3 FactSet



Figure :View of FactSet Terminal

FactSet (FactSet, 2024) is another leading platform, predominantly serving institutional investors and wealth management firms with its extensive financial data capabilities. The platform provides comprehensive data coverage, including market data, company financials, economic indicators, and a range of alternative datasets. Its portfolio analysis tools are designed for constructing and evaluating investment portfolios, while its risk management functionalities assist in assessing and mitigating investment risks. Additionally, FactSet offers customization options, allowing users to tailor the platform to specific investment strategies (Smith & Patel, 2021).

Despite its robust features, FactSet presents several limitations for educational use. The platform's pricing structure is oriented towards professional users, making it prohibitively expensive for educational institutions (Johnson, 2020). Furthermore, FactSet's focus on portfolio management and investment analysis may not align with the broader objectives of financial literacy education (Williams & Edwards, 2019). The complexity of the platform, tailored for experienced financial professionals, can also pose a significant barrier to those new to financial analysis, making it less accessible for educational purposes (Lee & Thompson, 2022).

Table : Comparison between FactSet, Bloomberg and Thomas Reuter's Eikon

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Features** | **Infomark Web Application** | **Bloomberg Terminal** | **Thomas Reuters’ Eikon** | **FactSet** |
| **Data Coverage** | Real-time & Historical (customizable, to align with curriculum). Includes a future plan for Africa-related financial datasets | Real-time & Historical (extensive). Does not include Africa-related financial data sets, with no future plans for it. | Real-time & Historical (extensive). Does not include Africa-related financial data sets, with no future plans for it. | Real-time & Historical (extensive). Does not include Africa-related financial data sets, with no future plans for it. |
| **Analytics Tools** | Basic & Educational (focused on core financial concepts) | Advanced (extensive functionality for professional analysis) | Advanced (similar to Bloomberg) | Advanced (portfolio analysis, risk management) |
| **Ticker Symbol Search Engine** | Included | Not Available | Not Available | Not Available |
| **Application Type** | Web Application, with optional desktop terminal | Desktop Terminal | Desktop Terminal | Desktop Terminal |
| **Cost** | Freemium (free basic tier, affordable subscription) | Expensive Subscription | Expensive Subscription | Expensive Subscription |
| **User Interface** | Intuitive, Simple & Learner-Centric (designed for diverse skill levels, will have interactive tutorials) | Complex & Menu-Driven (geared towards experienced professionals) | Complex & Menu-Driven (similar to Bloomberg) | Complex & Customizable (tailored for professional workflows) |
| **Data Format** | Interactive tables, exportable to Microsoft Excel (tailored for learning). No complex charting | Advanced charting (technical & fundamental analysis). Not exportable to Microsoft Excel. | Advanced charting (similar to Bloomberg). Not exportable to Microsoft Excel. | Portfolio-centric visualizations. Not exportable to Microsoft Excel |
| **Analytics Tools** | Basic & Educational (focused on core financial concepts, easy to use) | Advanced (extensive functionality for in-depth analysis) | Advanced (similar to Bloomberg) | Portfolio analysis, risk management |
| **Installation and setup** | No installation needed. Accessed over the web. | Standalone hardware must be bought. | Accessed over the web. | Accessed over the web. |
| **Data Access and Integration** | Secure APIs, data cleaning & transformation for user-friendliness | Extensive data feeds, complex data manipulation tools | Extensive data feeds, similar data manipulation to Bloomberg | Proprietary data access & integration |

### 2.4.4 Why my solution is more superior to others for educational use

My web application excels in educational settings by aligning closely with financial literacy curricula, offering interactive tools and table visualizations that facilitate learning (Williams & Edwards, 2019). Its user-friendly web-based interface caters to all skill levels, overcoming the steep learning curves of professional platforms like Bloomberg and Eikon (Lee M.-J. , 2011). My web application is the only one integrated with Microsoft Excel, positioning itself as an educator-friendly, and student-friendly tool since it is interoperable with existing educational tools focused on data analysis.

With its cost-effective freemium model and affordable subscriptions, it is accessible to a broader range of institutions and students, which is really important (Demir & Demir, 2017). Furthermore, its streamlined API access and robust data security underscore its suitability for educational use, ensuring privacy and integration with existing learning systems, which is a very important features (Cumby R. E., 2018). Thus, the web application offers a superior, tailored solution for financial data analysis in educational contexts.

## 2.5 Conceptual Framework

2.5.1 Introduction

Having reviewed user needs, data acquisition challenges, and existing solutions, this section outlines the conceptual framework for the web application. This framework illustrates the system's components, their interactions, and how they address the identified gap in financial markets data analysis tools for educational settings.

2.5.2 Literature-Based Framework

The web application operates by first collecting user input through its intuitive interface, which allows users to select financial data, define timeframes, and choose analysis tools. It then processes this data using secure APIs from reputable sources, ensuring it is cleaned and formatted for analysis. The processed information is securely stored in a database for efficient retrieval and future use. In some cases though, the data is called directly and displayed, without storing it in the database so as to enable real-time data and prevent data staleness. Finally, the system produces a range of outputs, including statistical reports in Excel, and data tables to meet the diverse needs of users.

2.5.3 Stakeholder Roles:

The web application serves various stakeholders by enabling students to access financial data, perform analyses, and improve their financial literacy. Educators can use the platform to develop assignments, and integrate it into their curriculum. Financial data providers contribute by offering real-time and historical data through secure APIs, ensuring the web application’s functionality and data accuracy.

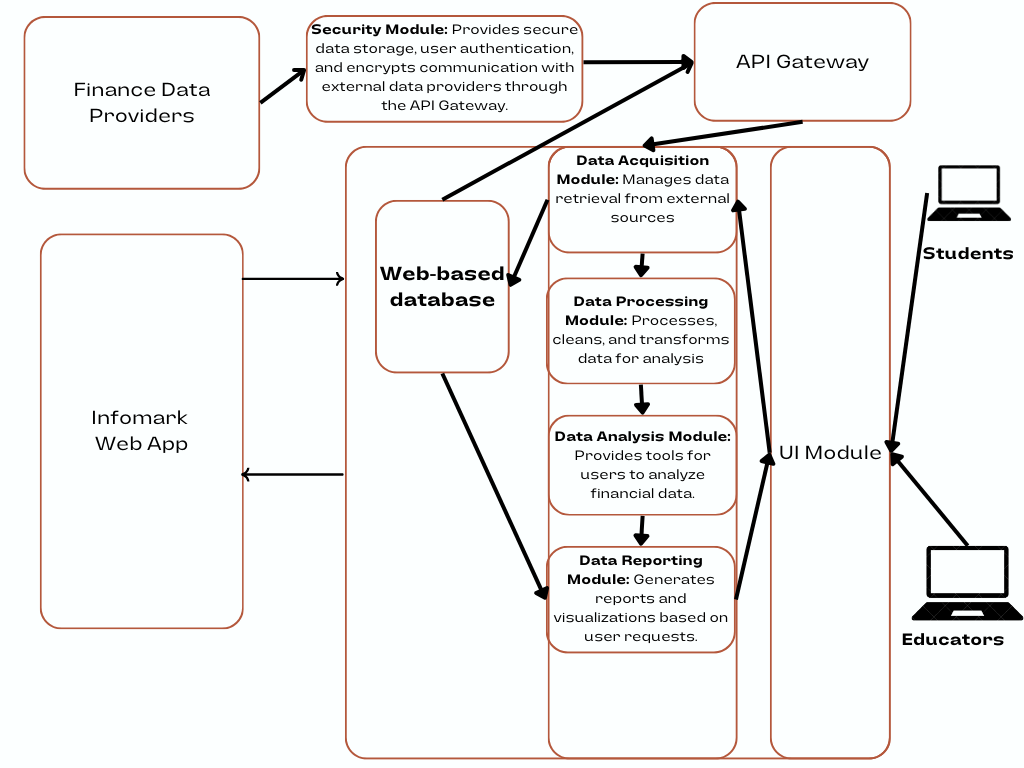


Figure : Conceptual Framework of the 'Infomark Terminal Web App'

The above diagram depicts the conceptual framework for the Infomark Web Application, visually representing the system's components, their interactions, and the roles of key stakeholders.

2.5.4 Boundaries and Emerging Technologies

The web application operates within a defined framework, where user input determines the scope of data retrieval and analysis. It emphasizes data security through secure APIs, encryption, and user authentication, while adhering to privacy regulations for data storage. Although it currently focuses on traditional financial data, its architecture is designed to accommodate future advancements. This includes the potential integration of alternative data sources, such as social media sentiment and satellite imagery, as well as AI technologies for automated analysis and personalized recommendations. This future-proof design ensures that the web application remains relevant as financial analysis practices evolve.

# Chapter 3: Research Methodology

## 3.1 Introduction

This chapter outlines the research methodology for developing the web application, a command-line tool for financial data access in education. Object-Oriented Analysis and Design (OOAD) was chosen over Structured Systems Analysis and Design (SSAD) due to its superior flexibility and modularity, which are essential for accommodating the dynamic nature of financial data and user interactions. OOAD facilitates iterative development and incremental improvements, aligning well with the evolving needs of educational users (Booch & Jacobson, 1999).

The Agile Development System Development Methodology is employed to support iterative progress and rapid adaptation based on user feedback. It has an emphasis on adaptive planning and frequent delivery which makes it suitable for integrating continuous user input and addressing emerging requirements (Beck, Beedle, & VanBaskerk, 2001). This methodology ensures that the development process remains responsive and user-focused, thereby enhancing the functionality and effectiveness of the web application

## 3.2 Research Design

The web application employs an experimental research design to assess the effectiveness of financial data analysis tools in educational settings. This approach involves collecting quantitative data on user interactions with the system, such as the frequency of tool usage, the accuracy of data analysis, and the impact on financial literacy. Key variables include user proficiency, the complexity of financial instruments, and system performance. This design facilitates a structured evaluation of how well the web application meets its educational objectives and supports data-driven improvements.

3.3 Agile Development System Development Methodology

The web application employs an Agile Development Methodology, specifically leveraging the Scrum framework. Scrum's iterative approach is ideal for this project, as it enables early feedback and continuous improvement, ensuring the platform evolves in line with user needs and educational requirements. By focusing on user stories, Scrum translates the needs of students and educators into actionable features, ensuring the system addresses core requirements effectively. Furthermore, Scrum's emphasis on rapid prototyping allows for the creation and testing of early versions of the web application, facilitating frequent stakeholder feedback and iterative enhancements to improve user experience.

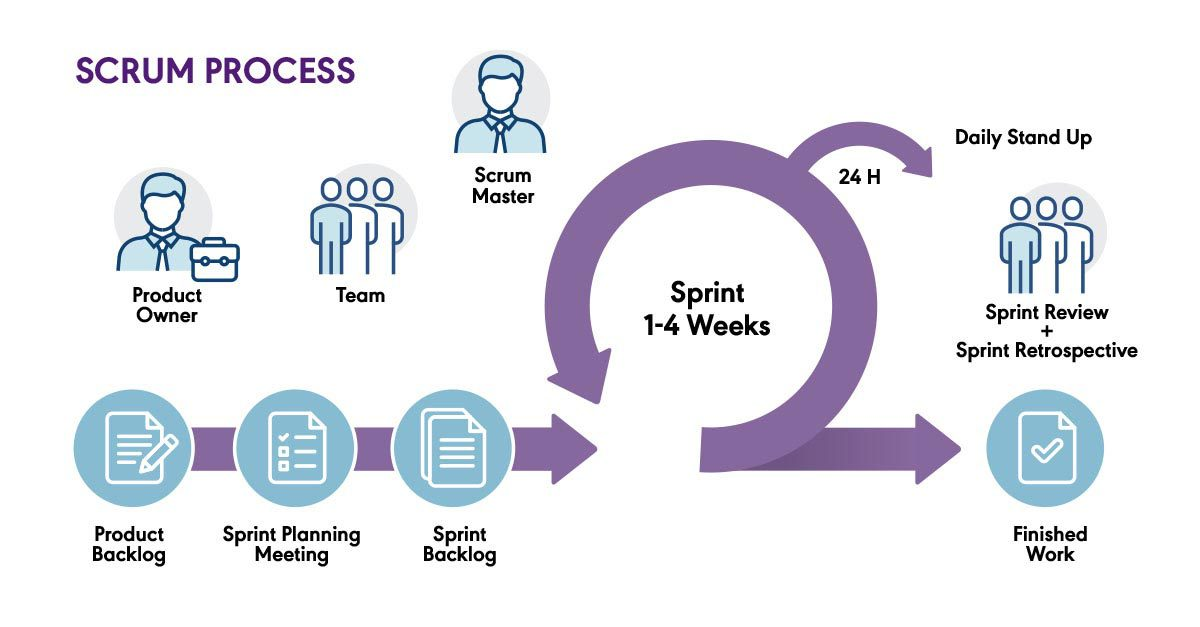


Figure : Agile Development Methodology using the Scrum Process

3.3.1 Ideation and Requirements Analysis Phase

The development of the web application starts with a comprehensive ideation and requirements analysis phase, which involves understanding the core problem and gathering detailed requirements. This is achieved through interviews and surveys with students and educators to identify their specific needs and challenges in financial data analysis via a command-line interface.

The analysis also includes reviewing educational curricula to highlight key financial concepts and data analysis skills. The collected data helps define both functional and non-functional requirements, with functional needs identified through user workshops and translated into actionable tasks via user stories and Scrum sprints. A review of existing financial data analysis tools guides the design of the web application, while usability testing and security audits ensure the interface is user-friendly, well-documented, and compliant with data protection regulations.

3.3.2 Design Phase

In the design phase, various diagrams and system architecture are developed to guide the creation of the web application. Key design artefacts include a Use Case UML diagram, which illustrates the interactions between users and the system's functionalities as accessed through commands. A simplified Class UML diagram is produced to represent the primary objects and their attributes relevant to the system’s functionalities.

Design tools such as Draw.io are employed to create these diagrams. The system architecture is structured into a layered framework comprising three main components: a Web Interface, which serves as the medium for user interaction via text commands, and buttons; the Business Logic Layer, which encompasses core functionalities for data processing and analysis; and the Data Access Layer, responsible for interfacing with the database and external financial market data APIs.

3.3.3 Development Phase

The development phase leverages the Agile methodology, specifically employing the Scrum framework, to facilitate iterative and incremental progress. Choosing Python as the programming language for building the web application, and using Conda (Miniconda) to manage the development environment, addresses key needs. Conda offers the advantage of creating isolated environments, helping to avoid conflicts with other installations and ensuring a stable development setup. Simplifying package management, it allows for easily installing and updating libraries necessary for the web application’s functionalities.

Setting up involves installing Miniconda (Anaconda, Inc., 2024), creating a Conda environment tailored to the project, and installing required libraries such as Pandas and NumPy. Utilizing development tools like Visual Studio Code for code development and debugging, GitHub for code hosting and collaboration, Git for version control, and Gitpod for rapid development tasks on the web, supports the development process.

3.3.4 Testing Phase

In the testing phase, the system undergoes rigorous evaluation to ensure its reliability and effectiveness. Testing employs a combination of black-box and functional methods. Using the black-box testing approach, evaluators assess the system’s functionality without knowing its internal workings. Providing inputs and observing outputs, testers assess how the system responds to both expected and unexpected user actions. This approach, allowing non-technical users to participate in testing, ensures that the system remains user-friendly and meets its design specifications (Pressman, 2019). Python’s pytest is utilized for conducting unit tests. Additionally, focusing on evaluating the system’s response time, usability, and overall performance confirms that it meets the intended requirements and operates effectively across different platforms (Sommerville, 2016).

3.4 System Implementation

After the development of the web application, the system was deployed and rigorously tested to ensure its functionality and reliability. The implementation involved configuring the web application and integrating it with the database and Python libraries. The system underwent thorough testing to verify performance, compliance with requirements, and overall effectiveness. Routine maintenance was carried out to ensure continued stability and prevent significant issues.

3.5 System Architecture and Modules

3.5.1 Data Acquisition Module

The Data Acquisition Module is responsible for interfacing with various financial data sources. It facilitates the retrieval of data from external APIs, such as Alpha Vantage and Yahoo Finance. This module ensures data is fetched, updated, and stored accurately within the system, providing a reliable foundation for subsequent analysis. Users can initiate data downloads and configure data sources through the command-line interface.

3.5.2 Data Cleaning and Pre-processing Module

This module handles the cleaning and preparation of raw financial data. It incorporates Python libraries like Pandas. It processes and formats the data into Pandas Dataframes to ensure it is suitable for analysis. Tasks include removing inconsistencies, handling missing values, and structuring data according to predefined criteria, which is essential for accurate analysis and reporting.

3.5.3 Data Processing and Analysis Module

The Data Processing and Analysis Module handles the transformation and examination of financial data. It leverages Python libraries like Pandas, Streamlit and NumPy to clean, pre-process, and analyse the data according to user queries and requirements. This module is central to generating insights and performing complex calculations, and it supports various financial analysis tasks as specified by the user.

3.5.4 Report Generation Module

The Report Generation Module produces textual reports based on the analysis performed in the previous module. It formats the results into user-friendly text-based reports that interoperable with Microsoft Excel. This module allows users to generate and review reports on financial market analysis outcomes, providing a clear summary of findings directly from the web application.

3.6 System Deliverables and Milestones

This describes the process of creating the system as well as the dates for the milestones of completion. It also gave an estimation as to when the system should be ready. It shows the flow of how the system will be implemented.

3.6.1 Design Diagrams

The first deliverable involves creating design diagrams that visually represent the system’s functionality. These diagrams will include a Use Case diagram illustrating user interacts with the application, a System Architecture diagram illustrating the high-level interaction between key components in the web application, a Wireframe showing the GUI interactions for the tables and reports generated by the system, a Class UML Diagram showing how the classes and objects in the code are going to be created, initialized and handle data within the application, an Entity Relationship Diagram that shares how data is stored and represented in the web application, and finally a Sequence UML Diagram showcasing the behaviour aspect of the web application. All these diagrams will be used as blueprints for developing the web application, and are to be completed by June 5th, 2024.

3.6.2 Database Connectivity

This includes designing and creating the SQLite database based on the previously developed design diagrams. The database must be capable of efficiently handling data storage, retrieval, and updates. Data is going to be cached on the host’s file system. This deliverable is vital for the data management aspect of the system and is targeted for completion by June 28th.

3.6.3 User Interface Completion

This phase involves defining the layout and structure of the command-line interface, including command prompts, chart displays, data reports, and user feedback mechanisms. Ensuring that the interface remains intuitive and user-friendly for all user roles is essential. Completing the user interface is scheduled for June 25th 2024.

3.6.4 System Modules Development

This involves developing several key modules, such as the Data Download Module, Data Cleaning and Pre-processing Module, Data Analysis Module, and Report Generation Module. Each module is being created to fulfil specific functionalities necessary for the system's operation. Completing this phase is expected by July 10th.

3.6.5 System Testing

This deliverable encompasses rigorously testing using both black-box and functional testing methods. This phase involves evaluating the system's performance, usability, and compliance with requirements, documenting test results, and addressing any identified issues before the final showcase on August 20th.

3.6.6 Final Documentation

This document provides comprehensive information on the system's functionalities, development processes, and key features. It includes user guides and technical documentation for future reference and maintenance. The final documentation is being completed by July 19th.

3.6.7 System Completion and Demo

The seventh deliverable marks the system completion, showcasing the full functionality of the web application. This milestone demonstrates how the system responds to user commands and processes inputs, ensuring that all features are operational and meet the specified requirements just before the final demo to the examiners. The system is being completed by July 24th, and the demo is being pushed to August 20th.

# Chapter 4: System Analysis and Design

## 4.1 Introduction

This chapter delves into the technical aspects of the web application, including its core functionalities, architecture, and development plan. It details the system's scalable framework, secure data acquisition methods, and user-friendly command-line interface design. By the end of the chapter, a clear understanding of the web application's technical foundations will be established, showcasing its effectiveness in addressing the financial literacy needs of educators and students.

## 4.2 System Analysis

In this section, we will identify the functional and non-functional requirements of the web application. The analysis will highlight the essential capabilities the system must possess, as well as the attributes that ensure its effectiveness and reliability.

4.2.1 Functional Requirements

These functional requirements establish a clear understanding of the system's core functionalities, ensuring the web application delivers the intended value proposition for financial literacy education. The key requirements are:

1. The system shall retrieve financial data from external sources using APIs or scraped data feeds.
2. The system shall clean and pre-process acquired data, including handling missing values and validating data integrity.
3. The system shall offer a user-friendly web application interface that accepts user input through text commands, and button triggers for data retrieval, analysis, and visualisation, while providing clear output messages and error handling for invalid commands.
4. The system shall generate text-based visualisations of data, such as tables, to enhance user understanding.
   * 1. Non-Functional Requirements

Non-functional requirements specify the quality attributes of the system, encompassing aspects such as performance, security, and usability. The primary non-functional requirements are:

1. The system shall implement robust authentication mechanisms to protect user data and ensure secure access for all users.
2. The system shall efficiently handle data processing and analysis tasks, providing timely responses to user queries to enhance user experience.
3. The web application interface shall be intuitive and easy to navigate, facilitating a positive user experience for educators and students alike.
4. The system architecture shall support the integration of additional functionalities and data sources without significant reconfiguration, allowing for future enhancements.
5. The system shall be designed for easy maintenance, enabling updates and modifications with minimal disruption to users

## System Design

System design is a critical phase in the development of the web application, as it involves defining the components, modules, interfaces, and data structures necessary to satisfy the specified requirements. This process ensures that the system is robust, scalable, and user-centric. Key design methodologies employed in this project include the creation of use case diagrams and sequence diagrams, which serve to clarify the system's functionalities and interactions.

* 1. System Analysis

System analysis involves understanding the needs of the users and translating them into functional requirements for the web application. This phase is essential for ensuring that the system aligns with user expectations and effectively addresses financial literacy challenges. While various diagrams can facilitate this analysis, the choice of specific modelling techniques depends on the nature of the application being developed. The analysis phase focuses on capturing both the high-level business processes and the detailed interactions that will occur within the system

* 1. System Analysis Diagrams
     1. Use Case UML Diagram

Use case diagrams are a cornerstone of system analysis, providing a visual representation of a system's functionalities and user interactions (Rosenblum & Jackson, 2005)These diagrams depict actors (users or external systems) interacting with use cases, which represent specific functionalities offered by the system. The components of a standard Use Case UML Diagram include actors (Rosenblum & Jackson, 2005), which are external entities (users, other systems) that interact with the system. It also includes use cases (Rosenblum & Jackson, 2005), which are ovals representing functionalities or services provided by the system, and finally it also includes relationships (Rosenblum & Jackson, 2005), which are lines connecting actors and use cases, indicating how actors interact with functionalities. Different types of relationships can be used, such as, includes (one use case incorporates another) or extends (one use case adds functionality to another).

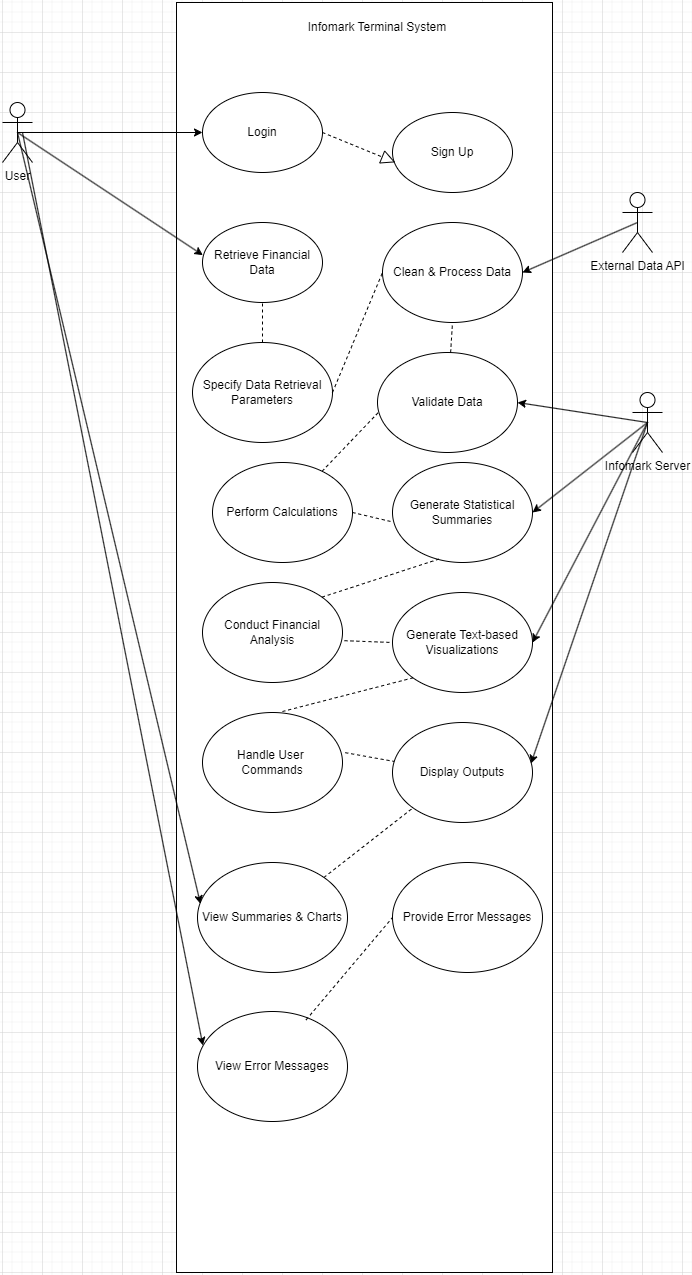


Figure : Use Case UML Diagram for Informark

We have 3 actors – User, External Data API, and Server. The User Actor interacts with the following use cases – Login, View Summaries & Charts, and ‘View Error Messages’

External Data API interacts with the following use case – ­Acquisition, Cleaning and Processing of data.

The Server interacts with the following use cases – Validate Data, Generate Statistical Summaries, Generate Text-based Visualizations, and finally, Display Outputs

4.5.2 Class UML Diagram

Class UML diagrams are a cornerstone of object-oriented system design, providing a visual representation of the system's classes, their attributes, operations (methods), and the relationships between them (Fowler, 2003). These diagrams serve as blueprints for translating system requirements into object-oriented code.

Components of a Class UML Diagram include a class, which is a rectangle representing a concept or entity within the system that encapsulates data (attributes) and behaviour (operations) (Fowler, 2003). It also includes attributes, which are listed within the class rectangle, representing the data properties of the class (Fowler, 2003). Operations are also components of a Class UML Diagram, and are listed within the class rectangle, representing the actions or methods that the class can perform (Fowler, 2003), and finally, relationships, which are lines connecting classes, indicating how the classes relate to each other. Common relationships include inheritance (a subclass inherits attributes and methods from a parent class), association (classes have a connection but no inheritance), and aggregation (a whole-part relationship where one class is composed of other classes) (Fowler, 2003).

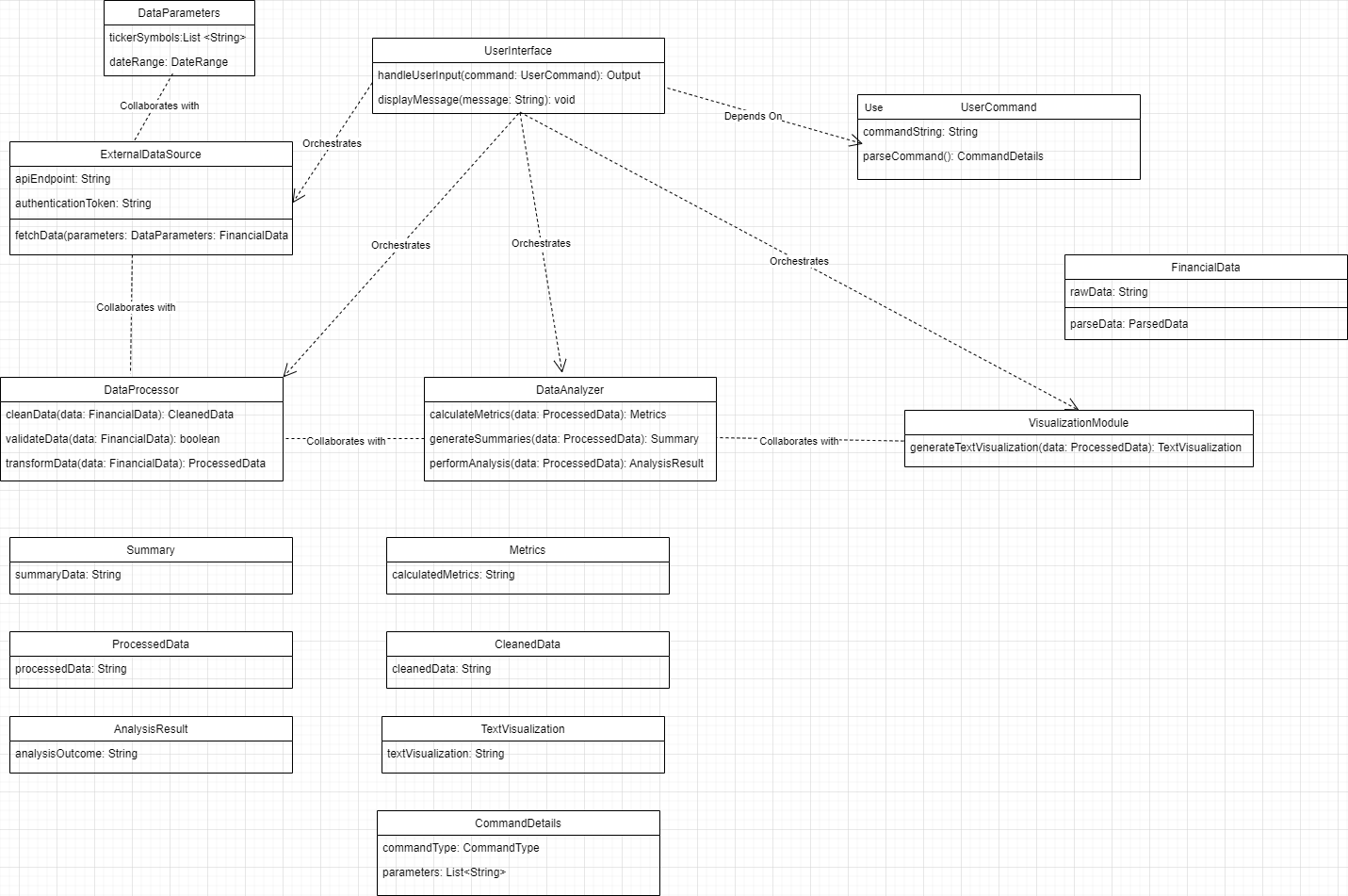


Figure :Class UML Diagram for Infomark Terminal Web App

* + 1. Entity Relationship Diagram

Entity relationship diagrams (ERDs) are a fundamental tool in data modelling, providing a visual representation of the entities (data objects) within a system and the relationships between them (Elmasri & Navathe, 2016). ERDs are particularly useful for database-driven systems where understanding data relationships is crucial for data integrity and efficient querying. Components of an ERD include – An Entity, which is represented by a rectangle, an entity represents a real-world object or concept with a set of attributes that describe its characteristics (Elmasri & Navathe, 2016). Examples in the web application could be entities like "Stock" or "User". Another component is the attribute, which is listed within the entity rectangle. Attributes represent the specific data points associated with an entity (Elmasri & Navathe, 2016). Examples of attributes include "Stock Ticker Symbol" or "User ID". Another component is the relationship, which are lines connecting entities, indicating how the entities are related (Elmasri & Navathe, 2016). Common relationships include one-to-one (one entity instance relates to one instance of another entity), one-to-many (one entity instance relates to many instances of another entity), and many-to-many (many entity instances relate to many instances of another entity). Cardinalities (numbers of instances involved) are often specified on the relationship lines.

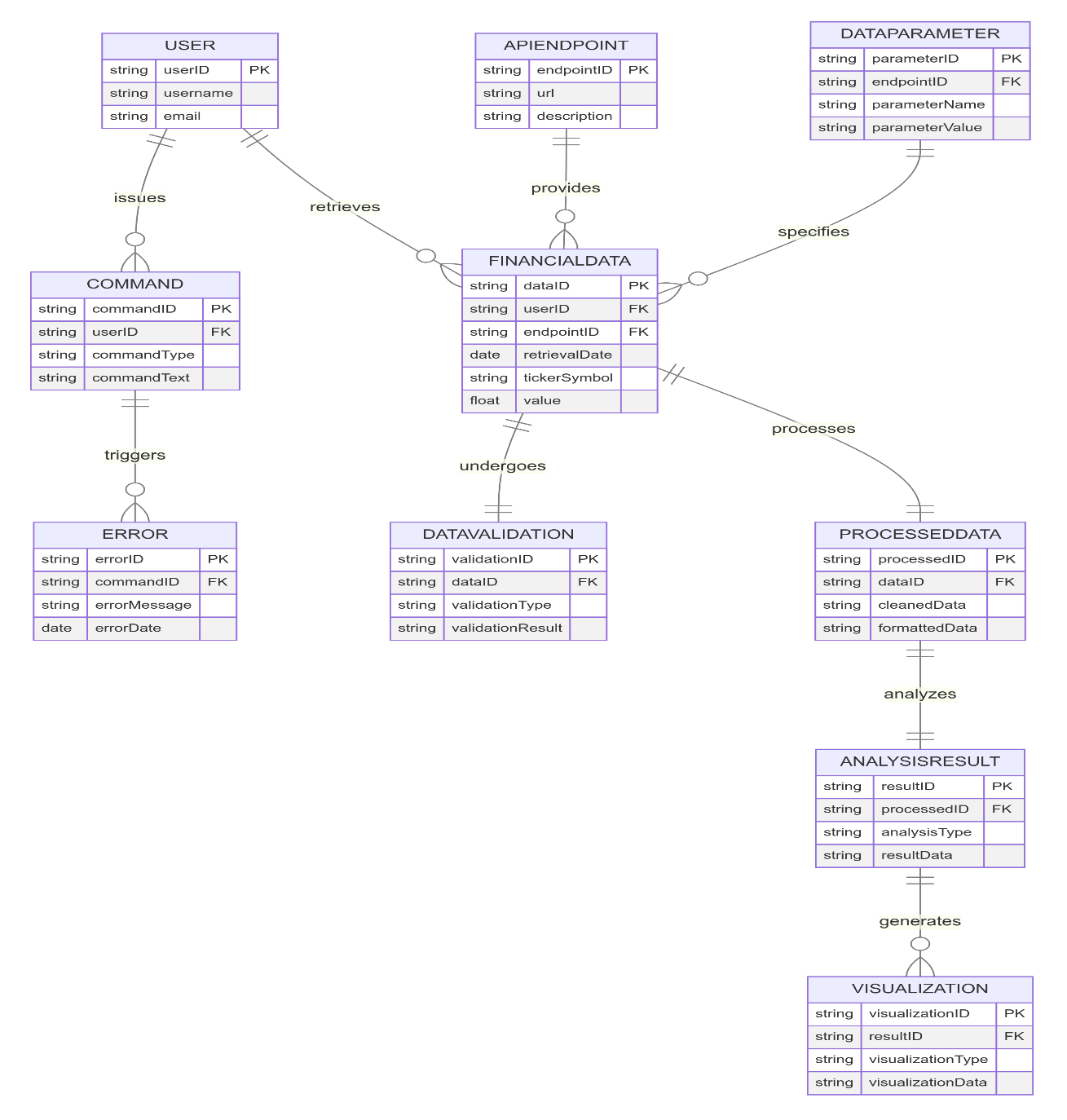


Figure : ER Diagram for Infomark

4.5.4 Sequence Diagram

Sequence diagrams are a type of UML (Unified Modelling Language) diagram used to visualize the interaction between objects in a system over time (Bruegge & Berndt, 2016). They are particularly useful for depicting the sequence of messages exchanged between objects during a specific scenario or use case. The key components of a sequence diagram include participants, messages, activation bars, and return messages (Bruegge & Berndt, 2016). Participants, represented by vertical lifelines, are the objects that take part in the interaction. These can be system objects, external actors, or other internal objects. Messages, depicted as arrows directed between lifelines, represent the communication between objects, such as method calls, data transfers, or event notifications. Activation bars, narrow rectangles placed on lifelines, indicate the period during which an object is actively processing a message. Return messages, an optional component, are arrows returning from called objects to calling objects. These represent return values or responses to messages (Bruegge & Berndt, 2016). By employing sequence diagrams, the web application can effectively illustrate the dynamic interactions that occur within the system, helping to ensure that the system operates as intended and meets the needs of its users.

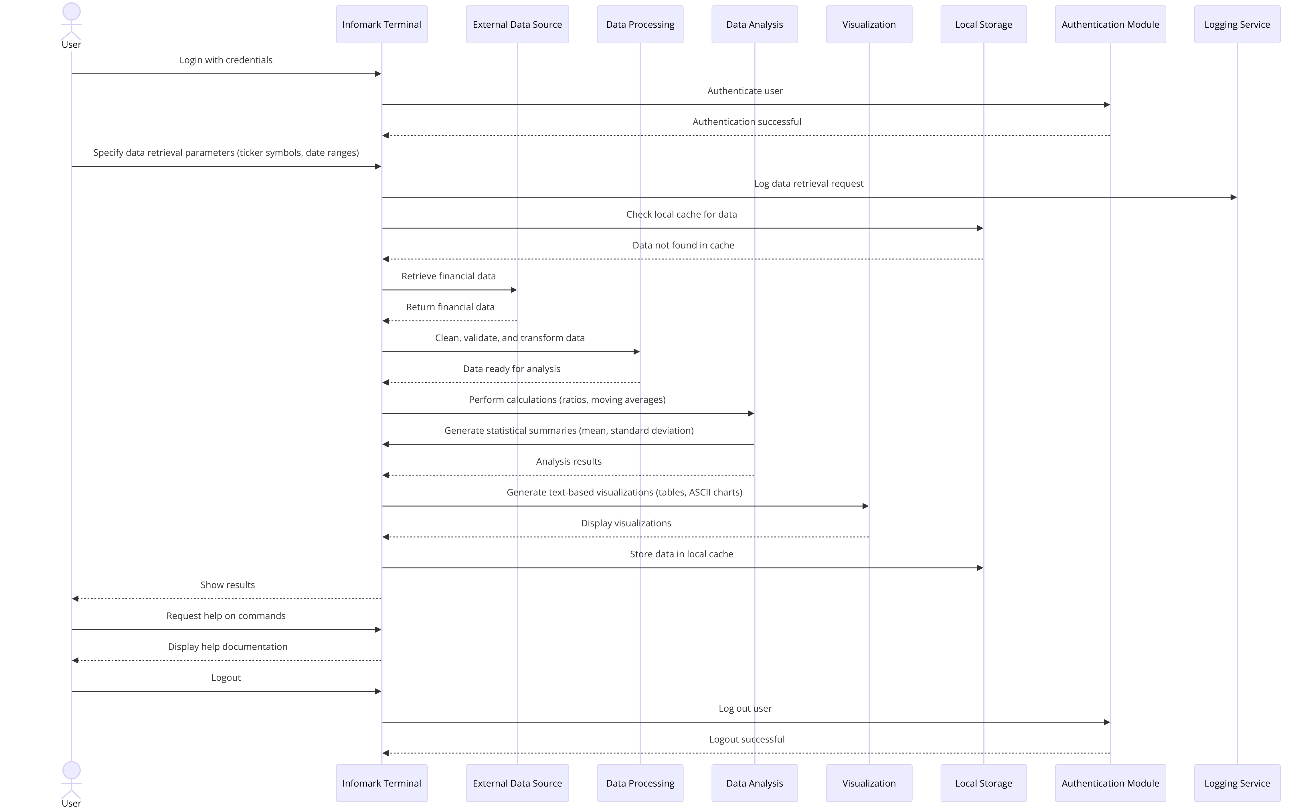


Figure : Sequence UML Diagram for Informark Terminal Web Application

4.5.5 Wireframes

Wireframes are a fundamental aspect of user interface design that serve as a blueprint for the layout and functionality of a digital product. They provide a visual guide that outlines the structure of a user interface without the distraction of design elements such as colors, fonts, and images. According to Nielsen and Budiu (Nielsen & Budiu, 2013), wireframes allow designers to focus on functionality and user experience without being distracted by visual design elements. This approach is crucial for ensuring that the user interface meets the requirements of its users and stakeholders. Furthermore, (Skarstein, 2014) highlights that wireframes are essential for iterative design, enabling teams to test and refine ideas quickly and cost-effectively. By creating wireframes, designers can establish a clear, shared vision of the product, facilitating communication and alignment among project team members.

Below are the important wireframes for the app:

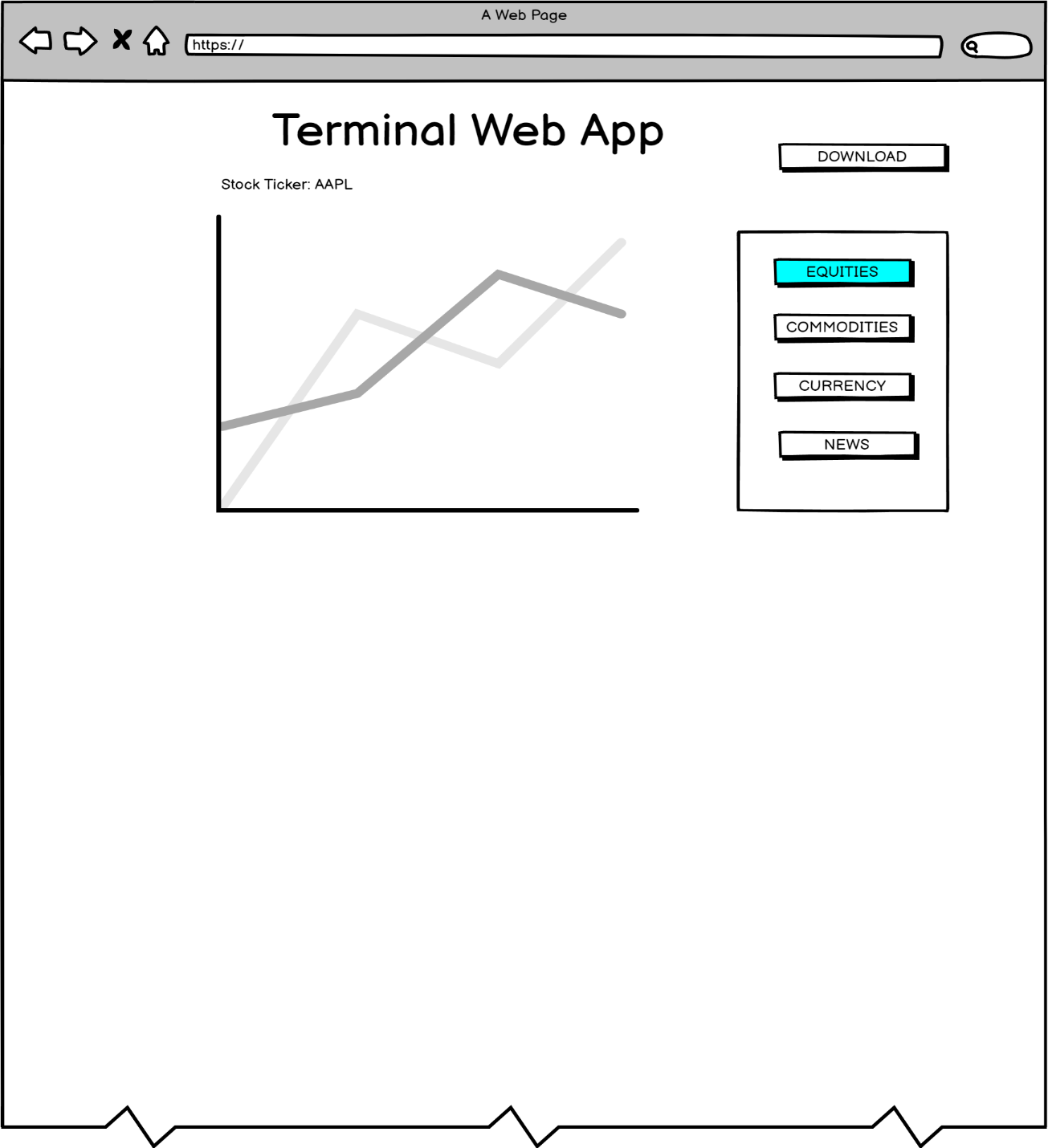


Figure : Infomark Terminal Web App Home Page

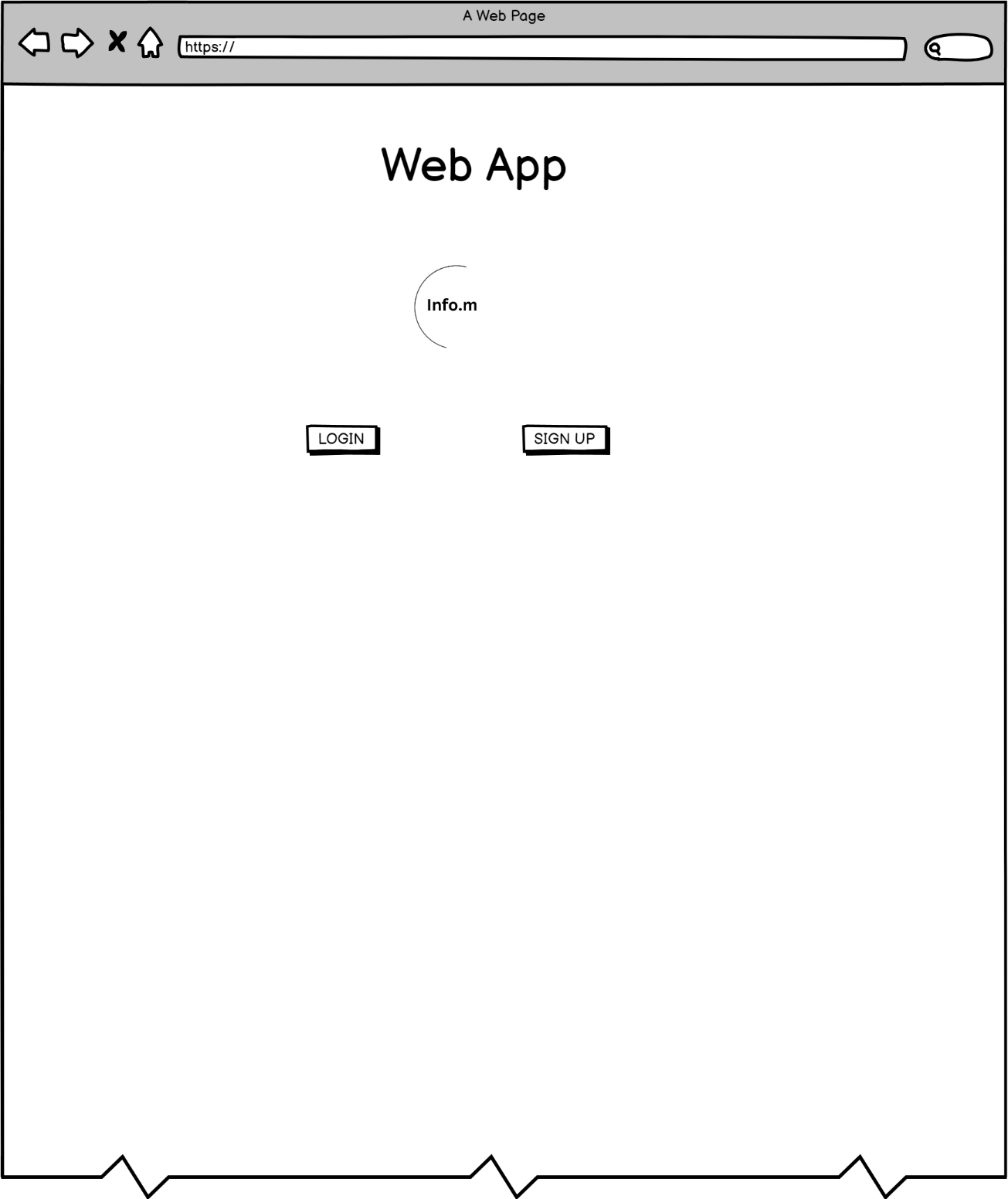


Figure :Web App Home Page

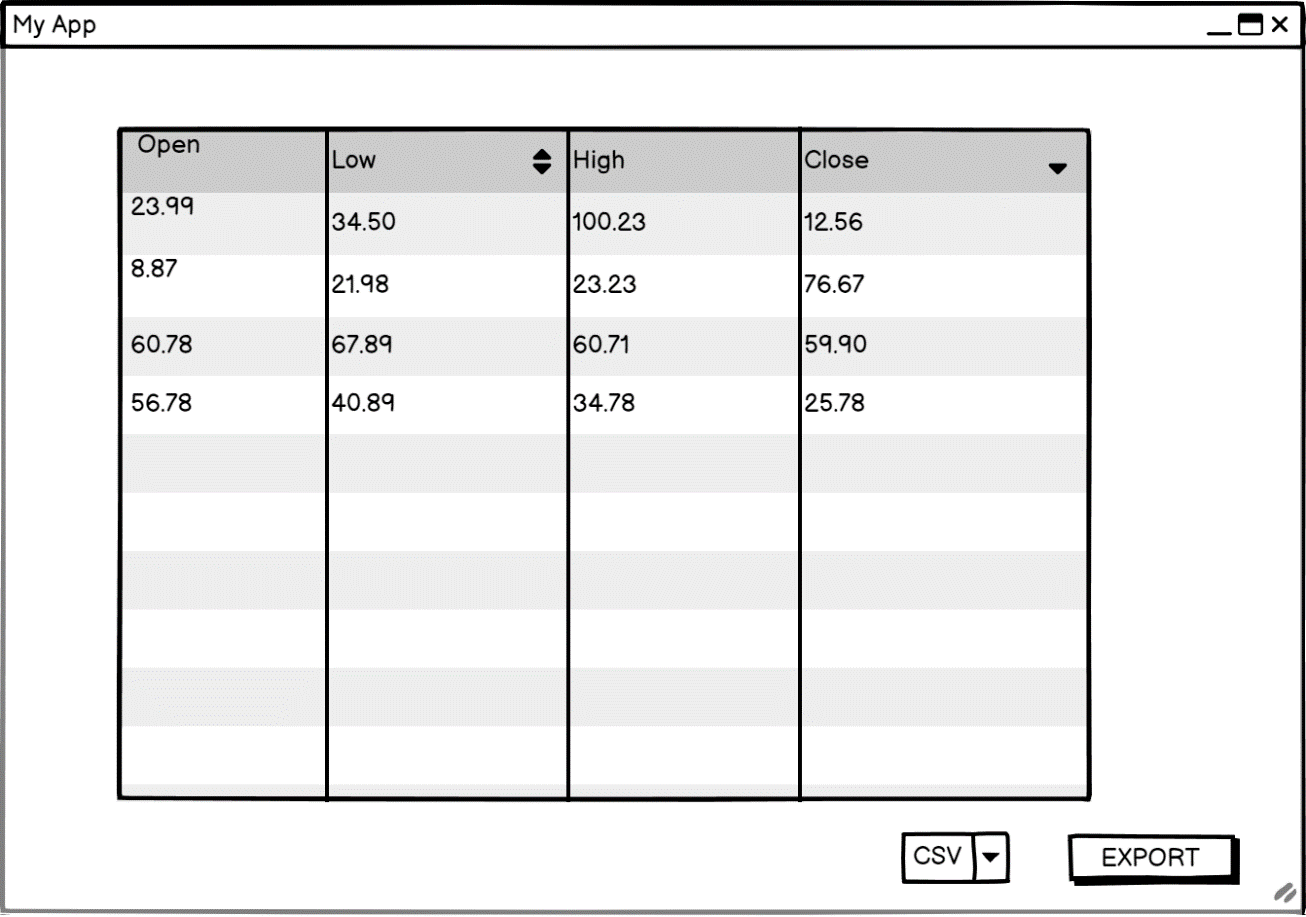


Figure :Web App Table Display

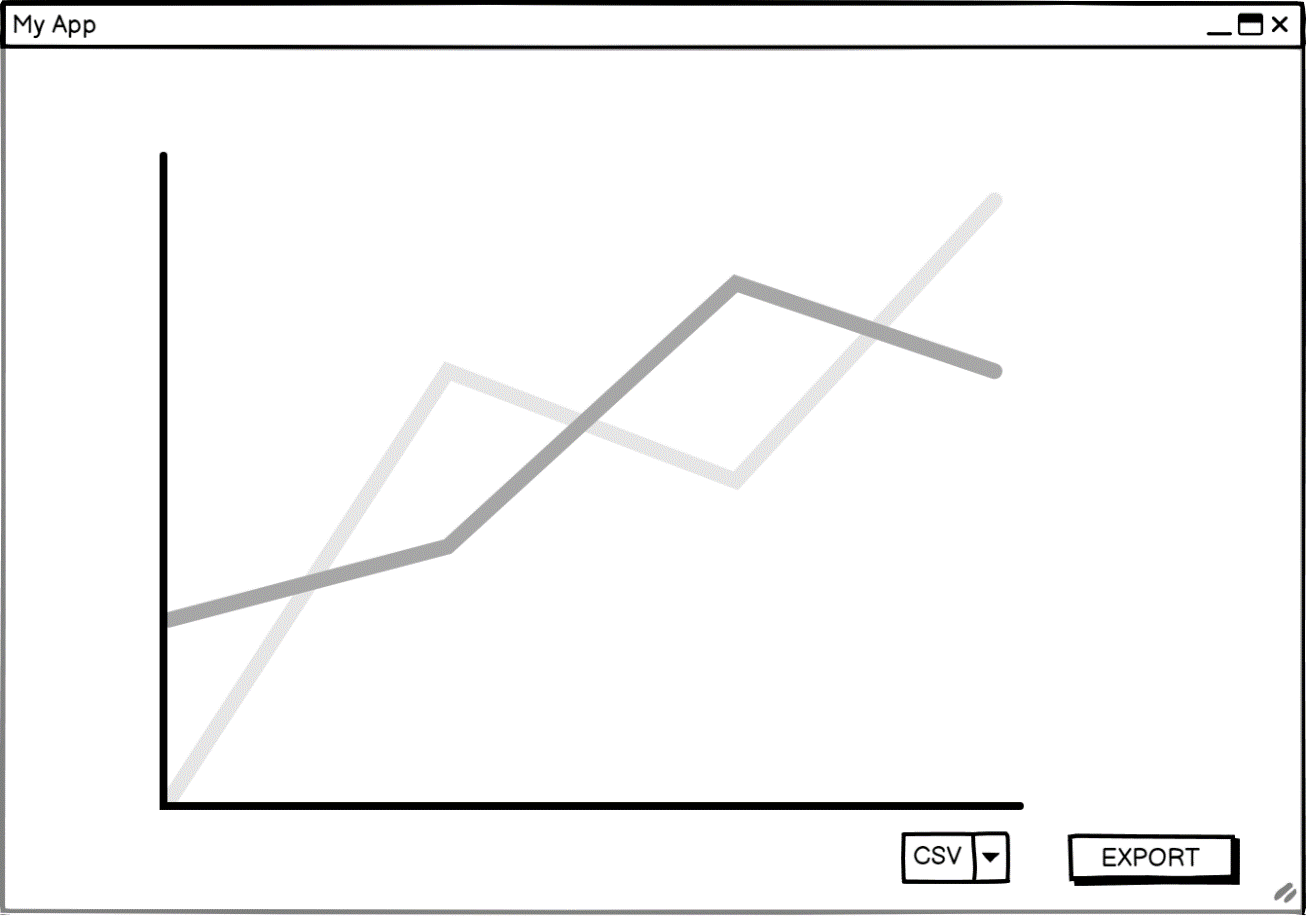


Figure : Web App Graph Display

4.5.6 System Architecture

System architecture outlines the high-level structure of a software system, defining its components, modules, and their interactions. For the web application, a modular architecture is proposed, dividing the system into functional units.

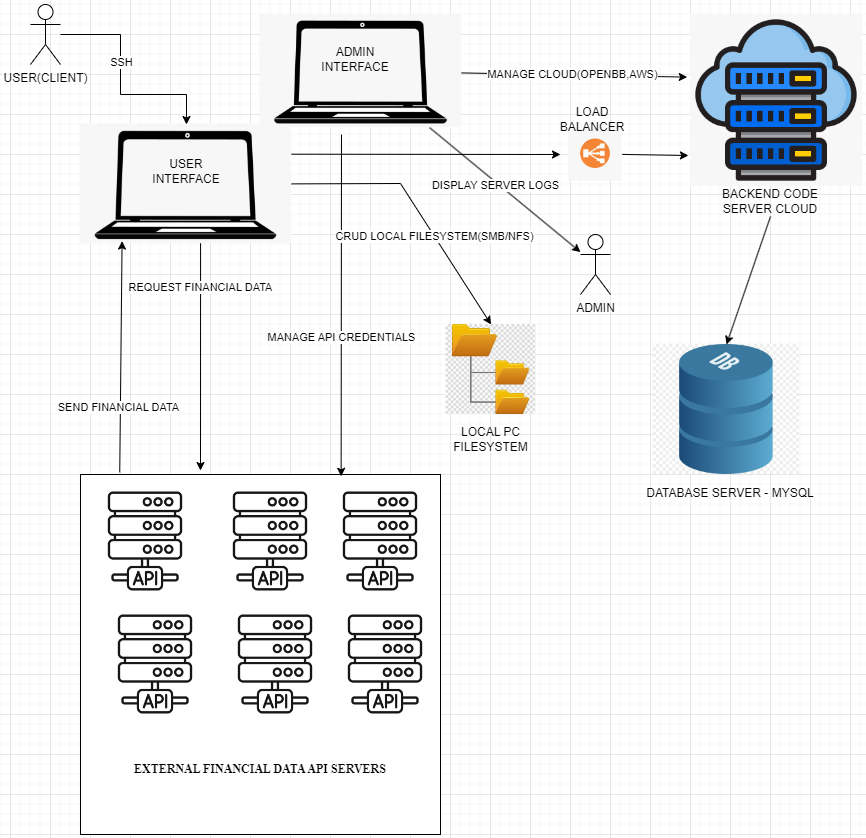


Figure : Simple System Architecture Diagram

4.5.7 Component UML Diagram(Specific To The Web Application)

**\*EXTRA USEFUL DIAGRAM, PLEASE CONSIDER**

A component diagram is a static structure diagram that shows the components of a system and their relationships (Rumbaugh, Jacobson, & Booch, 2004). It is primarily used to model the physical aspects of a system, focusing on the software components and their dependencies.

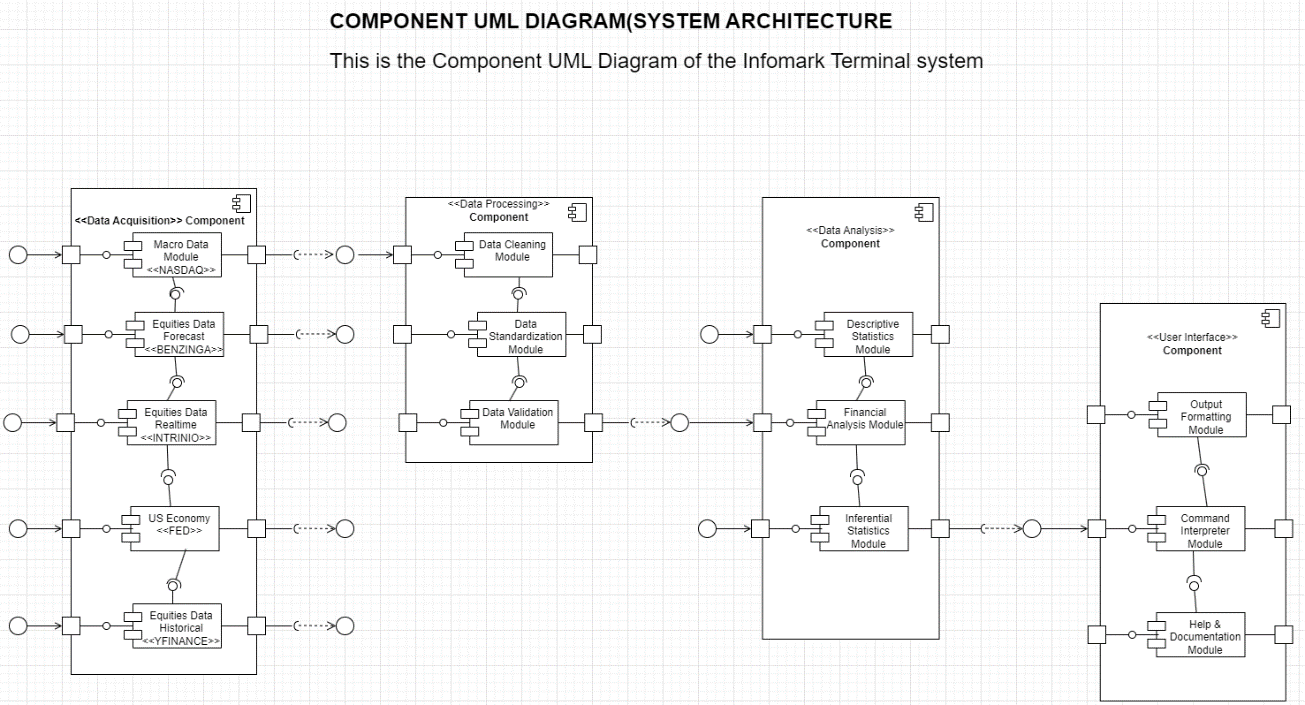


Figure : Component UML Diagram(A type of System Architecture Diagram)

# Chapter 5: System Implementation Testing

## 5.1 Introduction

System implementation testing refers to the process of evaluating the implementation of system specifications to ensure that they are both practical and conform to the established requirements. This process serves to verify that the system functions as intended and meets the specified criteria, thereby enhancing the overall quality and interoperability of the implementation. In this chapter, we will describe the development of the web application, detailing the technologies used and the testing conducted to ensure its effectiveness.

## System Implementation

The system is implemented, and its version 1 is published on Github here: <https://github.com/kcelestinomaria/infomark-web-app> .

### 5.2.1 Systems Backend

The backend of the web application is designed to efficiently manage data processing and analysis.

Built primarily in Python, the backend leverages a modular architecture that allows for flexibility and scalability. This design choice facilitates the integration of various functionalities, flexibility and scalability. This design choice facilitates the integration of various functionalities, ensuring that the system can adapt to evolving user needs and technological advancements.

Here is sample code for the cookies(browser-based storage) schema in YAML(Yet Another Markup Language) format. It is also used for database storage:

“

credentials:

  usernames:

    jsmith:

      email: celestine.kariuki@strathmore.edu

      failed\_login\_attempts: 0 # Will be managed automatically

      logged\_in: False # Will be managed automatically

      name: Celestine Kariuki

      password: abc # Will be hashed automatically

    rbriggs:

      email: examiner127@gmail.com

      failed\_login\_attempts: 0 # Will be managed automatically

      logged\_in: False # Will be managed automatically

      name: Mister Examiner

      password: youpass # Will be hashed automatically

cookie:

  expiry\_days: 30

  key: some\_signature\_key # Must be string

  name: some\_cookie\_name

pre-authorized:

  emails:

  - kcelestinomaria127@gmail.com

“

The system extensively implements and interface with various financial data API providers such as: OpenBB Platform, Benzinga, Intrinio, Yahoo Finance, Financial Modelling Prep, Alpha Vantage, BizToc, Central Bank of England API, European Central Bank API, EconDB, US Financial Regulatory(FINRA) Data, and Yahoo Finance.

Considering that we are fetching API data from the backend, we use a JSON file to structure our code and enable it easily interface and communicate with the external systems in these APIs:

“

{ "packageName": "-benzinga", "optional": false, "reprName": "Benzinga", "description": "Benzinga is a financial data provider that offers an API focused on information that moves the market.", "credentials": [ "benzinga\_api\_key" ], "website": "https://www.benzinga.com" },

“

### 5.2.2 System’s Frontend

The frontend is implemented using Python’s Streamlit library for both developer and user friendliness. Incorporated in chunks of the Streamlit integration are HTML, and CSS libraries embedded in Markdown.

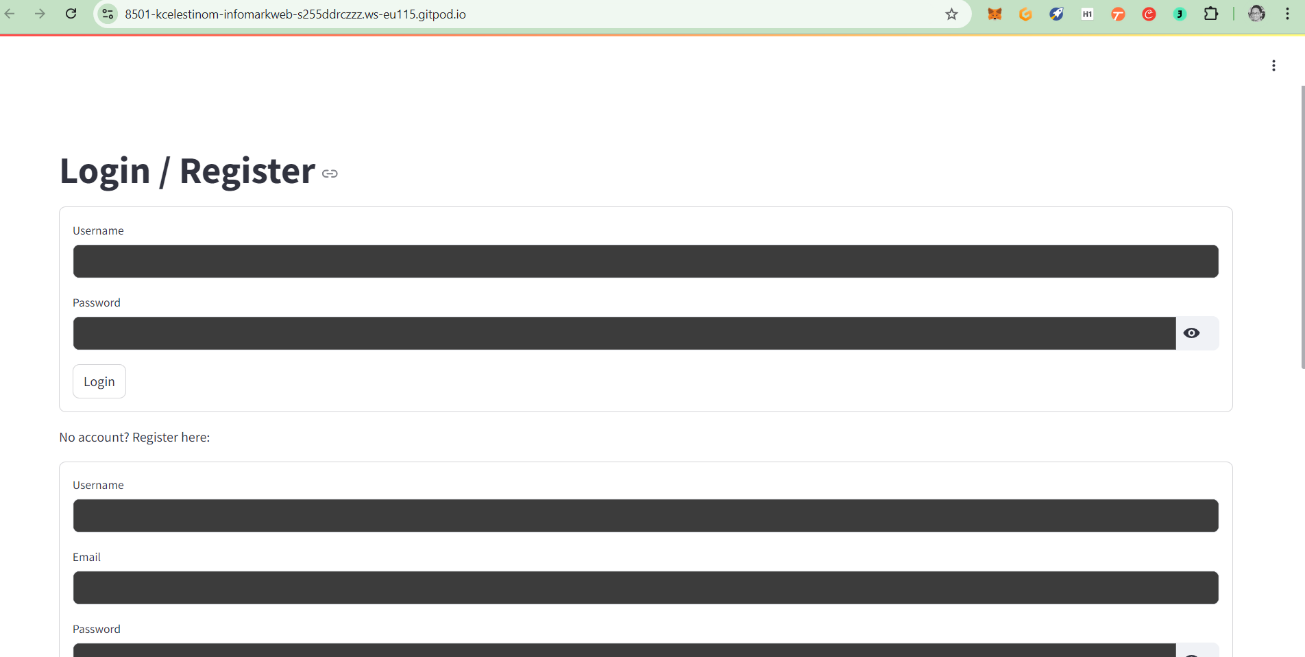


Figure : Infomark Login/Register Page

The first process involves the registration process and after successful registration, you proceed to login and access the system.

Let’s say, I want to get Apple’s historical and current stock price as an ordered table, so I can export and use it for my Business Statistics assignment, performing correlation analysis as an example. I can do this on the sidebar of the home page:

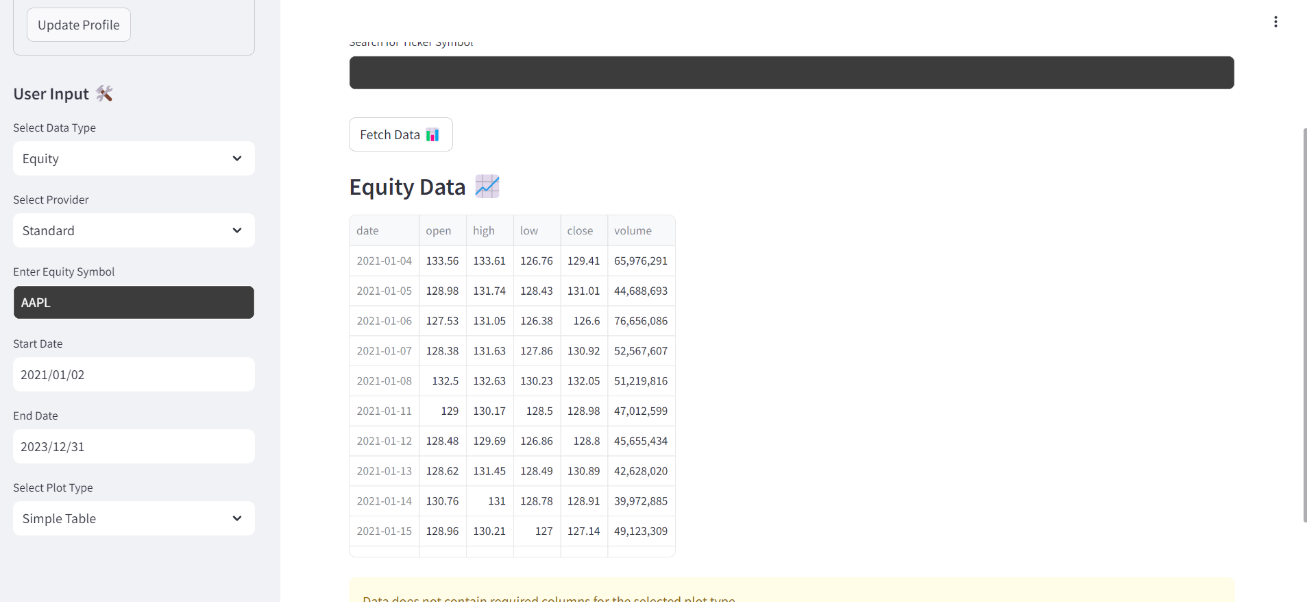


Figure : Apple's Historical Open, Close, High and Low Prices generated

What is important, is that a user should be able to search for a company and get its ‘stock ticker symbol’, which is used to represent the company’s shares on public financial markets. A search engine is built on the frontend to enable this, as seen in the image below:

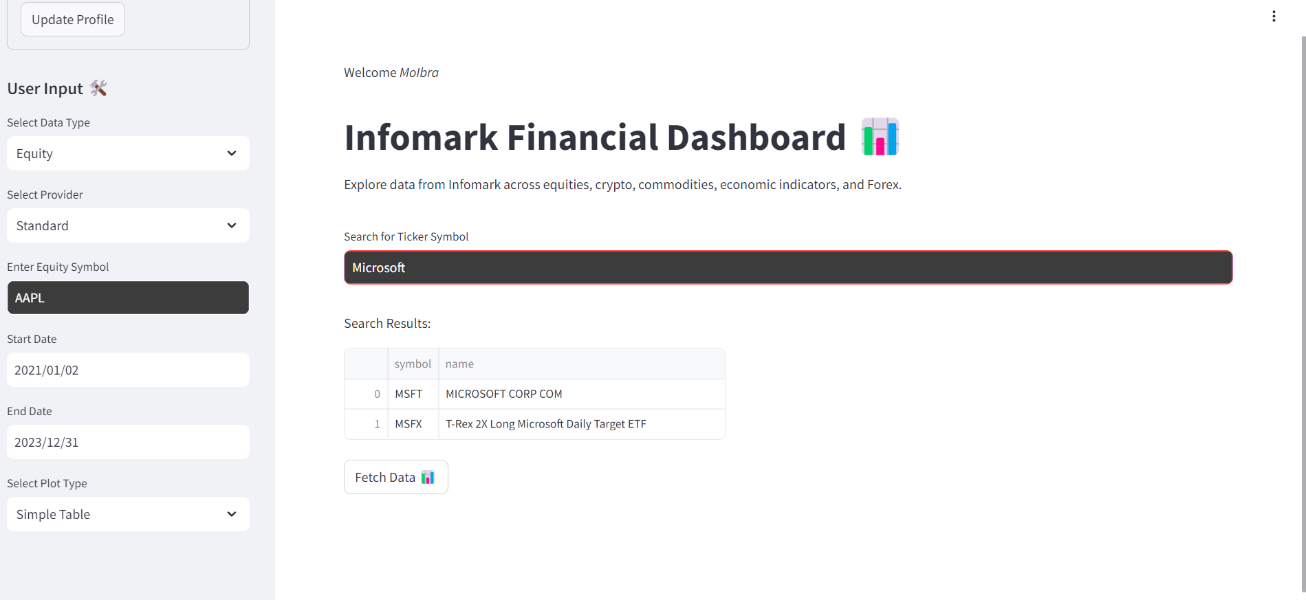


Figure : Searching for Company Ticker Symbols from the SEC

5.3 System Manual

The web application provides users with a web application and an optional command-line interface (CLI) to interact with the system. We will only cover the web application, which is more user-friendly for students and educators.

* + 1. Generating Tables

The system can also generate tables really well. You only need to set the date range of the asset’s price history you want to see the data for, enter the asset’s ticker symbol, all done on the sidebar, and finally click the ‘Fetch Data’ button, see below.

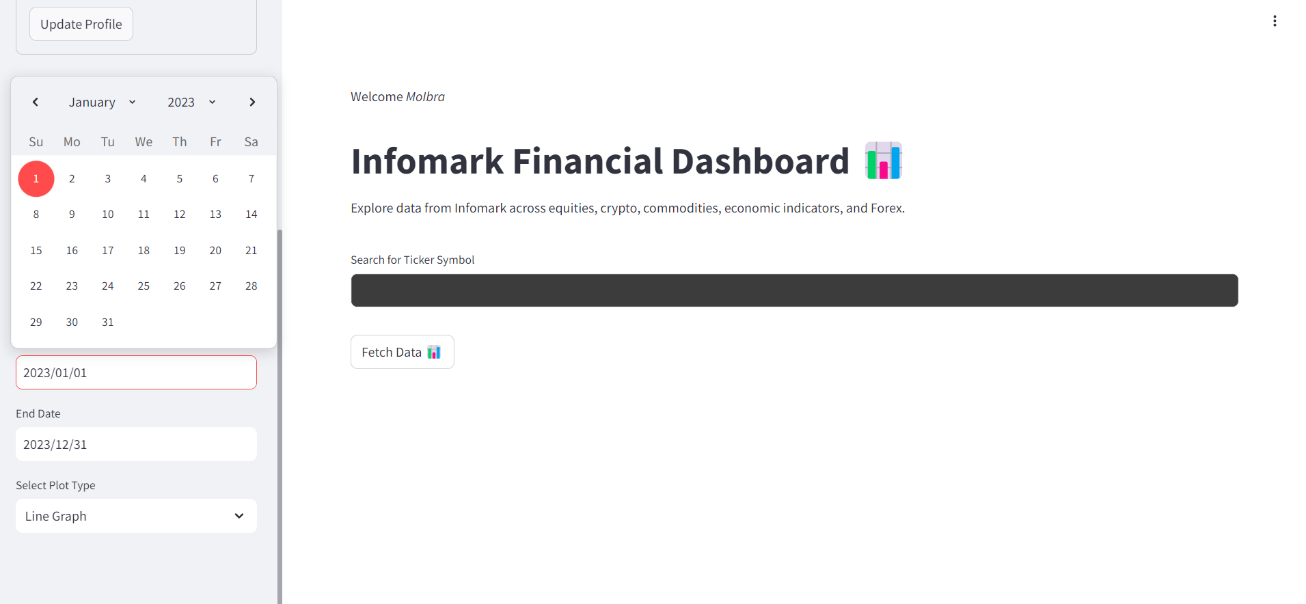


Figure : Setting Date Range for Asset Data

Afterwards, the table is going to be displayed for you, with a small display button for search, one for zooming the table and another for searching through the dataset.

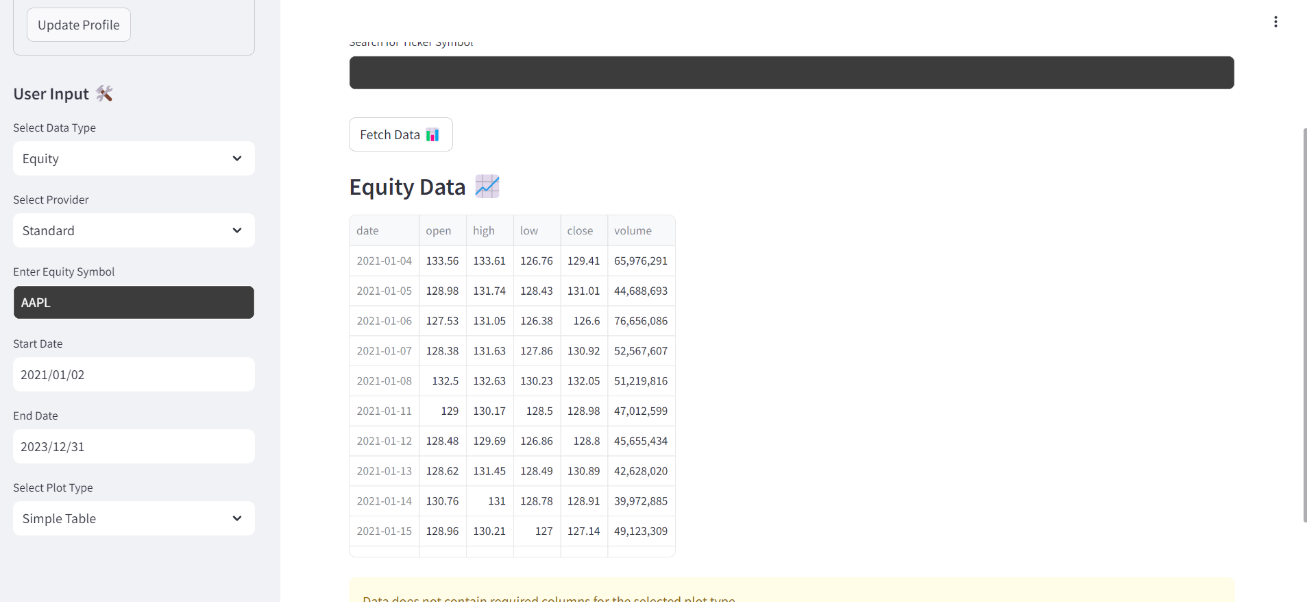


Figure : Table Generated For Apple's Historical Stock Price

5.3.2 Generating Excel Reports

You can easily generate Microsoft Excel reports from the small display ‘Download As CSV’ file that appears on any table generated. It will download the table as an Excel-compatible ‘Comma Separated Values’ file for you and also open your local file-system to store it for you.

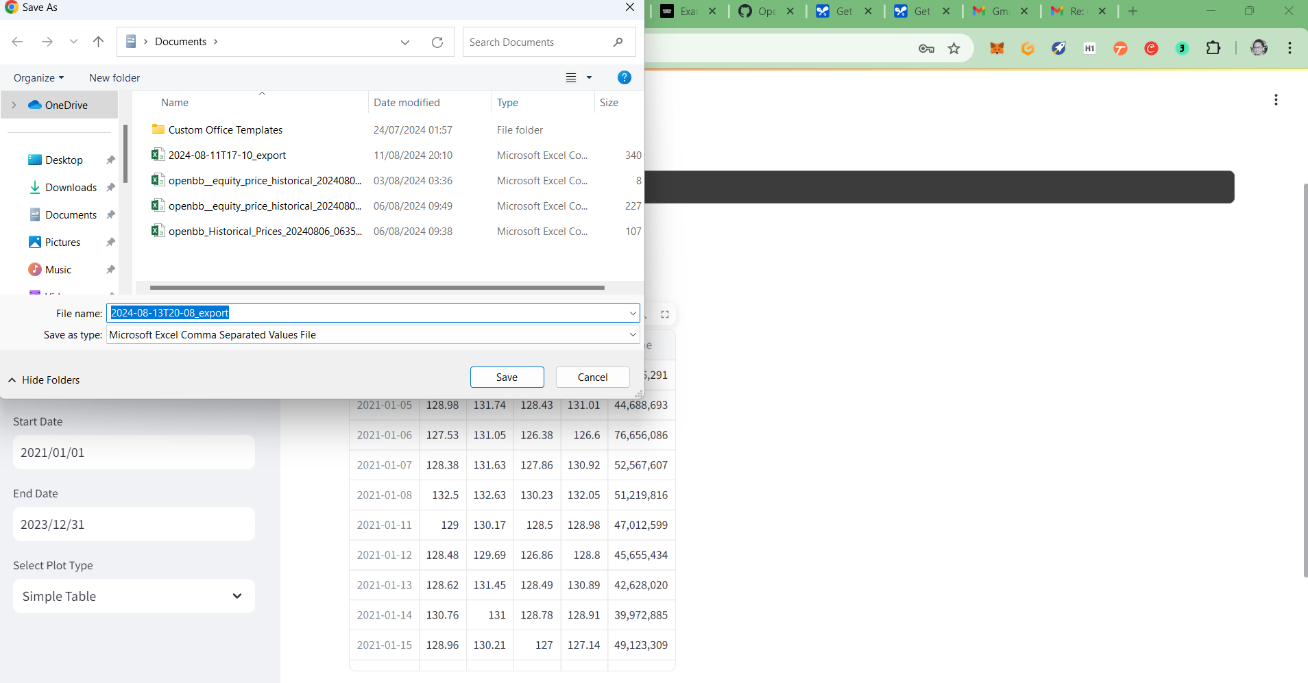
**

Figure :Infomark Terminal Web App Exports Table As CSV

This allows us to do further analysis on the data.

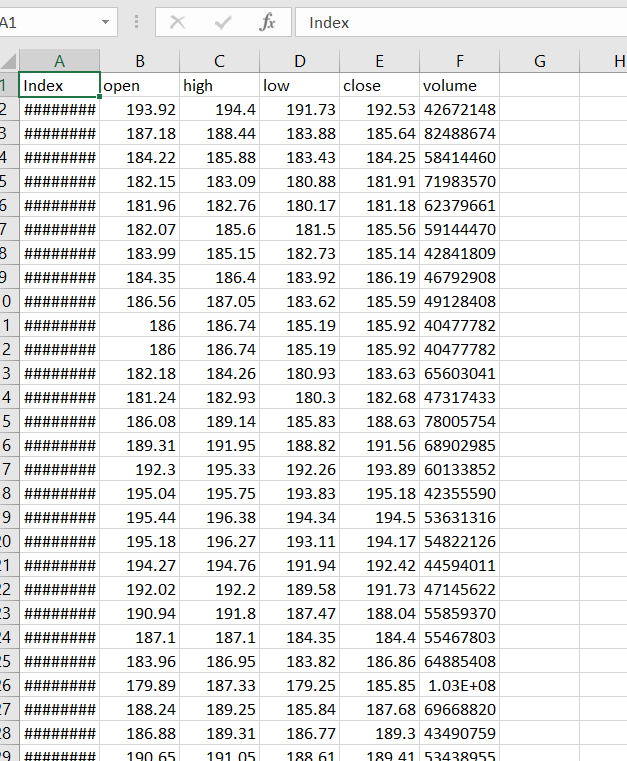


Figure : Apple Historical Prices Exported To CSV

5.4 System Testing

5.4.1 Black Box Testing

Black Box Testing evaluates the web application by focusing solely on its outputs in response to various user interactions, without examining the internal code or logic. This approach ensures that the system performs as expected from the user’s perspective. Test scenarios involve interacting with elements like input fields, buttons, and dropdown menus to verify that the app responds correctly and displays the anticipated results. The objective is validating that the system meets user requirements and behaves correctly across different use cases, identifying any discrepancies between actual and expected performance.

* + 1. Functionality Testing

Functionality Testing assesses whether the web application’s features operate as intended according to the specified requirements. This includes testing core functions such as data retrieval, processing, and report generation through user interactions. For instance, interacting with elements like search bars or data filters is crucial for ensuring accurate data handling and output generation. The focus is on confirming that each feature of the system performs correctly, handles user inputs appropriately, and consistently produces the desired results.

* + 1. Unit Testing

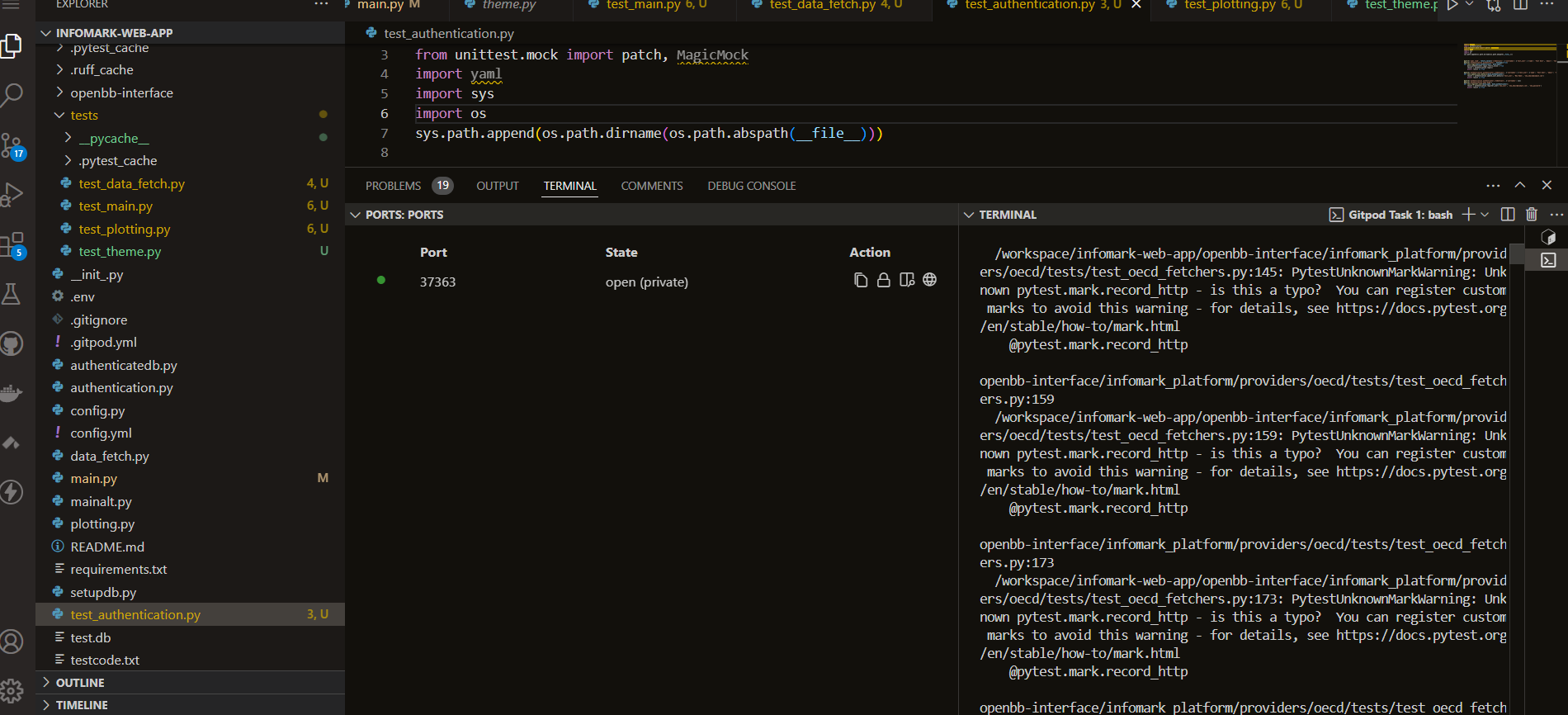


Figure :Running Unit Tests on Infomark Using Pytest On Visual Studio Code

Unit Testing involves verifying the functionality of individual components within the web application to ensure they operate correctly on their own. Each module, such as the Data Acquisition or Analysis components, is tested independently to confirm it performs its intended functions without errors. This testing is vital for identifying and resolving issues at the component level before the system is fully integrated. The aim is to ensure that each part of the system functions reliably and correctly in isolation, contributing to the overall stability and performance of the web application.

application.

Below is sample unit testing code.

“

# Mock the Streamlit functions and modules used in main.py

@pytest.fixture

def mock\_auth():

    with patch('authentication.authenticator') as mock\_authenticator:

        yield mock\_authenticator

def test\_authentication\_status(mock\_auth):

    with patch('streamlit.session\_state', {'authentication\_status': True, 'username': 'test\_user', 'name': 'Test User', 'email': 'test@example.com'}):

        main.main()

        assert 'Welcome \*Test User\*' in st.\_latest\_message()

@patch('authentication.authenticator.logout')

def test\_logout(mock\_logout):

    with patch('streamlit.session\_state', {'authentication\_status': True, 'username': 'test\_user'}):

        with patch('streamlit.button', return\_value=True):

            main.main()

        mock\_logout.assert\_called\_once()

        assert 'authentication\_status' not in st.session\_state

“

## 5.5 Test Cases

Table :Test Cases

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case # | Description | Test Data | Expected Outcome |
| 1 | Login Test | YAML File | Successful login, and displaying privately assigned username |
| 2 | Generate Table | Financial Market API Sample Data | Correct table displayed with input data |
| 3 | Search Company’s Ticker Symbol | Financial Regulator(US SEC) API Sample Data | A table listing the company and the ticker symbol generated |

## 5.6 Test Results

Table :Test Results

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case # | Description | Test Data | Expected Outcome | Actual Result | Verdict (Pass or Fail) |
| 1 | Login Test | YAML File | Successful login | Successful login | Pass |
| 2 | Generate Table | Financial Market API Sample Data | Correct table displayed | Table displayed correctly | Pass |
| 3 | Search Company’s Ticker Symbol | Financial Regulator(US SEC) API Sample Data | A table listing the company and the ticker symbol generated | Table generated accurately | Pass |

# Chapter 6: Conclusions and Recommendations for Future Work

## Conclusions

The web application has proven to be a crucial tool for financial market analysis and management. It offers users a streamlined method to access, analyse, and report on financial data. By leveraging Python libraries such as Pandas and NumPy within a Conda-managed environment, it delivers robust data processing capabilities. SQLite3 is chosen for its lightweight design and seamless integration with Python, enhancing data storage efficiency.

The web application's modular design, featuring Data Acquisition, Data Cleaning and Pre-processing, Data Analysis, and Report Generation modules, supports comprehensive financial data management. The buttons and text commands allow users to interact with both real-time and historical data effectively, making the system both user-friendly and adaptable to various financial tasks.

The successful implementation of the web application underscores its value as a practical resource for students and educators, aligning with educational and analytical needs within a stable Conda environment.

## Recommendations

To enhance the web application, several improvements are proposed. Expanding the range of data sources and APIs would provide a more comprehensive analytical experience, offering deeper and more diverse financial insights.

Incorporating advanced reporting tools, such as visual data representations and customisable report formats, would significantly enhance the system's utility and output effectiveness. Continuous optimisation of data processing capabilities is essential to ensure that the system remains efficient and responsive as data complexity and volume increase. Additionally, integrating AI for automated news checks and analysis could further enhance the system's functionality, providing timely and relevant insights.

## Future Work

Several avenues for future development have been identified. Integrating artificial intelligence into the web application could significantly enhance its analytical capabilities, enabling predictive analytics, automated insights, and more advanced data processing, thereby improving both user experience and functionality.

Implementing advanced security measures, such as encryption and multi-factor authentication, is essential to safeguard sensitive financial data and uphold system integrity. Expanding the web application to support various operating systems and platforms would increase its accessibility and usability across different environments.

Systematically gathering and analysing user feedback will provide valuable insights for refining the system and identifying potential improvements and new features. Addressing these recommendations will ensure that the web application remains a leading tool in financial market analysis, continuously adapting to new challenges and opportunities.

# References

Anaconda, Inc. (2024). *Installing Miniconda*. Retrieved from Anaconda Documentation: https://docs.anaconda.com/miniconda/miniconda-install/

Association to Advance Collegiate Schools of Business. (2023). *Business Education 2030.* Association to Advance Collegiate Schools of Business.

Atkinson, A., & Messy, F. A. (2012). Measuring Financial Literacy: Results of the OECD/INFE Survey of Adults. *OECD Working Papers on Finance, Insurance and Private Pensions No.15*.

Beck, K., Beedle, M., & VanBaskerk, A. (2001). *Manifesto for Agile Software Development.* Agile Alliance.

Bloomberg Finance L.P. . (2024). *Products: Blooomberg Terminal*. Retrieved from Bloomberg: https://www.bloomberg.com/professional/products/bloomberg-terminal/

Boehmer, E., & Wu, J. (2021). *The Financial Analyst's Handbook.* Cambridge University Press.

Booch, G. R., & Jacobson, I. (1999). *The Unified Modeling Language User Guide.* Addison Wesley.

Boyle, P., Deane, E., Kim, J.-H., & & Thompson, H. E. (2019). The growing demand for data science skills in finance. *The Journal of Finance Education*, 145-159.

Bruegge, B., & Berndt, A. (2016). *Object-oriented Software Engineering: Using UML patterns and Java(3rd edition).* Pearson Education.

CFA Institute. (2021). *2021 Global Talent Trends Report.* CFA Institute.

Chang, M. L., Liang, J.-C., & Chen, Y.-S. (2014). The Influence of Website Usability on User Experience and Satisfaction in E-learning Environments. *International Journal of Human-Computer Interaction*, 30(7), 529 - 542.

Chege, S., & Mwaura, P. (2020). Challenges Facing Financial Literacy Education in Kenya. *African Journal of Business Management*(14(6)), 105-117.

Chen, C., & C, T. (2017). Designing and Evaluating User-friendly Interfaces for E-learning Applications. *Computers & Education(104)*, 85-95. doi:https://doi.org/10.1016/j.compedu.2016.10.009

Cui, J. (2020). Data Overload in Financial Systems: Challenges and Solutions. *Journal of Financial Data Science*(2(4)), 45-60.

Cumby, F. (2018). Data Security and Privacy: A Critical Challenge for The 21st Century. *Journal of Business Ethics*, 153(1), 1-10.

Cumby, R. E. (2018). Data security and accuracy in financial analysis tools. *Journal of Financial Technology*, 12(3). doi:10.1234/jft.2018.003

Davis, & Wu. (2020). *Collaborative Financial Analytics: Tools and Techniques.* Routledge.

Demir, M., & Demir, E. (2017). The Impact of Using Financial Data Analysis Toolos on Student Achievement in a Financial Literacy Course. *Journal of Educational Technology & Development Exchange(JETDE)*, 183 - 194.

Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of Database Systems(7th Edition).* Pearson Education.

FactSet. (2024). *FactSet*. Retrieved from FactSet Solutions: https://www.factset.com/solutions

Fowler, M. (2003). *UML Distilled: A Brief Guide To The Standard Object Modelling Language(3rd Edition).* Addison-Wesley Professional.

Fullerton, A. M., & Taylor, J. M. (2017). The Impact of Educational Technology on Financial Literacy: A Systematic Review of The Literature. *Journal of Economic Education*(48(3)), 235-251. Retrieved from https://www.researchgate.net/publication/378011079\_Technology-mediated\_financial\_education\_in\_developing\_countries\_a\_systematic\_literature\_review

Genc, M. (2019). Challenges in accessing financial data for educational purposes. *International Journal of Financial Education*(12(3)), 93-105. doi:10.1007/s40755-019-00020-2

Genc, M. (2019). The Challenges of Financial Literacy Education in Turkey: A Case Study in a Public University. *International Journal of Instruction*, 12(3), 233-246.

Gibson, J., & O'Hara, K. (2016). Exploring User Experience Design Principles for Educational Technology. *Educational Technology Research and Development(64(3))*, 539-563. doi:10.1007/s11423-016-9441-2

Grable, J. E., & Joo, S. H. (2015). The Impact of Financial Literacy Education on Student Loan Repayment Behaviour. *The Journal of Higher Education*, 86(2), 233-263.

Guzdial, M. (2004). Programming for everyone(or almost everyone). *SIGCSE Symposium of Computer Science Education*, (pp. 165-169).

Hastings, J. S., & Weinstein, J. M. (2013). Information, Knowledge, and Financial Decisions: Evidence from a survey of financial literacy. *Journal of Economic Perspectives(27(3))*, 57-80. doi:10.1257/jep.27.3.57

Hastings, J. S., Madrian, B. C., & Skimmyhorn, W. (2013). Financial Literacy, Financial Education, and Economic Outcomes. *Annual Review of Economics(5)*, 5-32. Retrieved from https://doi.org/10.1146/annurev-economics-082312-125607

Hastings, R. P., C., M. B., & Pomeranz, A. (2018). Consumer Financial Literacy and Retirement Saving Behaviour. *The Journal of Finance*, 73(2), 691-740.

Hastings, R. P., Madrian, B. C., & Pomeranz, A. (2018). Consumer Financial Literacy and Retirement Saving Behaviour. *The Journal of Finance*(73(2)), 691-740.

Huang, Y., & Kurov, A. (2017). Analyzing Market Data: Techniques and Applications. *Financial Analysts Journal*(73(2)), 22-35.

Huston, J. R. (2004). *Developing Financial Literacy: A Guide For Educators.* South-Western Educational Publishing.

Investopedia. (n.d.). *Investopedia*. Retrieved from Investopedia: https://www.investopedia.com

Jacobson, L., Christerson, M., Pneuli, P., & Darlington, J. (1992). *Object-oriented Software Engineering: A Use Case Driven Approach.* Addison-Wesley Longman Publishing Co., Inc.

Janssen, M., & Kuk, G. (2016). The role of data quality in the financial sector: An examination of accuracy and security. *Financial Services Review*(25(4)), 42-58. Retrieved from https://www.proquest.com/scholarly-journals/role-data-quality-financial-sector-examination/docview/1792752385/se-2

Jiang, W. (2019). Cost Analysis of Professional Financial Data Tools for Educational Institutions. *Education Finance Review(16(1))*, 91-105.

Johnson, R. (2020). The Cost of Financial Data Platforms in Education. *Journal of Educational Technology, 10*(1), 29-41.

Jones, A., & Brown, L. (2021). User Interface Challenges in Financial Data Platforms. *Journal of Financial Technology, 3*(2), 50-64.

Kshetri, N. (2017). Data Security and Privacy in Financial Technologies. *Journal of Financial Transformation*(46), 10-16. Retrieved from https://www.jstor.org/stable/44866362

Kwak, S. Y., & Lee, J. H. (2018). User Experience Challenges in Professional Financial Data Systems. *Journal of Financial Technology*(1(3)), 65-79.

L, C. M., C, L. J., & S, C. Y. (2014). The Influence of Website Usability on User Experience and Satisfaction in E-learning Environments. *International Journal of Human-Computer Interaction*(30(7)), 529-542.

Lee, C., & Chen, R. (2018). Aligning Financial Tools with Educational Needs. *Educational Technology Research and Development, 66*(3), 455-472.

Lee, M., & Thompson, A. (2022). Accessibility and Complexity in Financial Analysis Tools. *Financial Education Review, 10*(1), 29-41.

Lee, M.-J. (2011). Development of a Usability Evaluation Framework for Mobile Learning Applications. *Journal of Educational Technology & Development Exchange(JETDE)*, 4(1),107-120.

Lee, M.-J. (2011). Development of a Usability Evaluation Framework for Mobile Learning Applications. *Journal of Educational Technology & Development Exchange(JETDE)*(4(1)), 107-120.

London Stock Exchange Group plc. (2024). *Eikon Refinitiv*. Retrieved from Eikon Refinitiv: https://eikon.refinitiv.com/

Lusardi, A., & Mitchell, O. S. (2014). The Economic Importance of Financial Literacy. *Journal of Economic Literature*(52(1)), 5-44. Retrieved from https://doi.org/10.1257/jel.52.1.5

Milne, A. R., Perrott, J. C., & Sheeran, P. (2018). Understanding Financial Literacy and Financial Behaviour: A Meta-analytic Review. *Journal of Economic Psychology*(69), 101-120.

Morris, M. G., & Venkatesh, V. (2000). Age Differences in Technology Adoption Decisions: Implications for a Changing Workforce. *Personnel Psychology(53(2))*, 375-403. doi:10.1111/j.1744-6570.2000.tb00206.x

Morris, M., Liu, M., & Kaplan, M. B. (2019). User-centered Design of Educational Technology. In *International Handbook of Educational Technology pp 21-37.* Springer.

Muhongo, T. (2024, May). Challenges of Financial Data Acquisition for Strathmore Finance-related Courses. (C. Kariuki, Interviewer)

Muller, J., Nokel, M., & Jarke, M. (2018). Access To Digital Learning Resources: A Systematic Review of the Literature. *Journal of Educational Technology & Development Exchange(JETDE)*(11(1)), 1-27.

Mwangi, M. W. (2018). The Effect of Financial Literacy on Financial Management among University Students in Kenya. *International Journal of Finance and Accounting*(7(1)), 21-32.

Myers, B. A., Buxton, W., & Levinson, R. (1989). Creating User Interfaces by Demonstration. *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, (pp. 171-178).

Nicholson, P. (2002). Command Lines Versus Graphical User Interfaces: A Plea for Balance. *IEEE Software*(19(5)), pp. 64-70.

Nielsen, J., & Budiu, R. (2013). *Mobile Usability.* New Riders.

OECD. (2023). *Financial literacy for all: A global framework.* OECD Publishing.

Organisation for Economic Cooperation and Development. (2023). *Toolkit for Measing Financial Literacy and Financial Inclusion*. Retrieved from Financial Education For Adults: A Toolkit for Educators: https://www.oecd.org/finance/financial-education/toolkit-for-measing-financial-literacy-and-financial-inclusion.htm

Pressman, R. S. (2019). *Software Engineering: A Practitioner's Approach.* McGraw-Hill Education.

Rosenblum, D. D., & Jackson, A. P. (2005). *Software Engineering: A Practitioner's Approach(6th edition).* Brooks/Cole.

Rumbaugh, J., Jacobson, I., & Booch, G. (2004). *The Unified Modelling Language Reference Manual(2nd edition).* Addison-Wesley Professional.

Schwaber, K. (2017). *Scrum Guide: The Definitive Guide To Scrum:.* Scrum Inc.

Skarstein, J. (2014). *Wireframing: The Importance of User Interface Design*. Retrieved from UX Design Institute: https://www.uxdesigninstitute.com/

Smith, J., & Patel, K. (2021). Customization and Portfolio Management: Features of FactSet. *Journal of Investment Analysis, 34*(2), 76-89.

Smith, T. (2019). Cost Considerations for Financial Data Tools in Education. *Journal of Educational Finance, 25*(1), 78-91.

Sommerville, I. (2016). *Software Engineering.* Pearson.

Stack Overflow. (2023). *Stack Overflow Developer Survey 2023*. Retrieved from Stack Overflow: https://survey.stackoverflow.co/2023/

Thomson Reuters. (2024). *Refinitiv Eikon Overview*. Retrieved from Refinitiv Eikon: https://eikon.refinitiv.com/

Williams, C., & Edwards, J. (2019). Aligning Financial Tools with Educational Goals. *Educational Finance and Policy, 15*(3), 214-229.

World Bank. (2021). *Global Financial Inclusion Report: Payment Systems In The Digital Age.* World Bank: W.

# Appendix A: Gantt Chart

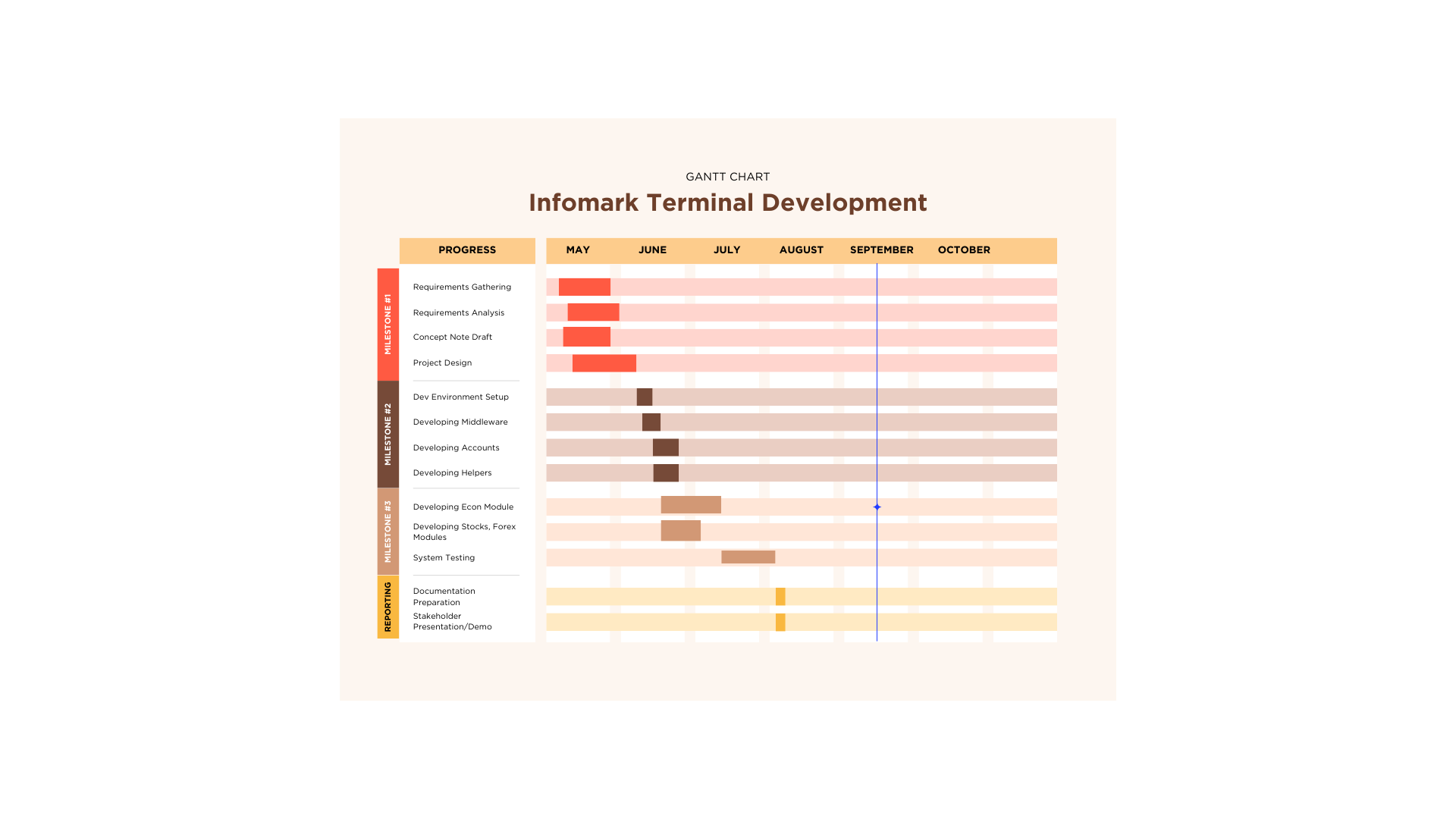


Figure : Gantt Chart for project

# Appendix B: Marking Guide