KLE Society's

KLE Technological University



**Open Ended (OE) Assessment Report**

**On**

**TRADING APPLICATION**

**Object Oriented Programming (20ECSC204)  
Object Oriented Programming Lab (20ECSP203)**

Submitted by

| **Name** | **Roll no** | **SRN** |
| --- | --- | --- |
| ANUSHIKA M K | 215 | 01FE21BCS062 |
| RITESH HIREMATH | 226 | 01FE21BCS151 |
| **Team Number:** 14 | | |

Faculty In-charge:

MANJUNATH GONAL SIR

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

Academic year 2021-22

| **1.** | **Introduction** | |
| --- | --- | --- |
|  | 1.1 | Overview of the problem statement………………………………………………………...2 |
|  | 1.2 | Features of Application…………………………………………………………………………….2-3 |
| **2.** | **Design** | |
|  | 2.1 | Class Diagrams………………………………………………………………………………………….4 |
| **3.** | **Implementation** | |
|  | 3.1 | Results……………………………………………………………………………………………………...5 |

| **1.**  1.1  1.2 | **Introduction**  Overview of the problem statement  The objective of this C++ project is to develop a Financial Trading Account Management System that enables users to engage in buying and selling transactions of various financial instruments. The system will provide a user-friendly interface to manage financial holdings, track account balance, and perform seamless trades for stocks, bonds, mutual funds, cryptocurrencies, real estate, and commodities.  Features of Application  The following features were considered for making the Financial Trading Account Management System application:  1. Financial Instrument Class Hierarchy:   * The code implements a class hierarchy for different financial instruments, including stocks, bonds, mutual funds, cryptocurrencies, real estate, and commodities. * Each financial instrument is represented by a specific class derived from the base class "FinancialInstrument."   - This feature allows for easy extension of new financial instruments in the future.  2. User Class:   * The "User" class represents users with a unique userID and account balance. * Users can add and remove holdings of different financial instruments based on their trading activities. * The class enables balance management by reducing or increasing the balance after trades.   3. Trading Account Class:   * The "TradingAccount" class acts as an interface for users to interact with the financial instruments and perform trades. |
| --- | --- |

* It provides methods for displaying available financial instruments and performing buying and selling transactions

- The class handles exceptions for insufficient balance, invalid instruments, and insufficient holdings.

4. Exception Handling:

* The code includes custom exception classes like "InsufficientBalanceException," "InvalidInstrumentException," and "InsufficientHoldingException."
* These exceptions are used to handle specific error scenarios during trading transactions, providing meaningful error messages to users.

5. Displaying Available Instruments:

* The "TradingAccount" class provides a method to display the available financial instruments along with their details, such as instrument ID, name, current price, market, and quantity available.
* This feature allows users to view the list of instruments they can trade.

6. Performing Transactions:

* The application enables users to perform buying and selling transactions for different financial instruments.
* Users can specify the instrument ID, quantity, and transaction type (buy/sell) to execute the trades.

7. User Holdings and Balance Display:

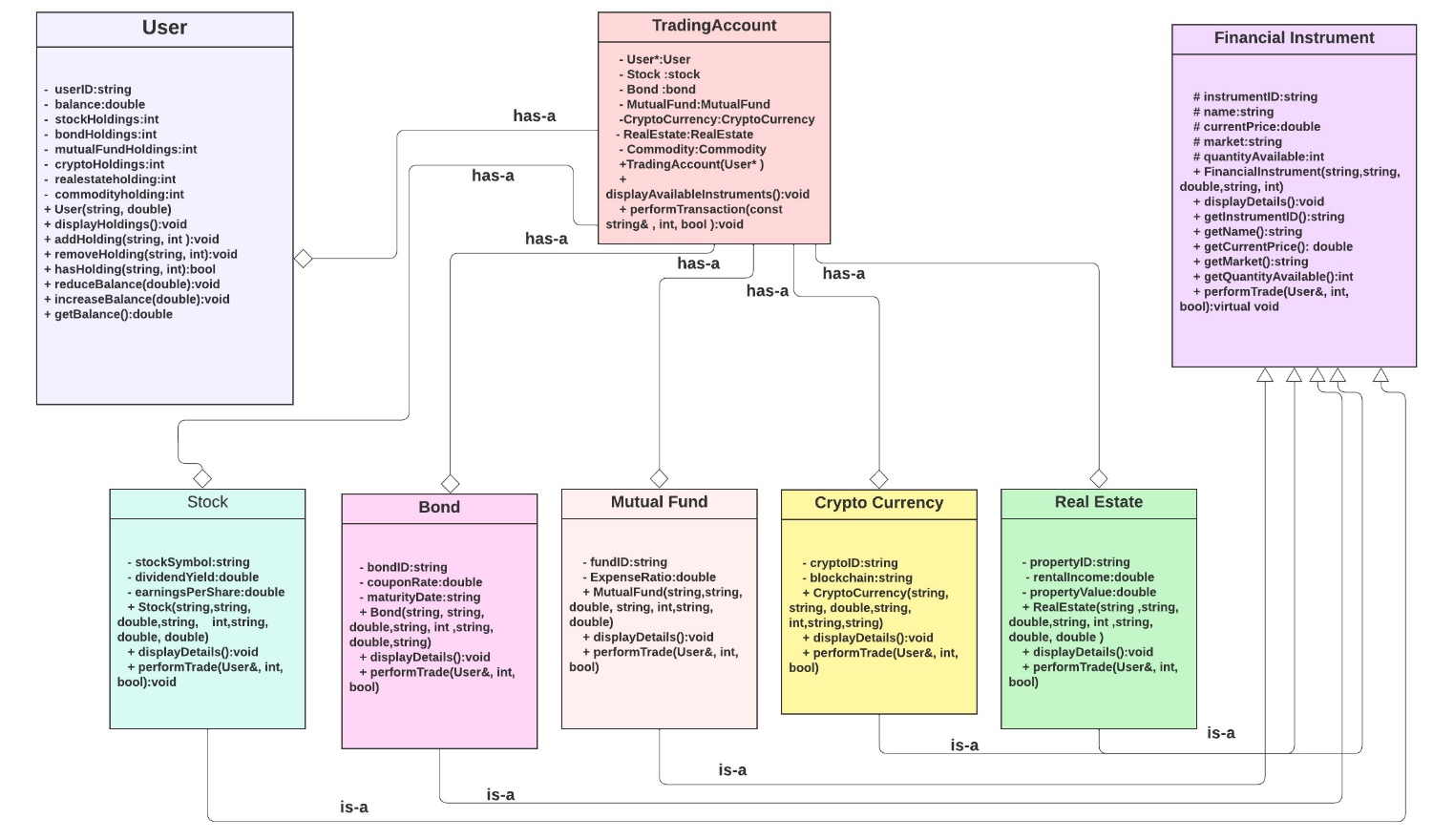
* The "User" class provides a method to display the user's current holdings for various financial instruments.
* Users can view the quantity of each instrument they hold, along with their account balance.

8. User Interaction:

* The main function demonstrates the functionality of the trading account system by creating a user, initializing a trading account for that user, and performing multiple buying and selling transactions.
* The user's holdings and balance are displayed after the transactions to show the impact of the trades.

These features aim to provide users with a basic but functional trading account system where they can manage their holdings, perform trades, and keep track of their account balance for various financial instruments. The code focuses on demonstrating the core functionalities of the

| **2.** | **Design** | |
| --- | --- | --- |
|  | 2.1 | Class Diagrams |



| **3.** | **Implementation** | |
| --- | --- | --- |
|  | 3.1 | Results |
| MAIN:  OUTPUT: |  | Screenshot 2023-07-25 123355Screenshot 2023-07-25 123220 |

