American College of Technology (ACT)

Department of Computer Science

Title:

Ethiopian Stock Market Simulation Platform for Learning, Regulatory Compliance and Market Preparedness

Name	ID
1. Biniyam Kefelegn	010/2022
2. Tesfaye Fekadu	074/2022
3. Sebele Fekede	

12-12-2024

1 Contents

Chapter 1: Introduction

1.1. Background of the project

The Ethiopian Capital Market Authority (ECMA) is establishing a formal securities exchange to stimulate economic growth and provide a platform for raising capital. This marks a transformative step in Ethiopia's financial landscape, offering unprecedented opportunities for investors, brokers, and regulators. However, the Ethiopian financial market is still in its infancy, and a significant lack of practical experience among stakeholders threatens the success of this initiative.

The Stock Market Simulation Platform is designed to address this gap by providing an interactive and practical learning environment. This platform mimics the functionality of a real stock exchange, allowing users to simulate trading activities, understand market behavior, and test regulatory policies without real-world financial risks. It empowers participants to gain hands-on experience in stock trading, market analysis, and regulatory compliance.

By fostering understanding and market readiness among traders, listed company admins, and the regulators, this platform plays a critical role in supporting the successful launch of Ethiopia's stock market.

1.2. Statement of the Problem

Ethiopia's new financial market faces challenges in developing skilled and knowledgeable participants. Due to a lack of practical experience, those participants may not be well-prepared to handle real-life situations related to trading, market analysis, and regulatory oversight. The absence of a structured educational tool makes it difficult for stakeholders to understand key market operations and comply with regulatory standards.

There is a need for an interactive simulation tool that provides practical experience in stock market trading, market analysis, and regulatory compliance, following the guidelines set by the ECMA.

1.3. Objectives of the project

1.3.1. General Objective:

To develop an interactive, user-friendly stock market simulation platform that prepares Ethiopian traders, companies and regulators for the upcoming Ethiopian stock exchange.

1.3.2. Specific Objectives:

1. Role-Based Registration and Simulation: Design a platform that allows users to register under specific roles (e.g., Traders, Listed Company Admins) and actively participate in simulated stock market activities.
2. Advanced Trading Engine: Develop a trading engine supporting two key order types (market and limit orders) with real-time matching and execution capabilities based on the Price-Time Priority Algorithm.
3. Regulatory Monitoring and Testing Module: Create a module enabling ECMA regulators to oversee simulated market activities, enforce compliance, and test policy impacts effectively.
4. Order Notification System: Build a notification mechanism to inform traders when their orders are matched and executed in real time.
5. Configurable Trading Hours: Provide regulators with the capability to define the platform's working hours, ensuring trading activities are restricted to specific time windows.
6. Listed Company Administration: Enable company admins to publish their stocks and set declared dividend ratios, ensuring realistic and dynamic market simulations.

1.4. Scope of the project

The platform is designed to simulate essential functions of a stock market environment, focusing on features such as trading simulation, order matching, price discovery, regulatory compliance. It offers role-specific functionality for traders, listed company administrators, and regulators, ensuring a tailored experience for each type of user. While

the platform enables simulated trading and learning in a secure environment, it does not handle actual financial transactions or operate as a real trading system.

1.5. Limitation of the project

- The platform operates in a simulated environment, and outcomes may not fully replicate real-world market conditions.
- Only two order types (market and limit orders) are supported, which may limit complex trading strategies.
- Market data and analysis tools are designed for simulation purposes and may lack real-world accuracy.
- The platform relies on active participation from all user roles (Traders, Company Admins, and Regulators) to simulate a balanced and functioning market.
- External factors, such as unforeseen technical challenges or incomplete adherence to real-world policies, may affect the accuracy of the simulation.

1.6. System Development Methodology

1.6.1. System Development Approach

The development of the Ethiopian Stock Market Simulation Platform follows an agile software development methodology to ensure iterative progress, continuous feedback, and adaptability throughout the development lifecycle. Given the absence of a stock market platform in Ethiopia, the platform draws extensively on guidelines and resources from the Ethiopian Capital Market Authority (ECMA) website, along with global best practices researched from platforms such as Investopedia and other real-world stock market simulations. The methodology is divided into the following key phases:

1.6.2. Requirement Analysis and Design

Requirement Analysis

- Refer to ECMA directives, manuals, and regulatory frameworks available on their official website to ensure platform compliance with all local regulations.
- Supplement local research by examining global best practices and documentation from platforms like Investopedia to understand the dynamics of stock market operations and adapt them to Ethiopia's context.

Designing

- Develop the platform's architecture to include core modules such as the trading engine, user management, portfolio management, and regulatory module.
- Design user-friendly interfaces tailored for different roles: traders, listed company admin user and administrators or regulators.
- Outline the Profit and Dividend Calculation module.

Development

Develop core modules, including:

- **User Management**:
  - ✓ Build a comprehensive user management system supporting multiple roles such as traders, listed company administrators, and regulators. Include KYC verification and role-based permissions
- **Trading Engine**:
  - ✓ Develop a high-performance trading engine capable of efficiently processing stock buy/sell orders and matching transactions with precision.
  - ✓ Ensure compliance with ECMA's regulatory framework, integrating rules for trading restrictions, transaction limits, and suspension management.

Testing and Quality Assurance

Testing:

Conduct comprehensive testing at all levels:

- **Unit Testing**: Test individual components (e.g., trading engine, portfolio updates) for correctness.
- **Integration Testing**: Validate the smooth interaction between modules such as trading, portfolio management.

Deployment and Training

- The platform is still in the development phase, and deployment has not yet been completed.

1.6.3. System Development Tools

To efficiently develop the Ethiopian Stock Market Simulation Platform, the following tools and technologies are being utilized:

- **Development Frameworks:**
  - ✓ **Backend:** Django, for handling server-side logic, APIs
  - ✓ **Frontend:** Angular, for building a responsive, interactive, and user-friendly interface.
- **Programming Languages:**
  - ✓ **Python:** For robust and scalable backend development.
  - ✓ **Typescript/JavaScript:** For efficient frontend development with Angular.

- **Database Management:**
  - ✓ **PostgreSQL:** For managing structured and reliable data storage.

- **Version Control and Collaboration:**
  - ✓ **Git:** For source code versioning and management.
  - ✓ **GitHub:** For team collaboration, issue tracking, and code review.
- **Additional Tools:**
  - ✓ **Postman:** For API testing and debugging.
  - ✓ **VS Code:** As the primary IDE for development.

.

1.7. Significance of the Project

This project is crucial for Ethiopia's capital market development as it provides a practical learning tool for future market participants. By simulating real market conditions, the platform will help users gain confidence, understand trading mechanisms, develop regulatory compliance skills, and prepare for the real Ethiopian stock exchange. It will also serve as a useful tool for ECMA to test regulatory policies in a controlled environment.

1.8. Beneficiaries of the Project

The Ethiopian Stock Market Simulation Platform is designed to address the knowledge and skill gaps among participants in Ethiopia's emerging financial market. The beneficiaries of the project include the following

- **Traders (Public Individuals)**:
  - ✓ Experience a practical, risk-free environment to learn stock trading and portfolio management.
  - ✓ Gain confidence and hands-on experience to participate in the Ethiopian stock market.
  - ✓ Understand how regulatory policies and market dynamics influence trading strategies and investment decisions.

- **Listed Company Representatives (Company Admins)**:

  - ✓ Manage company stocks, including publishing shares and setting declared dividend ratios.
  - ✓ Simulate interactions with the market to prepare for real-world listing scenarios.
  - ✓ Gain insights into investor behavior and the impact of trading activities on company stocks.

- **Regulators (ECMA Representatives)**:

  - ✓ Monitor simulated market activities to ensure compliance with ECMA regulations.
  - ✓ Test and refine regulatory policies in a controlled environment to mitigate implementation risks.
  - ✓ Simulate market scenarios to assess policy impacts and improve governance mechanisms.

1.9. Feasibility Study

The feasibility study assesses the Ethiopian Stock Market Simulation Platform's technical, economic, and operational viability. It ensures the platform meets stakeholder needs and achieves its objectives effectively.

1.9.1. Technical Feasibility

This section assesses the technical viability of developing and implementing the platform.

- **Platform Design and Development**

  - ❖ **Backend**: Django is leveraged for its scalability, reliability, and ability to handle complex server-side logic and APIs.
  - ❖ **Frontend**: Angular is chosen for creating a highly responsive and interactive user experience.

- **Database**: PostgreSQL provides robust, structured data management with support for complex queries and scalability.

- **Technical Expertise**
  - The team includes skilled developers with experience in Python, Django, Angular, PostgreSQL, and RESTful API development.
  - Tools like Git and GitHub enable efficient source code management and seamless collaboration among team members.

- **Scalability and Future Upgrades**
  - The platform's modular architecture ensures adaptability for future enhancements, including advanced analytics, sentiment analysis tools, and risk management modules.

- **Expected Benefits**
  - **Empowering Traders**:
    - ✓ Provides a risk-free environment to learn, practice, and build confidence in stock trading.
  - **Supporting Regulators**:
    - ✓ Refines policies and tests market mechanisms to ensure compliance and integrity.
  - **Enhancing Company Readiness**:
    - ✓ Equips companies to manage stocks, simulate listings, and understand market dynamics.
  - **Economic Growth**:
    - ✓ Reduces market errors and boosts liquidity through training and awareness.
  - **Technological Advancement**:
    - ✓ Promotes innovation and lays a foundation for future market tools.

1.9.2. Operational Feasibility

This aspect evaluates whether the project aligns with the needs of its stakeholders and can be effectively implemented.

- **Regulators (ECMA Representatives):**
  - ✓ The platform serves as a comprehensive testing ground for regulatory policies, enabling ECMA to monitor compliance, assess policy impacts, and refine governance mechanisms.
- **Traders (General Public):**
  - ✓ Empowers individuals to enhance their trading knowledge and confidence by simulating real-world stock market activities in a risk-free environment.
- **Listed Company Representatives (Company Admins):**
  - ✓ Allows company admins to simulate the listing process, manage stocks, and setting up the declared dividend ratio on the system.

Ease of Use:

- ✓ The platform is designed with a highly intuitive and user-friendly interface, ensuring all participants can easily access and navigate its features.
- ✓ Training programs and resources are provided to enable stakeholders to maximize the platform's capabilities and achieve their objectives.

1.10. Project schedule

The Ethiopian Stock Market Simulation Platform will be completed in approximately 4 months, divided into four phases with specific milestones and activities.

Phase	Days	Key Activities	Critical Modules/Tasks
Planning Phase	Days 1-5	- Create product backlog - Define project scope and success criteria	N/A
Analysis Phase	Days 6-15	- Analyze requirements and prioritize user stories - Refine acceptance criteria - Estimate effort for tasks	Define stock market simulation requirements and dependencies
Design Phase	Days 16-30	- Design application architecture and workflows - Develop database schema - Define branding and UX/UI - Create technical documentation	Database Design for Transactions and User, User Portfolios, Order , Trade, Notification, Dividend and Listed Company Table
Development Phase	Days 31-90	- Backend API development - Stock market simulation engine - Web UI development - Integration of third-party services	**Stock Market Simulation Engine (Days 31-60) Backend APIs (Days 60-75) and UI Integration (Days 75-90)**
Testing Phase	Days 91-100	- Test backend, APIs, and web platform - Conduct user acceptance testing - Analyze and resolve bugs	**Testing Stock Simulation and Transaction Modules**
Implementation Phase	Days 101-108	- Deploy backend and web platform - Configure domain - Provide training and support - Collect final feedback	**Deployment and Domain Setup**
Maintenance Phase	Ongoing	- Fix bugs - Add new features - Optimize performance - Conduct backups and monitor user feedback	Continuous optimization and future feature updates

This schedule ensures timely delivery while allowing for iterative development and stakeholder engagement.

1.11. Project Budget

Comprehensive Budget Table

Category	Item	Frequency	Cost (ETB)	Description
One-Time Costs	Printing and Laminating	One-time	1,000	For initial presentations and materials.
	.gov.et domain Registration for year from ethiotelecom	One-time	550	Purchase and configure a custom domain.
	Initial Marketing	One-time	3,000	Digital ads and promotional campaigns setup.
	Content Creation	One-time	8,000	Hire content creator for web content and documentation.
Recurring Costs (4 Mo)	Broadband Internet (6 MB)	4 months reserve	4,375	Internet subscription for platform operation.
	VPS Hosting from Hostwinds	One Year reserve	($49.99) 6200	Web hosting for backend operations (8 GB RAM, 4 CPUs).
	Miscellaneous Maintenance	4 months reserve	6000	Reserve for unexpected maintenance during initial period.
Contingency Reserve	Platform Maintenance	One-time	10,000	Reserve fund for unforeseen issues and updates.

Chapter 2: Requirement Analysis

2.1. Current System Description

Ethiopia currently lacks both a formal stock market and a simulation platform to facilitate understanding and practice of stock market operations. As such, no existing system in Ethiopia performs the functionalities that this project aims to address.

2.1.1 Major function of the current system

The absence of an operational stock market system means that:

- No mechanisms exist for simulating trading activities or analyzing market dynamics.
- Regulatory bodies lack tools to test compliance frameworks and policies.
- Educational resources and platforms for stakeholders to understand stock markets are non-existent.

2.1.2 Problem of Existing System

The lack of a stock market and simulation platform results in several challenges:

- **Knowledge Gap**: Stakeholders, including potential traders, companies and regulators, lack practical exposure to stock market operations, trading strategies, and compliance requirements.
- **No Practical Training Tools**: There is no simulated environment where users can practice trading, market analysis, or regulatory testing in a risk-free setting.
- **Regulatory Challenges**: The Ethiopian Capital Markets Authority (ECMA) has no platform to test regulatory policies or simulate the impact of those policies.
- **Limited Public Awareness**: The public has minimal access to tools or educational resources to understand stock markets, limiting their readiness for a functional exchange in the future

2.2. Requirement Gathering

2.2.1 Requirement Gathering Methods

To build a simulation platform that addresses the challenges, the following methods were employed:

- **Observation**: Studying simulation platforms in established markets to identify industry best practices, design considerations, and feature sets.
- **Document Review**: Reviewing ECMA directives, legal documents, and regulatory frameworks to ensure alignment with Ethiopia's evolving capital market regulations.

2.2.2 Business Rules

Regulatory Compliance:

The platform must strictly adhere to ECMA's regulatory frameworks, ensuring compliance in all simulated activities.

Role-Based Access:

Only registered users, including **Traders**, **Listed Company Representatives**, and **Regulators**, are allowed access to functionalities relevant to their roles.

Trading Engine Simulation:

The trading engine must accurately replicate real-world stock trading mechanisms, supporting multiple order types (e.g., market and limit orders) and employing a Price-Time Priority Algorithm for execution.

Work Hour Management:

Regulators have the authority to define the platform's active trading hours, ensuring trading activities occur only during designated periods.

Trader Suspension:

Regulators can suspend traders for policy violations, ensuring a fair and compliant trading environment.

Listed Company Operations:

Listed company representatives can manage their stocks, including publishing shares and setting declared dividend ratios, in compliance with ECMA regulations.

2.3. Proposed System Description

2.3.1. Overview

The proposed system is an Ethiopian Stock Market Simulation Platform, designed to mimic real-world trading activities, provide market analysis tools, and incorporate regulatory oversight features. This platform will serve as a comprehensive learning tool for stakeholders, including

- **Traders**:
  - Designed for individual participants to learn effective trading strategies, enhance their portfolio management skills, and understand the intricacies of market operations.

- Traders can place buy/sell orders, track their portfolio, and receive notifications about their trades, simulating a complete trading experience.

❖ **Listed Company Administrators**:

- Enables representatives of listed companies to manage their company profiles, publish stocks, and declare dividends.
- They can also monitor their company's trading activity, ensuring a realistic simulation of administrative responsibilities in a stock market ecosystem.

❖ **Regulators**:

- Provides tools for simulating regulatory oversight, approving or rejecting trader and company registrations, monitoring trading activities, and managing compliance violations.
- Regulators can generate comprehensive reports, set market alerts, and test the effectiveness of regulatory policies in a controlled environment.

By providing a secure, virtual environment, the Ethiopian Stock Market Simulation Platform empowers stakeholders to gain practical experience, deepen their understanding of stock market dynamics, and contribute to the successful establishment and sustainability of Ethiopia's upcoming stock exchange.

2.3.2. Functional Requirements

❖ **User Registration and Role Management**

- Supports multiple user roles: **Traders**, **Listed Company Admins**, and **Regulators**.
- Role-based access control ensures secure, tailored access to features.

❖ **Trading Engine**

- **Order Placement**: Supports multiple order types (market, limit).
- **Order Matching**: Utilizes **Price-Time Priority Algorithm**, ensuring fair and efficient execution by prioritizing best price and order submission time.
- **Execution and Market Depth**: Simulates real-world trading with real-time order books, bid-ask spreads, and partial matching capabilities.

- **Portfolio Management**
  - Allows traders to manage virtual portfolios, monitor performance, and track trading history.
  - Provides performance analytics and risk assessment tools.
- **Regulatory Tools**
  - Enables regulators to monitor compliance, enforce rules, and generate detailed reports.
- **Market Analytics**
  - Offers real-time updates, historical market data, and visualization tools for trend analysis and decision-making.

2.3.3. Nonfunctional Requirements

2.3.3.1. Performance

The platform must provide a responsive and efficient user experience, handling trading simulations, regulatory operations, and administrative tasks with minimal latency and seamless interactions.

2.3.3.2. Scalability

The system should support the addition of new features and the ability to scale to accommodate an increasing number of users and activities as the platform grows alongside Ethiopia's market readiness.

2.3.3.3. Availability

The platform must remain highly accessible and reliable, ensuring continuous operation during the designated transaction periods set by the system's regulators. Outside these periods, the platform should remain available for non-transactional activities such as portfolio management, monitoring, and administrative tasks, with minimal downtime for maintenance.

2.3.3.4. Reliability

The system must ensure the accuracy and consistency of all simulations, including trading, order matching, and compliance monitoring, to provide a realistic and dependable market environment.

2.3.3.5. Maintainability

The platform's codebase must be modular and well-documented to allow for straightforward updates, troubleshooting, and the integration of new functionalities as needed.

2.3.3.6. Security

Robust security measures must be implemented, including secure authentication, role-based access control, and data encryption, to safeguard user data and trading activities.

2.3.3.7. Usability

The user interface must be designed for ease of use, providing clear navigation and accessible features tailored to the needs of traders, listed company administrators, and regulators.

Chapter 3: System Model

3.1. Scenarios

3.1.1. Use Case Model

The Use Case Model provides a structured representation of the interactions between users (actors) and the system as well as the role of the trading engine as a sub-system. Highlighting the key functionalities offered by the Ethiopian Stock Market Simulation Platform. It serves as the foundation for understanding system requirements and user roles.

3.1.1.1.Actor Identification

The platform supports the following primary actors:

1. **Trader**:

Role: The trader represents an individual participant in the simulation who interacts with the system to perform trading activities.

Responsibilities:

- Registers with the platform to gain access.
- Places buy or sell orders using the trading interface.
- Monitors order statuses, manages portfolios.
- Receives notifications about executed trades, and system updates.

2. Listed Company Admin:

Role: Acts as a representative of a listed company managing stock-related activities.

Responsibilities:

- Registers the company and manages its profile.
- Publishes company stocks and declares dividends for shareholders.
- Monitors trading activities involving their listed stocks.
- Generates reports and manages stock visibility on the platform.

3. **Regulator**:

Role: Represents the governing body overseeing the simulation to ensure compliance and regulatory adherence.

Responsibilities:

- Approves or rejects user registrations for traders and listed company admins.
- Monitors market activities for compliance and can suspend traders if necessary.
- Sets system working hours to regulate trading periods.
- Generates compliance and market reports to ensure transparency and fairness.

4. Trading Engine (Sub-System):

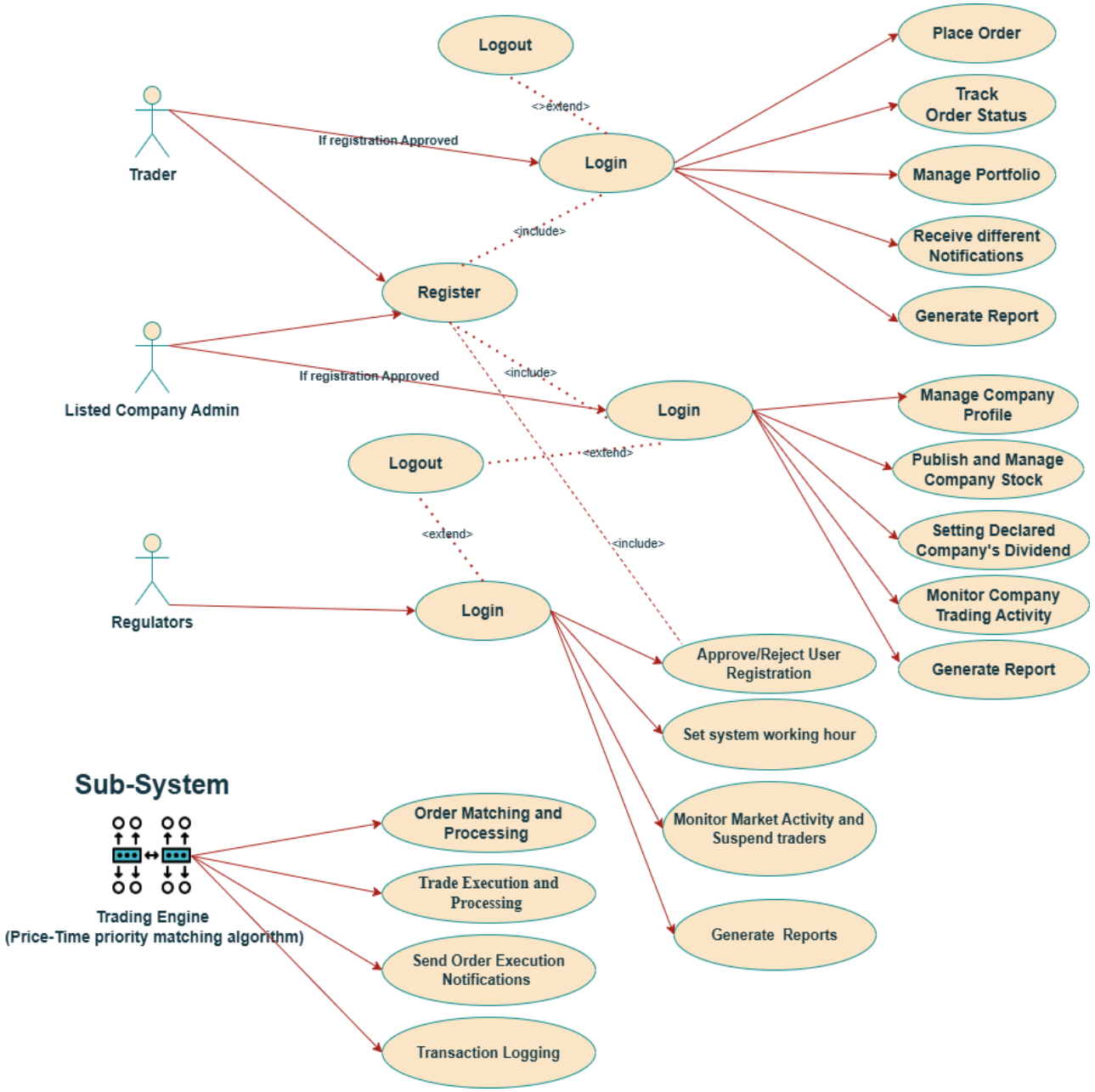
Role: The trading engine operates as a core sub-system, automating critical trading functionalities.

Responsibilities:

- Matches buy and sell orders using a price-time priority algorithm.
- Executes trades in real-time, updating user portfolios and order statuses.
- Sends notifications to users upon successful trade execution.
- Logs all transactions for regulatory and auditing purposes.

3.1.2. Use Case Diagram

The use case diagram visually represents the interactions between actors (Trader, Listed Company Admin, Regulator) and their respective functionalities. It also highlights the role of the trading engine as a sub-system for automating trading-related processes. Refer to the provided diagram for detailed visualization.



Use Case: Trader Operations

Actors	**Trader**
Description	This use case allows traders to register, place orders, manage portfolios, track orders, and receiving system notifications.
Preconditions	The trader must have a KYC approved and active account.
Post conditions	User can Login, place orders, manage portfolios, receive system notifications and can receive dividend for their owned stocks
Events	➢ **Register User**: Traders register by providing required details such as username, email, and password and provided required document (KYC documents). ➢ **Login**: Traders log in to access their trading dashboard and to perform trading activity. ➢ **Place Order:** Traders place buy/sell orders in the system using different order types. ➢ **Track Order Status**: Traders monitor the status of their orders in real time. ➢ **Manage Portfolio**: Traders can view, analyze, and manage their stock holdings. ➢ **Receive Notifications**: Traders receive alerts on order execution, portfolio updates.
Alternative Events	➢ **Invalid Login Details**: Displays an error if incorrect credentials are provided, prompting the trader to re-enter their details. ➢ **KYC is not verified**: Displays and error if user is not verified by the regulators. ➢ **Order Cancellation**: Alerts the trader if an order cannot be matched or executed due to lack of order matching at the end of the day.
Exceptions	**System Downtime**: The system notifies the trader of temporary unavailability during order placement, ensuring the issue is resolved promptly.

Use Case: Listed Company Administrator Operations

Actors	Listed Company Admin
Description	This use case supports listed companies in publishing stocks, declaring dividends, monitoring activities, and generating reports.
Preconditions	Listed company administrators must have a verified and approved account.
Post conditions	Stocks are listed, dividends declared, or reports generated successfully.
Events	➢ **Register User**: Admins register by providing company details for system inclusion. ➢ **Login**: Admins log in to manage their company profile and stock-related activities. ➢ **Manage Company Profile**: Admins update company details such as stock offerings and contact information. ➢ **Publish Stock**: Admins list and update their company's stock for trading. ➢ **Declare Dividends**: Admins announce dividends for shareholders. ➢ **Monitor Trading Activity**: Admins monitor activities associated with their listed stocks. ➢ **Generate Report**: Admins generate stock performance and trading activity reports.
Alternative Events	➢ **Invalid Login Details**: Displays an error if incorrect credentials are provided, prompting the trader to re-enter their details. ➢ **KYC is not verified**: Displays and error if user is not verified by the regulators. ➢ **Invalid Stock Data**: Displays an error if incomplete or invalid stock data is entered during publishing.
Exceptions	System Error: Temporarily halts stock publishing or dividend declarations during a system malfunction.

Use Case: Regulator Operations

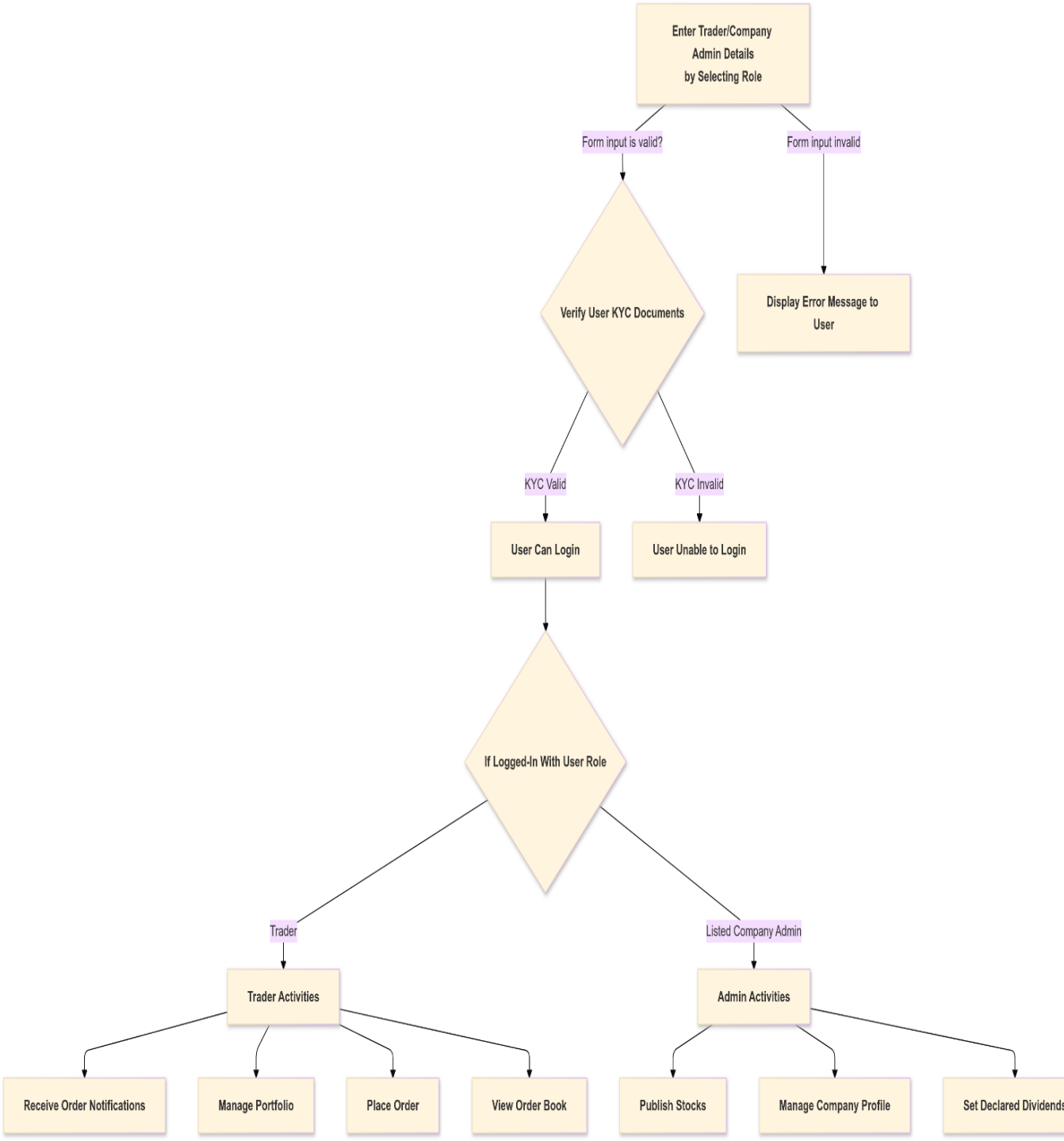
Actors	Regulator
Description	This use case allows regulators to monitor market activities, ensure compliance, and approve/reject user registrations.
Preconditions	Regulators must have authorized accounts with sufficient permissions.
Post conditions	Regulator can Set System working hour for each days, suspend traders from buying/selling specific stock or global in the platform and oversee the overall trading activity.
Events	➢ **Login**: Regulators log in to monitor and manage compliance activities. ➢ **Approve/Reject Users**: Regulators validate and approve or reject trader and listed company registrations. ➢ **Set System Working time**: Regulators will set system working time. ➢ **Suspend traders:** Regulators can suspend traders from trading specific stock or from the platform. ➢ **Generate Reports**: Regulators create detailed compliance and activity reports.
Alternative Events	➢ **Invalid Login Details**: Displays an error if incorrect credentials are provided, prompting the trader to re-enter their details.
Exceptions	Data Retrieval Issue: Alerts regulators if there is an issue accessing compliance or trading data and escalates the issue for resolution.

Use Case: Trading Engine Operations

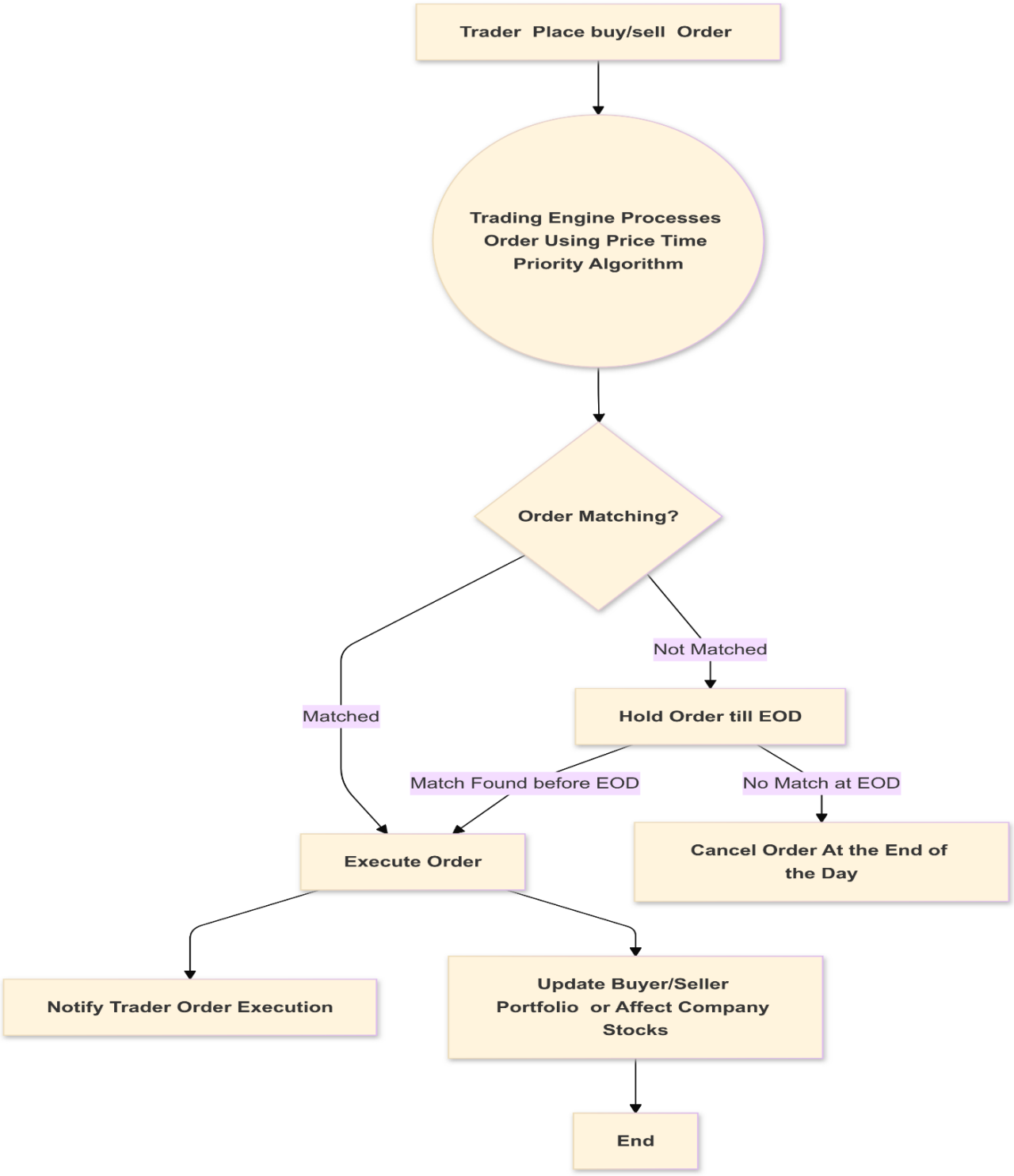
Use Case	Trading Engine Operations
Actors	Trading Engine (Sub-System)
Description	The trading engine automates core trading functionalities, ensuring efficient order matching, execution, and transaction logging.
Preconditions	Valid buy and sell orders must be placed by traders within the system's trading hours.
Post conditions	Orders are matched and executed, user portfolios are updated, and transactions are

	logged and send the notification for the traders.
Events	➢ **Order Matching and Processing**: Matches buy and sell orders using a price-time priority algorithm. ➢ **Trade Execution and Processing**: Executes matched orders and updates the order book and user portfolios. ➢ **Send Order Execution Notifications**: Notifies users about the successful execution of their trades. ➢ **Transaction Logging**: Records all completed transactions for compliance and auditing purposes.
Alternative Events	**Order Mismatch**: If no matching order is found for a placed order, it remains in the pending queue until a match is available and will be cancelled at the end of day.
Exceptions	System Downtime: If the trading engine encounters a failure, ongoing operations are paused, and pending transactions are queued for later processing.

3.1.3. Activity Diagram

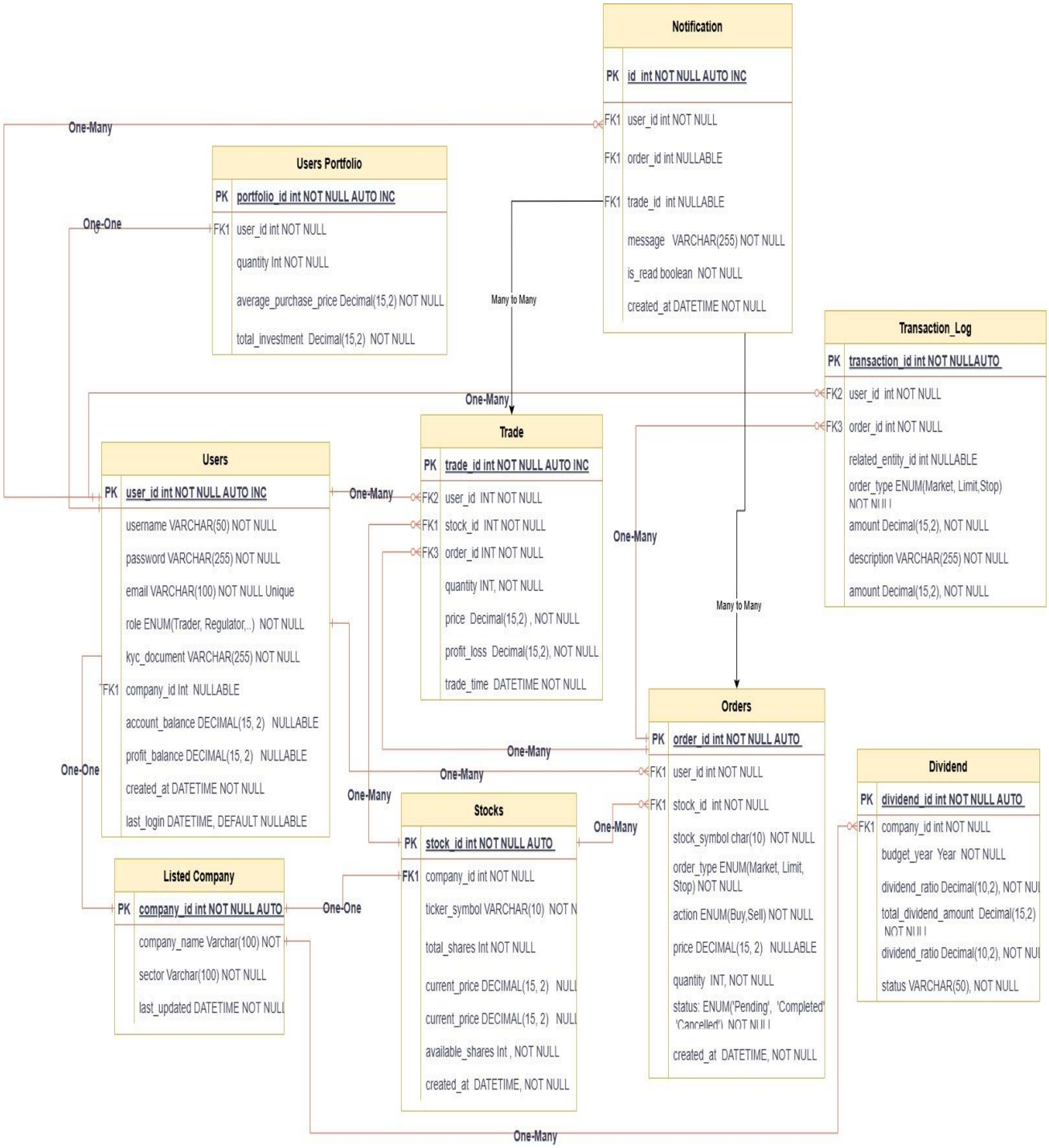


User Role Activity



Trading Engine Order Processing Activity Diagram

3.1.4. Class Model



3.1.5. Data Dictionary

Users

Attribute Name	Data Type	Constraints	Description
user_id	INT	PRIMARY KEY, NOT NULL, AUTO_INCREMENT	Unique identifier for each user.
username	VARCHAR(50)	NOT NULL	User's login name.
password	VARCHAR(255)	NOT NULL	Encrypted password for authentication.
email	VARCHAR(100)	NOT NULL, UNIQUE	User's email address.
role	ENUM	NOT NULL	Role of the user (e.g., Trader, Regulator).
kyc_document	VARCHAR(255)	NOT NULL	KYC document provided by the user.
company_id	INT	NULLABLE	Associated company identifier.
account_balance	DECIMAL(15,2)	NULLABLE	Current account balance.
profit_balance	DECIMAL(15,2)	NULLABLE	Profit/loss balance.
created_at	DATETIME	NOT NULL	Account creation timestamp.
last_login	DATETIME	DEFAULT NULLABLE	Timestamp of the last login.

Users Portfolio

Attribute Name	Data Type	Constraints	Description
portfolio_id	INT	PRIMARY KEY, NOT NULL, AUTO_INCREMENT	Unique portfolio identifier.
user_id	INT	NOT NULL	Foreign key linking to Users table.
quantity	INT	NOT NULL	Quantity of stocks in the portfolio.
average_purchase_price	DECIMAL(15,2)	NOT NULL	Average purchase price of stocks.
total_investment	DECIMAL(15,2)	NOT NULL	Total investment amount.

Stocks

Attribute Name	Data Type	Constraints	Description
stock_id	INT	PRIMARY KEY, NOT NULL, AUTO_INCREMENT	Unique identifier for each stock.
company_id	INT	NOT NULL	Foreign key linking to Listed Company.
ticker_symbol	VARCHAR(10)	NOT NULL	Stock ticker symbol.
total_shares	INT	NOT NULL	Total shares issued by the company.
current_price	DECIMAL(15,2)	NOT NULL	Current stock price.
available_shares	INT	NOT NULL	Number of shares available for trading.
created_at	DATETIME	NOT NULL	Stock record creation date.

Trade

Attribute Name	Data Type	Constraints	Description
trade_id	INT	PRIMARY KEY, NOT NULL, AUTO_INCREMENT	Unique identifier for each trade.
user_id	INT	NOT NULL	Foreign key linking to Users.
stock_id	INT	NOT NULL	Foreign key linking to Stocks.
order_id	INT	NOT NULL	Foreign key linking to Orders.
quantity	INT	NOT NULL	Quantity of stocks traded.
price	DECIMAL(15,2)	NOT NULL	Price at which the trade occurred.
profit_loss	DECIMAL(15,2)	NOT NULL	Profit or loss from the trade.
trade_time	DATETIME	NOT NULL	Timestamp of the trade.

Orders

Attribute Name	Data Type	Constraints	Description
order_id	INT	PRIMARY KEY, NOT NULL, AUTO_INCREMENT	Unique identifier for each order.
user_id	INT	NOT NULL	Foreign key linking to Users.
stock_symbol	VARCHAR(10)	NOT NULL	Stock symbol for the order.
order_type	ENUM	NOT NULL	Type of order (e.g., Market, Limit, Stop).
action	ENUM	NOT NULL	Action type (e.g., Buy,

			Sell).
price	DECIMAL(15,2)	NOT NULL	Order price.
quantity	INT	NOT NULL	Quantity of stocks in the order.
status	ENUM	NOT NULL	Order status (e.g., Pending, Completed, and Cancelled).
created_at	DATETIME	NOT NULL	Timestamp when the order was created.

Listed Company

Attribute Name	Data Type	Constraints	Description
company_id	INT	PRIMARY KEY, NOT NULL, AUTO_INCREMENT	Unique identifier for the company.
company_name	VARCHAR(100)	NOT NULL	Name of the listed company.
sector	VARCHAR(100)	NOT NULL	Sector of the company.
last_updated	DATETIME	NOT NULL	Last update timestamp for company information.

Dividend

Attribute Name	Data Type	Constraints	Description
dividend_id	INT	PRIMARY KEY, NOT NULL, AUTO_INCREMENT	Unique identifier for each dividend entry.
company_id	INT	NOT NULL	Foreign key linking to Listed Company table.
budget_year	YEAR	NOT NULL	Year for which the dividend is allocated.
dividend_ratio	DECIMAL(10,2)	NOT NULL	Ratio of dividend payout.
total_dividend_amount	DECIMAL(15,2)	NOT NULL	Total amount allocated as dividends.
dividend_ratio	DECIMAL(10,2)	NOT NULL	Dividend percentage.
status	VARCHAR(50)	NOT NULL	Status of the dividend (e.g., Paid, Pending).

Transaction Log

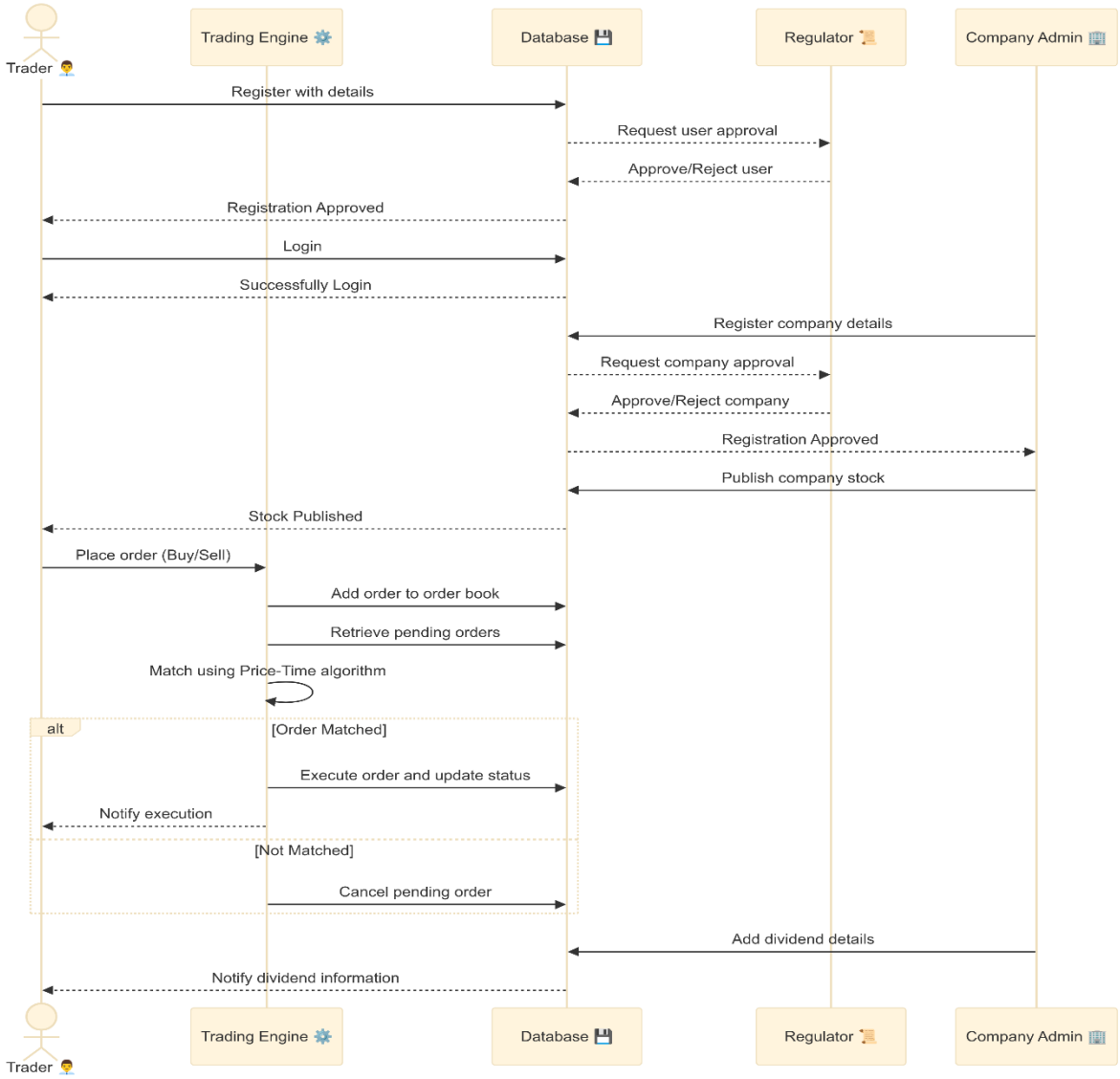
Attribute Name	Data Type	Constraints	Description
transaction_id	INT	PRIMARY KEY, NOT NULL, AUTO_INCREMENT	Unique identifier for the transaction log entry.
user_id	INT	NOT NULL	Foreign key linking to Users table.
order_id	INT	NOT NULL	Foreign key linking to Orders

			table.
related_entity_id	INT	NOT NULL	Entity ID related to the transaction.
order_type	ENUM	NOT NULL	Type of order (e.g., Market, Limit).
amount	DECIMAL(15,2)	NOT NULL	Amount involved in the transaction.
description	VARCHAR(255)	NOT NULL	Details about the transaction.

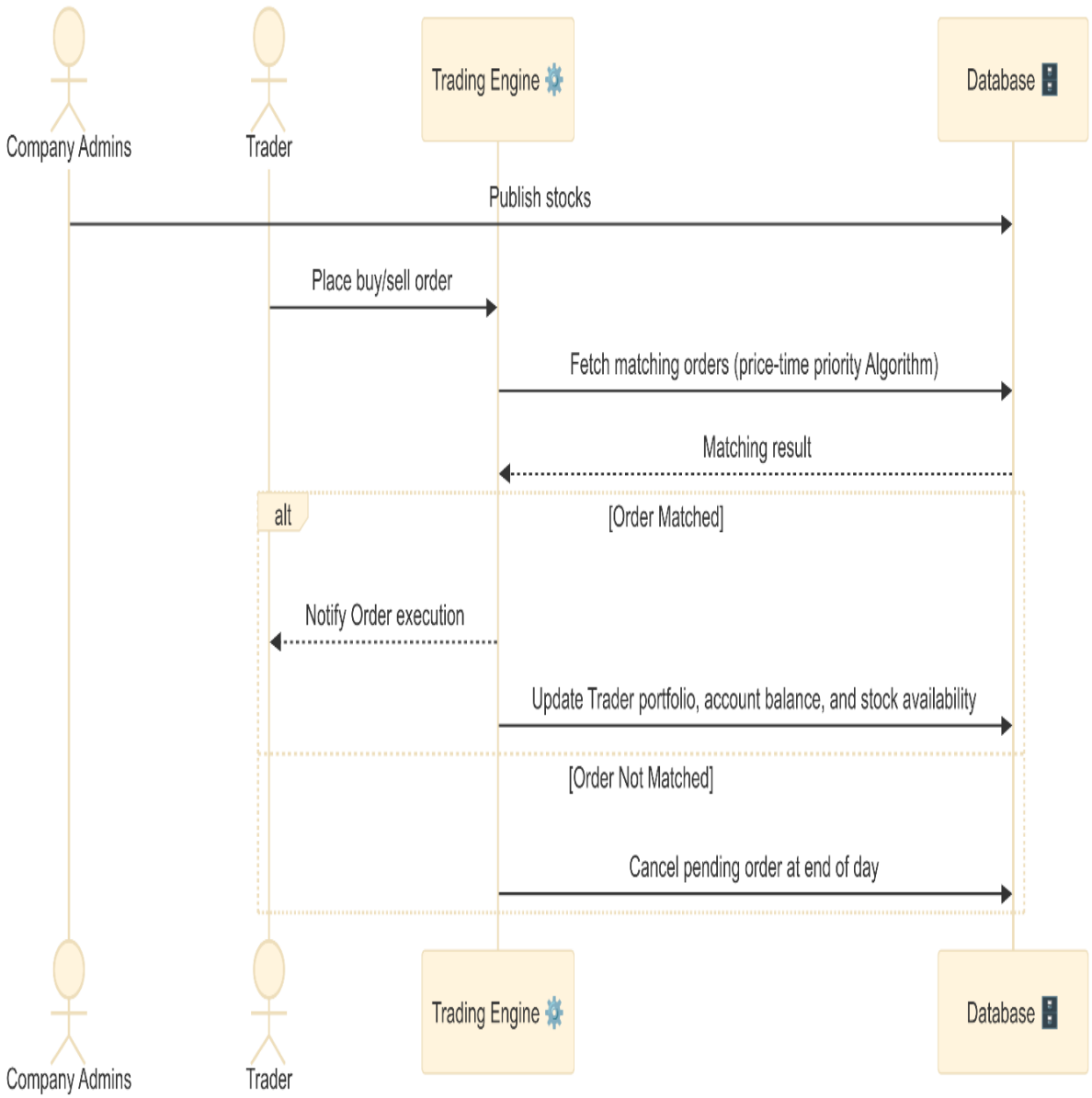
Notification

Attribute Name	Data Type	Constraints	Description
id	INT	PRIMARY KEY, NOT NULL, AUTO_INCREMENT	Unique identifier for each notification.
user_id	INT	NOT NULL	Foreign key linking to Users table.
order_id	INT	NULLABLE	Foreign key linking to Orders table.
trade_id	INT	NULLABLE	Foreign key linking to Trade table.
message	VARCHAR(255)	NOT NULL	Notification message.
is_read	BOOLEAN	NOT NULL	Indicates whether the notification has been read.
created_at	DATETIME	NOT NULL	Timestamp when the notification was created.

3.1.6. Sequence Diagram



Trading Activity within User Roles



Trading Engine Process Sequence Diagram