

DeSign

Team 29



University of Sussex



Contents

[User interface 0](#_Toc482781151)

[Use Case Diagrams 0](#_Toc482781152)

[Sequence diagrams 1](#_Toc482781153)

[UML Class diagram 2](#_Toc482781154)

[Testing Strategy 3](#_Toc482781155)

# User interface

The start button will be activated once a file has been loaded in. this will start the simulation.

The pause button will put the simulation on hold.

**File**

Load File

--------------

Exit

Simulation Settings

Simulation speed:

Simulation Information

Market State: Bull/Bear

Current Simulation Date:

dd/mm/yyyy

Start

Pause

Graph

Client

Companies

Exchanges



The main body of the interface will contain 3 tabs being client, companies and exchanges. There will also be graph visualising the calculation and stock market. The client graph will represent the client’s total worth over time. The companies graph will be showing the companies worth overtime and exchange tab will be showing the exchange indices.

The first part of the information pane will be an interactive bar which can be moved from left to right. Far Left being the slowest and far most right being the fastest that the simulation can run. The bar will be adjusted by the user. This can be used to meet the domain requirement 1 where its part intended as a game and as a learning platform where the user can slow down the simulation to observe the data.

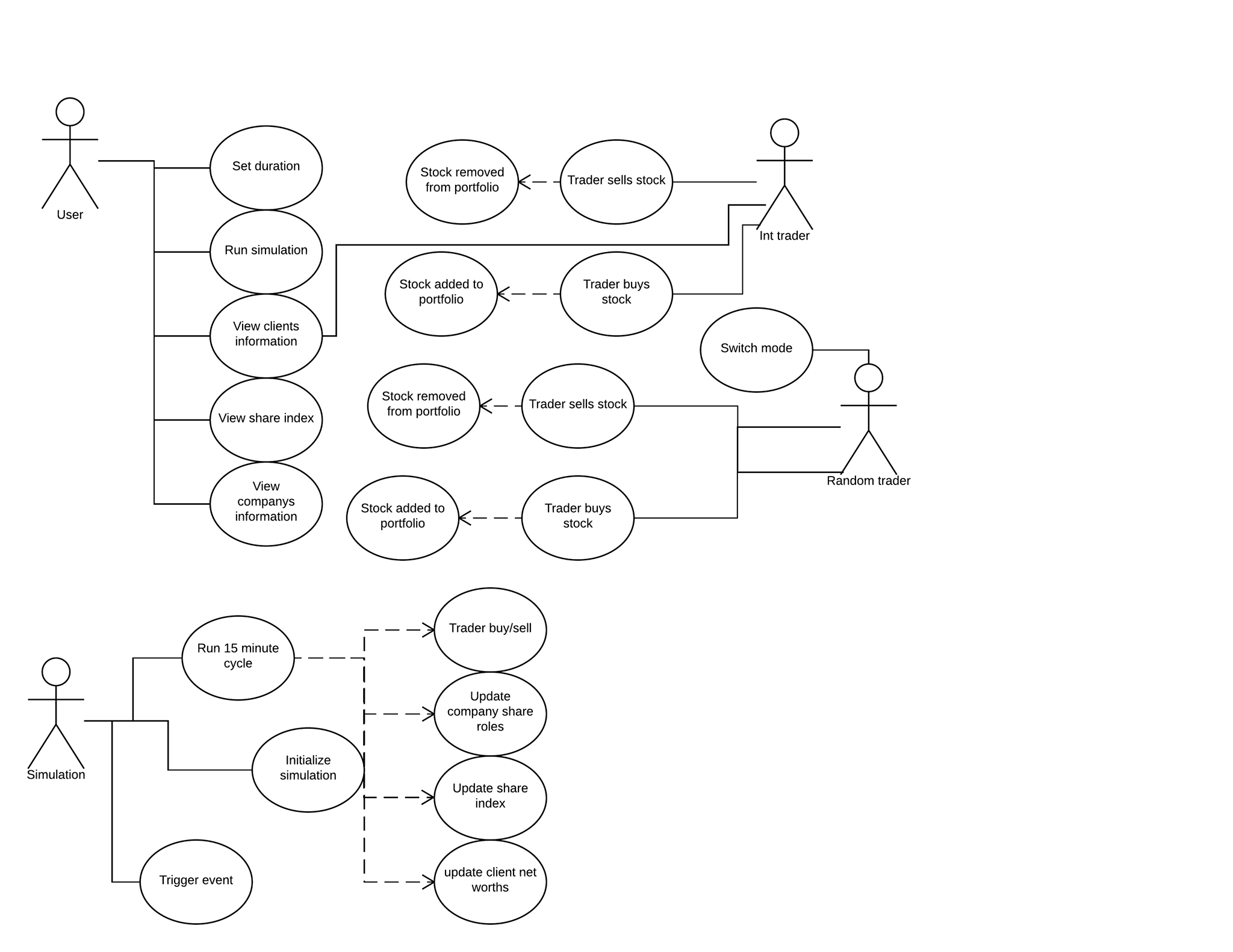
In this part of the information pane there will be 2 piece of information. The first information is the statues of the market as to whether it’s a bull or bear market and the will be updated as the simulation runs.

The second part of the information will be the date of the simulation displayed in full format of date/month/year.

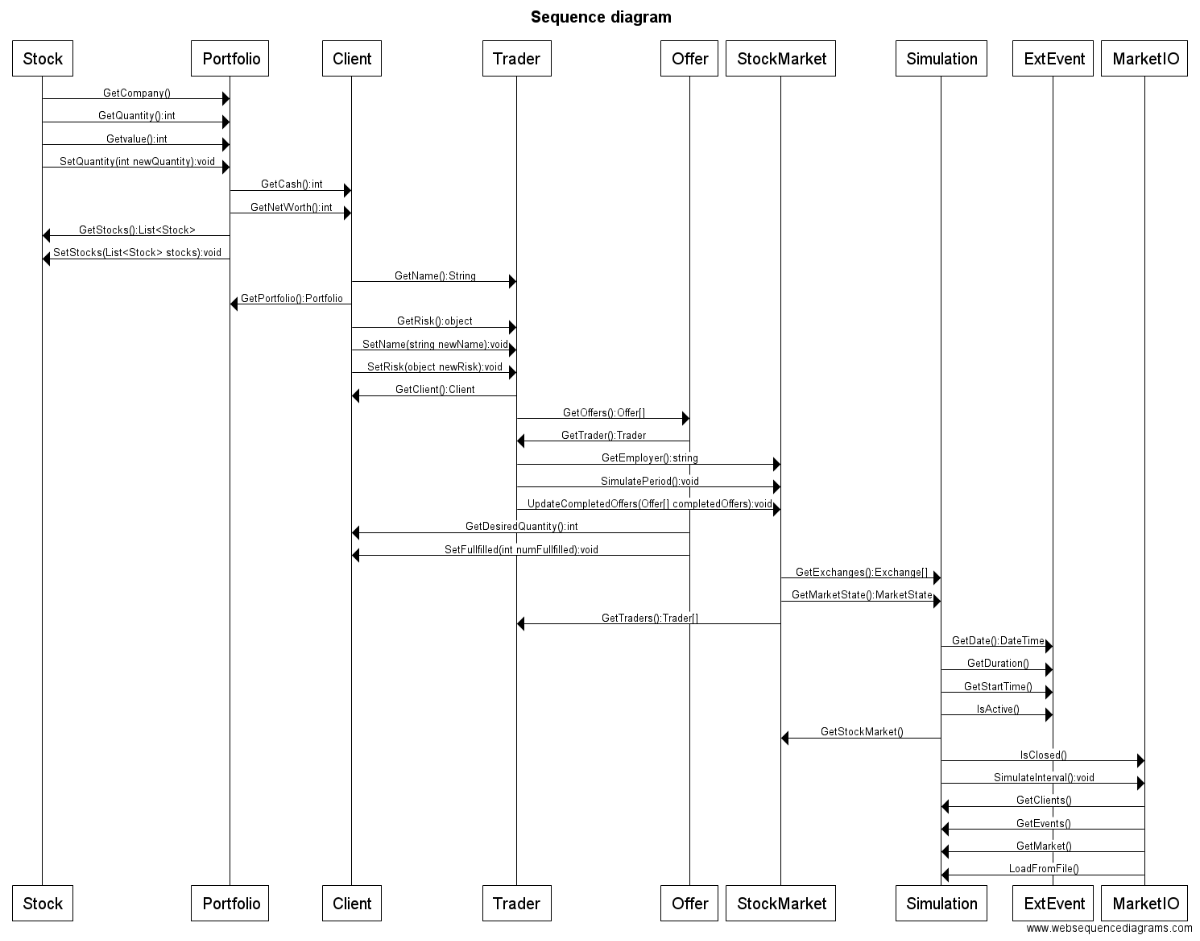
The interface will be having a menu bar which will allow the simulation to be loaded in which can be used to run the stock market.

The menu bar will also have an exit option where users can exit the simulation interface/software.

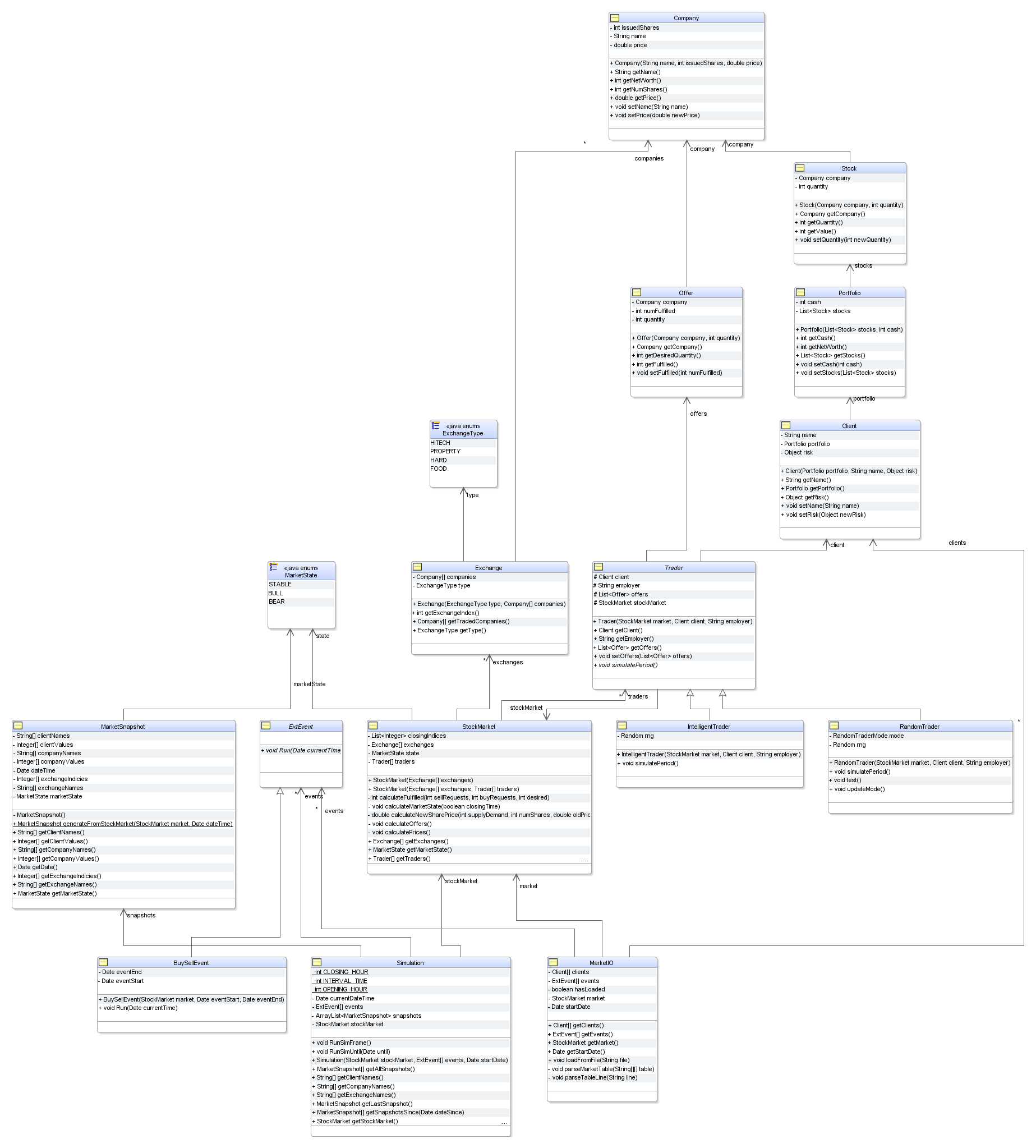
# Use Case Diagrams



# Sequence diagrams



# UML Class diagram



# Testing Strategy

To make the software as dependable and reliable as possible we are going to follow 8 guidelines. We will try achieve all of the 8 guidelines.

1. Limit the visibility of information in a program.
2. Check all inputs for validity
3. Provide a handler for all exceptions
4. Minimise the use of error prone constructs
5. Provide restart capabilities
6. Check array bounds
7. Include timeouts when calling external components
8. Name all constants that represent real world values.