CHAPTER – 3

**SOFTWARE AND HARDWARE SPECIFICATIONS**

**3.1 Software Specifications**

**A. Overview**

The **Real Time Streaming Data Web Deployment** project aims to provide users with a robust web platform for monitoring live stock prices and financial data. Utilizing TradingView widgets, the project will deliver real-time updates and interactive visualizations to enhance user engagement and decision-making in the financial market. By leveraging cutting-edge web technologies, the application strives to create an intuitive interface that allows users to easily track market trends and analyze financial assets.

The project targets a diverse audience, from casual investors to seasoned traders, who require immediate access to market information. The primary goal is to democratize financial data access and empower users to make informed decisions based on real-time analytics.

* User-friendly interface designed for accessibility.
* Interactive widgets for a better understanding of market dynamics.
* Comprehensive support for various financial instruments.

**B. Functional Requirements**

To achieve its objectives, the application must meet several functional requirements that facilitate user interaction and data visualization. Key functionalities include:

**Real-time Stock Data**: Users should be able to view live stock prices and historical data without the need for page refreshes. The platform will employ WebSocket connections for instant updates.

**Search and Filtering**: The application must allow users to search for specific stocks or financial assets and filter results based on various criteria such as market capitalization, price changes, and trading volume.

**User Interaction**: Users will have the capability to customize their dashboard, selecting which stocks to follow closely and arranging widgets according to their preferences.

**Charting Capabilities**: The application will offer interactive charts that enable users to visualize stock trends, utilizing TradingView's advanced charting tools.

The successful implementation of these functional requirements will ensure a seamless user experience and facilitate informed investment decisions.

**C. Non-functional Requirements**

In addition to functional specifications, the project emphasizes several non-functional requirements that are critical for the application's performance, security, and reliability. These include:

**Performance**: The system must load and update real-time stock data quickly, minimizing latency to ensure a fluid user experience. Load testing will be conducted to verify that the application can handle high traffic during peak market hours.

**Scalability**: The architecture should be designed to scale horizontally, accommodating an increasing number of users and financial assets as the project grows. This involves using cloud services that allow for dynamic resource allocation.

**Security**: User data and API communications must be secured using industry-standard encryption protocols to prevent unauthorized access and data breaches. Regular security audits will be conducted to ensure compliance with best practices.

**Availability**: The system should maintain high availability, particularly during market hours when user engagement peaks. Strategies such as load balancing and redundancy will be employed to achieve this goal.

**Maintainability**: The codebase should follow clean coding standards and modular design principles to facilitate future updates and maintenance efforts. Documentation will be provided to support developers in managing the system.

By adhering to these non-functional requirements, the project will ensure a reliable and secure platform for users.

8

**D. Technology Stack**

The technology stack for this project has been carefully selected to meet the functional and non-functional requirements while ensuring efficient development and deployment. Key technologies include:

**Frontend**: ReactJS will be utilized to build a dynamic user interface, ensuring responsiveness and a smooth user experience. TradingView widgets will be integrated for live data visualization, while Ant Design will provide a modern UI framework.

**Backend**: The backend will be developed using Node.js with Express to handle API requests and manage server-side logic efficiently. This technology choice allows for non-blocking I/O operations, crucial for real-time applications.

**Database**: MongoDB will be employed for storing any necessary historical data or user preferences, allowing for flexible data management and scalability.

**APIs**: The TradingView API will serve as the primary data source for real-time stock information, providing essential market data and charting capabilities.

**Deployment**: The application will be hosted on cloud platforms like AWS or Heroku, leveraging their services for scalability and reliability. A content delivery network (CDN) will be used to enhance load times and accessibility for users across different regions.

This technology stack is designed to support the project's objectives effectively while ensuring a high-quality user experience.

**E. External Interfaces**

The application will interface with various external components to gather data and enhance user interaction. These interfaces include:

**TradingView API**: This API will be the main source of live financial data, enabling real-time updates and charting features. It will provide comprehensive market information, allowing users to track multiple stocks seamlessly.

**Cloud Hosting Services**: AWS or Heroku will host the web application, ensuring that it is accessible to users at all times. These platforms will provide the necessary infrastructure to support high traffic volumes.

9

**CDN (Content Delivery Network)**: A CDN will be implemented to improve the loading speed of the application by caching static assets and distributing them to users based on their geographic location. This will enhance the overall performance and user experience.

By integrating these external interfaces, the project will ensure a robust and responsive platform that meets user demands.

**F. Performance Requirements**

To ensure optimal functionality and user satisfaction, the application must meet specific performance criteria:

**Real-Time Updates**: The system should refresh stock data every few seconds to provide users with the latest information without requiring manual refreshes.

**Concurrent Users**: The application must handle at least 100 concurrent users simultaneously, ensuring that performance remains stable during peak times.

**Response Time**: User requests for stock data must be processed and responded to within one second to maintain a smooth user experience.

**Time Zone Handling**: The application should accurately update and display stock prices according to different time zones, catering to users from various regions.

Meeting these performance requirements is essential to delivering a high-quality user experience.

**G. Constraints**

The project may face several constraints that could impact its implementation and functionality. These constraints include:

**Dependency on TradingView**: The application's reliance on the TradingView API means that any delays or outages on their end could affect the real-time data experience for users. Contingency plans should be in place to handle such situations.

10

**Time Zone Adjustments**: Accurately managing multiple time zones presents a challenge, especially during daylight saving changes. The application must be designed to handle these changes seamlessly to provide accurate data.

**Regulatory Compliance**: Depending on the regions served, the application may need to adhere to specific financial regulations and compliance standards, which could affect its operation and data handling processes.

Awareness of these constraints will be essential in guiding the project's development and ensuring a successful outcome.

11

**3.2 Hardware Specifications**

**A. Server Requirements**

**Processor**: Minimum dual-core processor with a clock speed of 2.0 GHz or higher.

**RAM**: At least 8 GB of RAM for smooth multitasking and handling multiple user requests simultaneously.

**Storage**: SSD storage with a minimum capacity of 256 GB to ensure quick data retrieval and application responsiveness.

**B. Client Requirements**

**Browser Compatibility**: Latest versions of major web browsers (Chrome, Firefox, Safari) to support modern web standards and ensure a smooth user experience.

**Internet Connection**: Stable broadband connection with a minimum speed of 5 Mbps for optimal real-time data streaming and interaction.

**C. Network Infrastructure**

**Bandwidth**: Sufficient upload and download bandwidth to accommodate multiple concurrent users accessing live data without delays.

**Router**: A reliable router capable of handling high traffic volumes, ideally supporting Wi-Fi 5 or higher for enhanced connectivity.

**D. Backup and Recovery**

**Backup Solutions**: Regular backups should be configured on a separate storage device or cloud service to prevent data loss.

**Power Supply**: Uninterruptible Power Supply (UPS) to ensure continuous operation and data protection during power outages.

12

**3.3 User Specifications**

**A. End-User Requirements**

* **Device Compatibility**: Users should have access to modern devices such as smartphones, tablets, laptops, or desktops that support up-to-date web browsers.
* **Browser**: Users must have an updated version of web browsers like Google Chrome, Mozilla Firefox, Microsoft Edge, or Safari to ensure compatibility with the web application.
* **Operating System**: The platform is compatible with popular operating systems, including Windows, macOS, Linux, iOS, and Android.

**B. Internet Connectivity**

* **Connection Speed**: A stable internet connection with at least 3 Mbps download speed is recommended for seamless real-time streaming of stock market data.
* **Network Stability**: Consistent network connectivity is crucial to avoid interruptions or delays in receiving up-to-the-minute stock prices.

**C. User Interaction Requirements**

* **Basic Computer Literacy**: Users should be familiar with browsing the web, navigating through different sections of the web application, and interpreting financial data, charts, and widgets.
* **Account Creation and Login**: Users must be able to create accounts and securely log in to access personalized services, watchlists, and other features of the platform.

**D. Security and Privacy**

* **Secure Connection**: Users should access the web application through secure connections (HTTPS) to ensure their data and financial information are protected.
* **Authentication**: Two-factor authentication (2FA) should be enabled for users who prefer an added layer of security when logging in.

13