

Software Architecture & Design – Sprint 2

Team Members:

Evan Balson – BAL18466416

Maahia Rahman – RAH23614335

Hala Bakhtiar – BAK23592238

Mahboubia Rezaei – REZ23579670

Roehampton University

Course Number: CMP020N207S

Lecturer: Dr Shekoufeh Kolaheidouz Rahimi

18/03/2025

Overview	3
Key Features:.....	3
Architecture & Design:.....	3
How is this achieved?	3
UML2 Composite Component	4
Business Concept Model	5
Use Case Model.....	6
System Interfaces	8
Business Type Model.....	10
References:	12

Overview

StockShark is a web application designed to provide users with quick and accessible stock data, powered by real time information fetched from the Yahoo Finance API. Whether you are seasoned investor or just starting to explore the stock market, StockShark offers a streamlined experience to analyze and compare stocks within a two-year timeframe.

Key Features:

- Effortless stock search: easily find stocks by their company name or ticker symbol.
- Historical data comparison: visualize and compare the performance of multiple stocks over a customizable period, up to two years.
- Real-time data: leverages the Yahoo Finance API to provide up to date stock information.

Architecture & Design:

StockShark employs clean architecture, a software design that emphasizes separation of concerns and independent from frameworks and external tools. [1]

Clean architecture is a software design pattern aims for:

- Framework independence: allows to easily switch to a different web framework or database without rewriting the heart of StockShark. [1]
- Easy testing: because the business logic is isolated, we can easily write tests to ensure that it works correctly. That leads to a more reliable application. [1]
- User interface independence: the user interface is separate from the core logic. This means we redesign the website without affecting how StockShark processes stock data.[1]
- External API independence: we can change the external API we are using to get data, without impacting the business rules. [1]

How is this achieved?

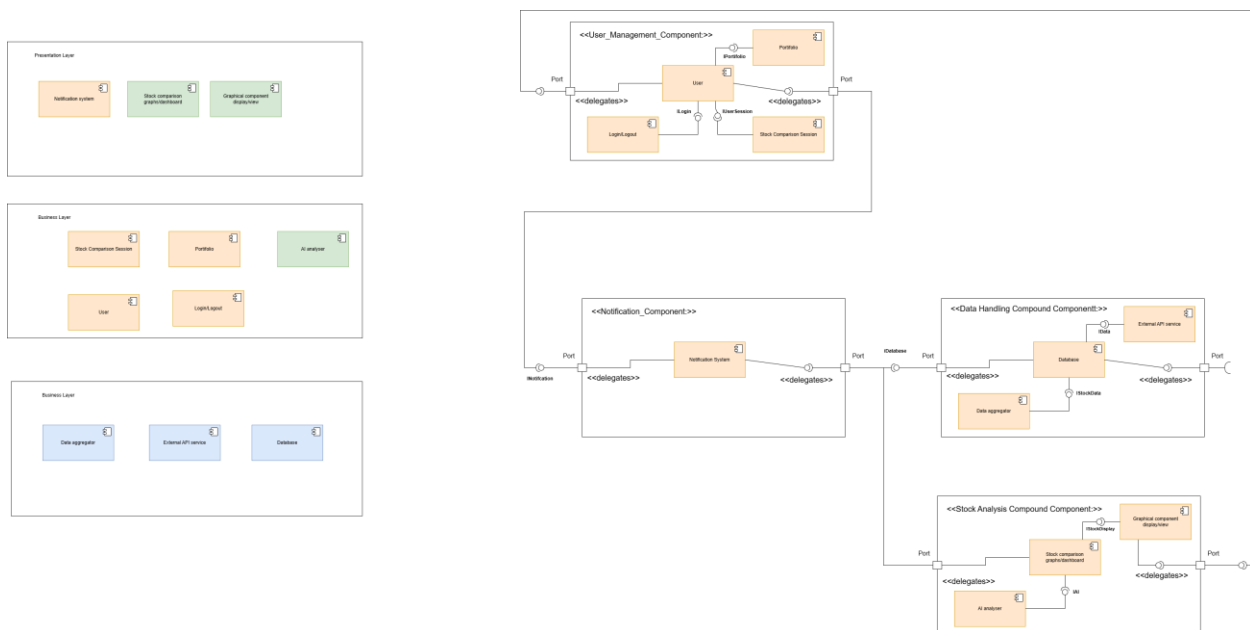
StockShark's architecture is built upon the core principles of Clean Architecture, which aims to create systems that are independent of frameworks, testable, and maintainable. This design follows the SOLID principles, making the system strong and easy to change.

- Single/Closed Principle (OCP): StockShark is designed to be open for extension but closed for modification. New features are added by introducing new code, rather than altering existing, stable code.

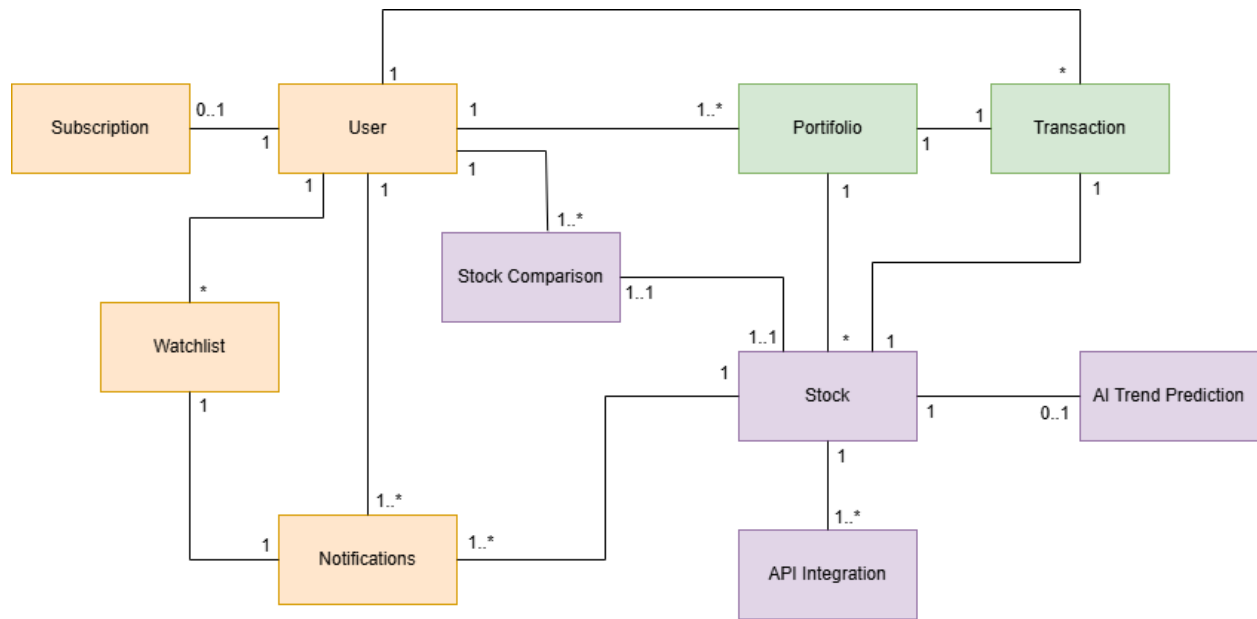
- Dependency Inversion Principle (DIP): High level modules, such as use cases that implement business logic, do not depend on low level modules, such as data access layers. Instead, low level modules depend on abstractions defined by high level modules.

By separating these layers, we create a system that's flexible, maintainable, and easy to understand. This makes it easier for our team to add new features, fix bugs, and adapt to changing requirements, ultimately providing you with a better StockShark experience.

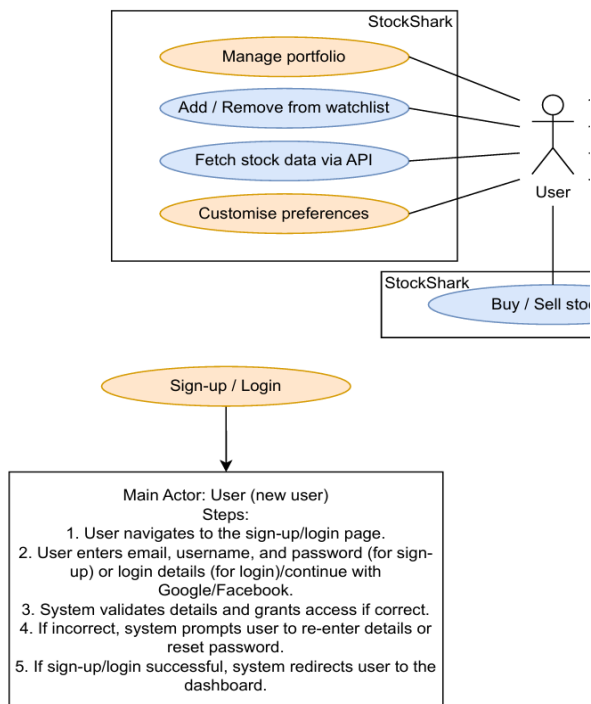
UML2 Composite Component

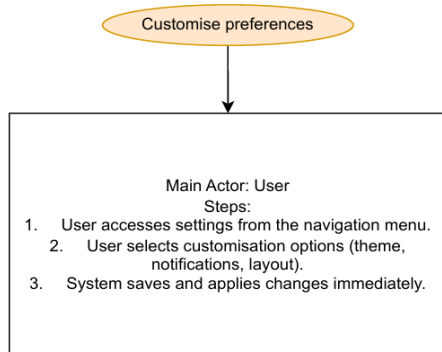
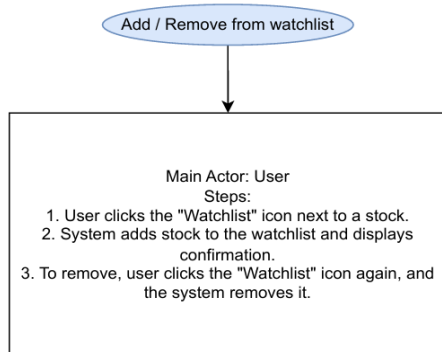
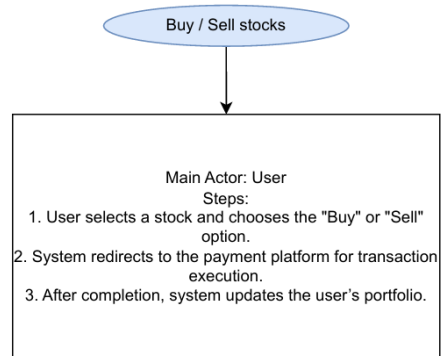
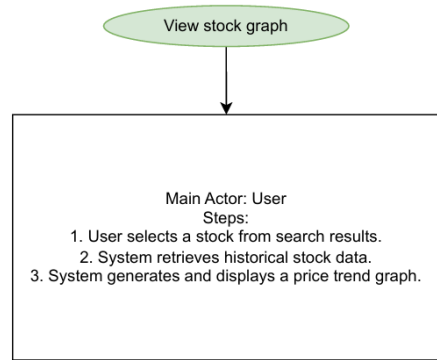


Business Concept Model

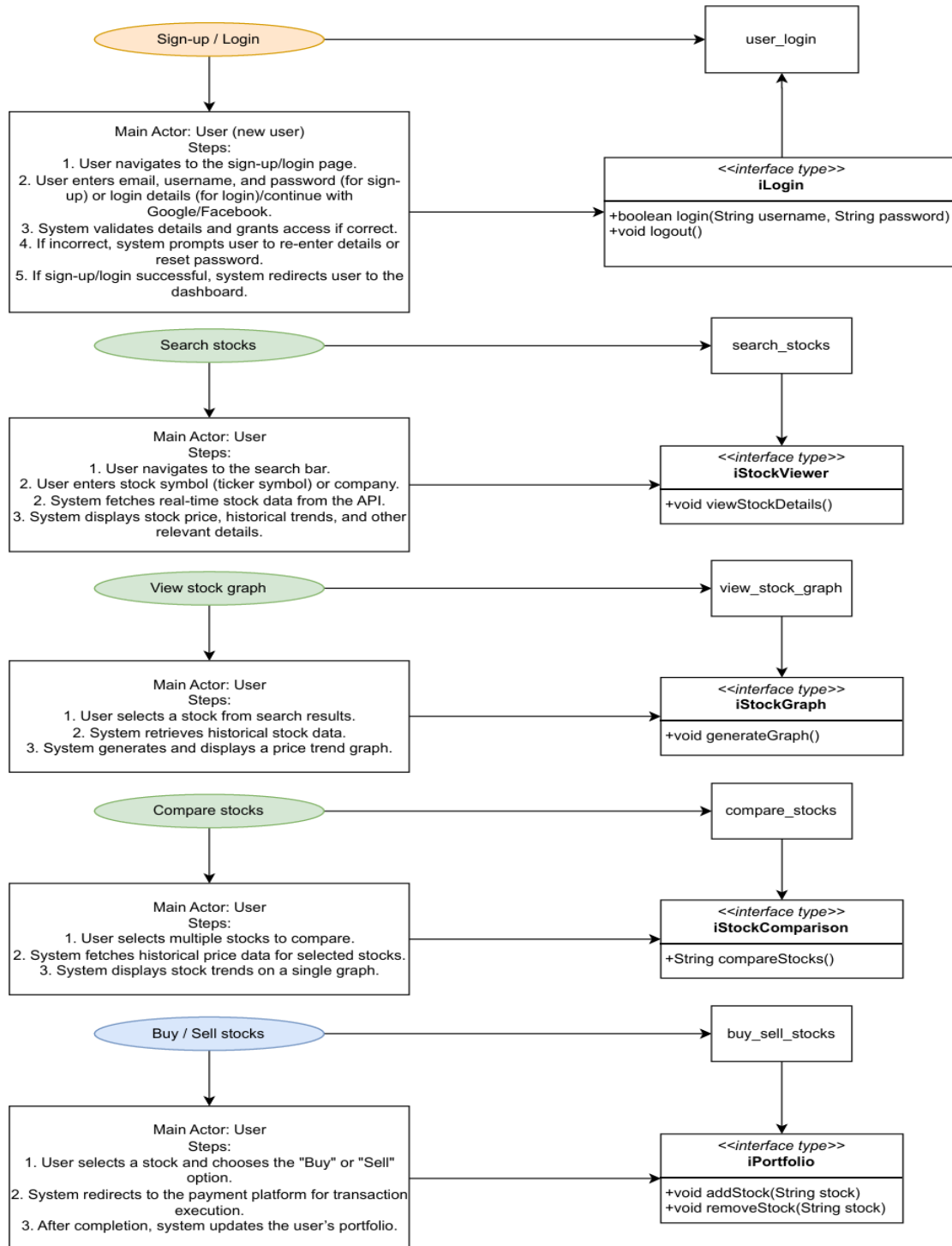


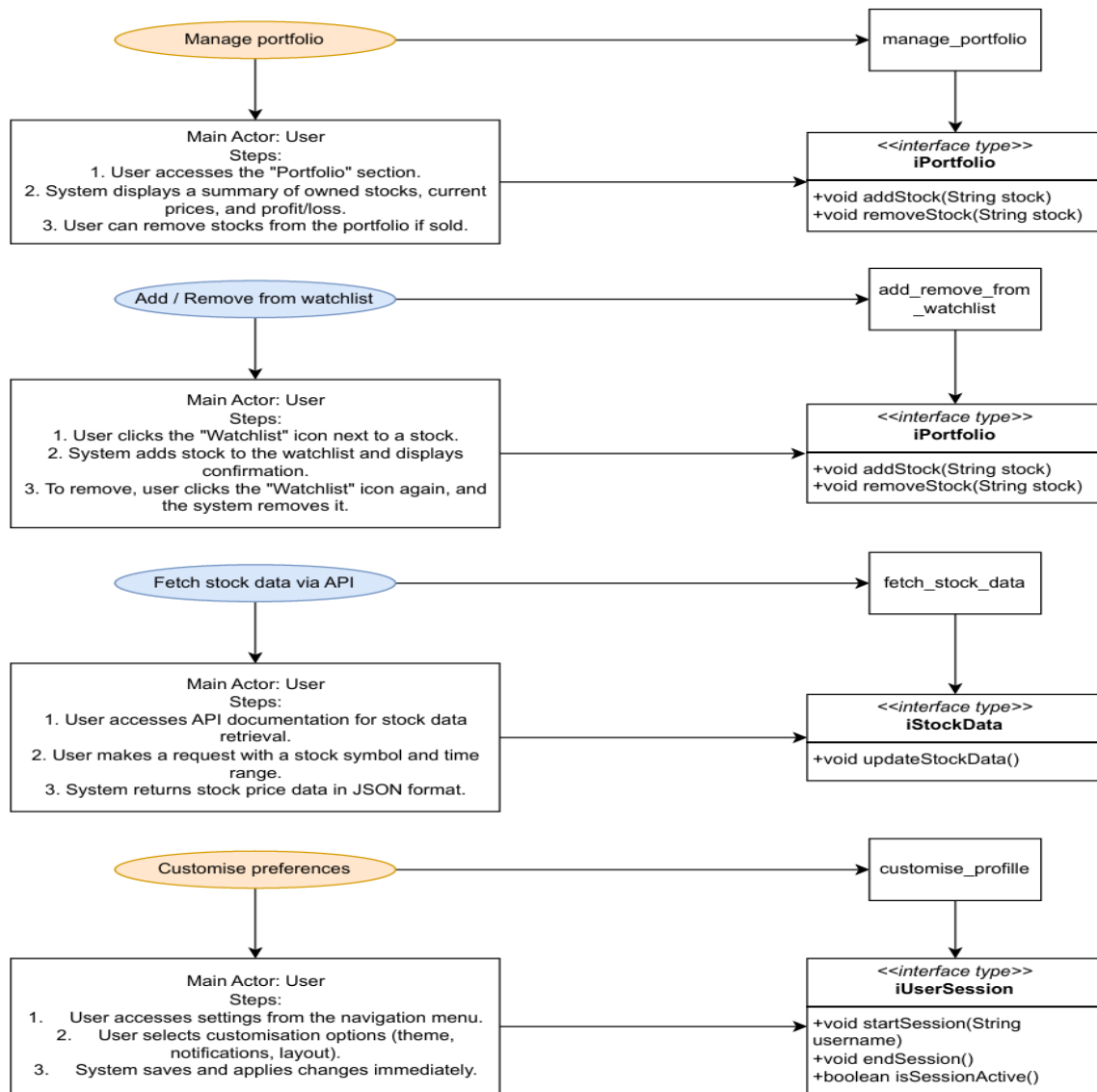
Use Case Model



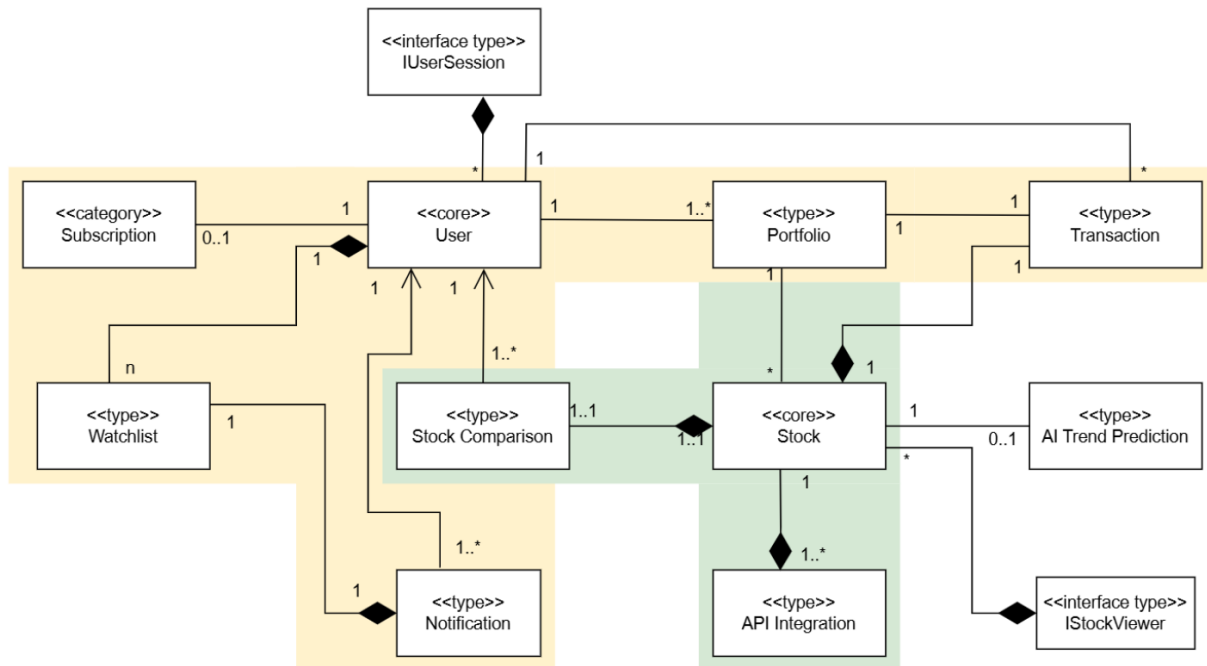


System Interfaces

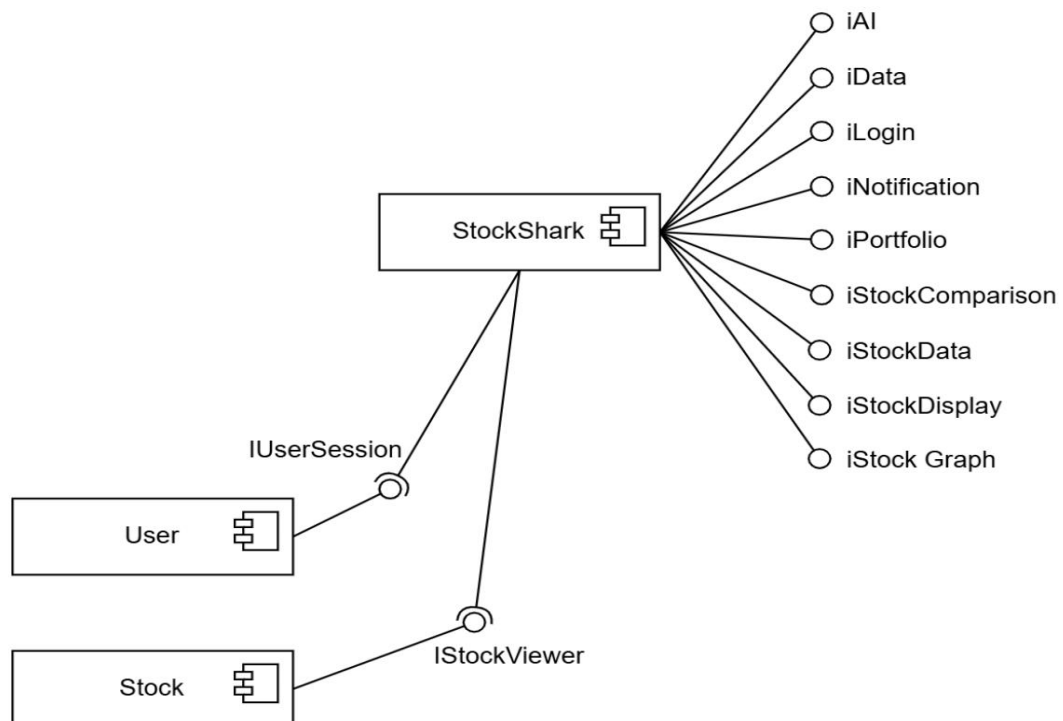




Business Type Model



Initial System Architecture



References:

[1]. GeeksforGeeks, “*Complete Guide to Clean Architecture*”, [Online], Available: [Complete Guide to Clean Architecture - GeeksforGeeks](#), [Accessed: March. 16, 2025]