**Trading Simulation**

Inhalt

[Description 2](#_Toc103032634)

[Use Cases 3](#_Toc103032635)

[Use Case-Diagramm 3](#_Toc103032636)

[Use Case-Description 3](#_Toc103032637)

[CRC-Cards 7](#_Toc103032638)

[Class Diagramm 9](#_Toc103032639)

[Domain Model 10](#_Toc103032640)

[Updated Class Diagramm 11](#_Toc103032641)

[Design Changes 11](#_Toc103032642)

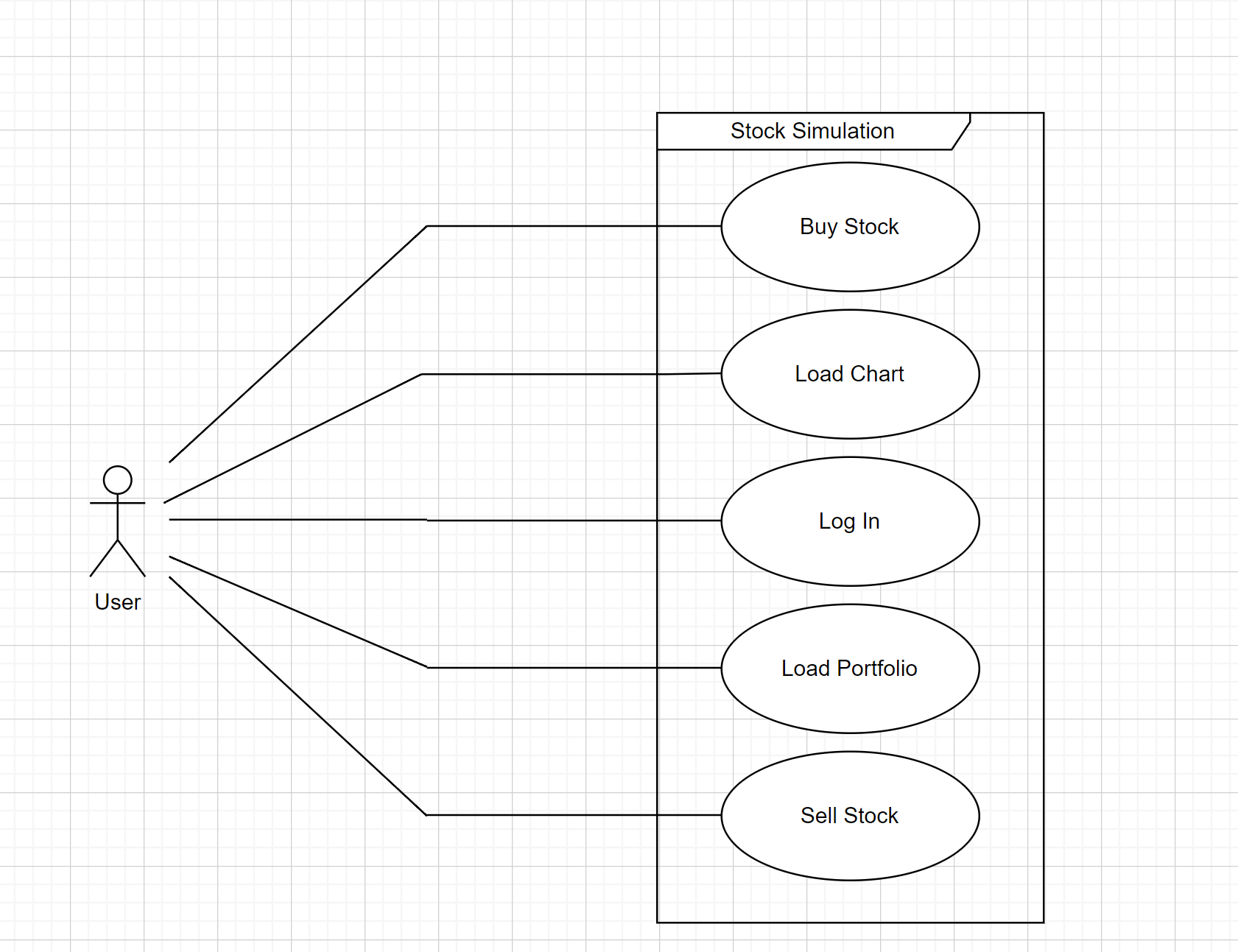
[How to use our program 12](#_Toc103032643)

# Description

Our program is intended to be a stock trading trainer/simulator. For the user it is possible to log in and have a personal portfolio. Since this project does not use a database, the portfolio gets saved into a file after a transaction was executed or the program got closed. The user can buy and sell stocks in the stock market and the stocks will appear in the user profile page. Also, there is an option to search for a specific stock and look at its stats and a chart. Next to the stock and user page there will be a page to display all stocks and the base information like price etc. The goal of this simulator is to gain as much money as possible.

# Use Cases

## Use Case-Diagramm



## Use Case-Description

|  |  |
| --- | --- |
| **Use Case** | Login |
| **Use Case ID** | 1 |
| **Description** | The user can log into his account |
| **Precondition** | User has an account |
| **Actors (Primary)** | user |
| **Procedure:**   1. Enter username and password into text fields 2. Click “Login” | |
| **Postcondition:** | User has access to his account |
| **Alternative flows:** Wrong credentials | |

|  |  |
| --- | --- |
| **Use Case** | Wrong credentials |
| **Use Case ID** | 1.1 |
| **Description** | The user tries to log into an account with wrong credentials |
| **Precondition** | none |
| **Actors (Primary)** | user |
| **Procedure:**   1. Enter username and password into text fields 2. Click “Login” 3. User gets informed that his login data was wrong | |
| **Postcondition:** | none |

|  |  |
| --- | --- |
| **Use Case** | Load Chart |
| **Use Case ID** | 2 |
| **Description** | The user loads a chart from a stock |
| **Precondition** | The user is authenticated, the stock is available in the system |
| **Actors (Primary)** | user |
| **Procedure:**   1. Search for a stock 2. The chart of the stock should be displayed | |
| **Postcondition:** | none |

|  |  |
| --- | --- |
| **Use Case** | Buy a stock |
| **Use Case ID** | 3 |
| **Description** | The user can buy a specific stock |
| **Precondition** | The user is authenticated |
| **Actors (Primary)** | user |
| **Procedure:**   1. Load chart of desired stock 2. Click “Buy Stock” 3. Money is withdrawn from user 4. Share gets transferred to user’s portfolio | |
| **Postcondition:** | User owns shares |
| **Alternative flows:** Not enough money | |

|  |  |
| --- | --- |
| **Use Case** | Not enough money |
| **Use Case ID** | 3.1 |
| **Description** | The user buys a specific stock |
| **Precondition** | The user is authenticated |
| **Actors (Primary)** | user |
| **Procedure:**   1. Alternative flow starts after step 2 in the Main flow 2. Money cannot be widthdrawn because user hasn’t enough money 3. Transaction stops | |
| **Postcondition:** | none |

|  |  |
| --- | --- |
| **Use Case** | Sell a stock |
| **Use Case ID** | 4 |
| **Description** | The user sells a specific stock |
| **Precondition** | The user is authenticated, owns shares of stock to sell |
| **Actors (Primary)** | user |
| **Procedure:**   1. Load chart of desired stock 2. Click “Sell Stock” 3. Share gets withdrawn from user 4. Money gets transferred to user’s portfolio | |
| **Postcondition:** | none |
| **Alternative flows:** none | |

|  |  |
| --- | --- |
| **Use Case** | Load Portfolio |
| **Use Case ID** | 5 |
| **Description** | The user retrieves his own portfolio to see his shares and his balance |
| **Precondition** | The user is authenticated |
| **Actors (Primary)** | user |
| **Procedure:**   1. Load Portfolio 2. The balance and owned share of the user are displayed | |
| **Postcondition:** | none |
| **Alternative flows:** none | |

# CRC-Cards

|  |  |
| --- | --- |
| **Class:** Stock  **Responsibilities**  - Blueprint of Stock | **Collaborators** |

|  |  |
| --- | --- |
| **Class:** Portfolio  **Responsibilities**  - Holds stocks | **Collaborators**  Share |

|  |  |
| --- | --- |
| **Class:** Display  **Responsibilities**  - UI | **Collaborators**  UserDisplay  StockOverviewDisplay  StockDisplay  Searchbar |

|  |  |
| --- | --- |
| **Class:** Filehandler  **Responsibilities**  - read and write to file | **Collaborators** |

|  |  |
| --- | --- |
| **Class:** StockDisplay  **Responsibilities**  - Display a specific stock | **Collaborators**  Stock |

|  |  |
| --- | --- |
| **Class:** StockOverviewDisplay  **Responsibilities**  - Displays a list of stocks | **Collaborators**  Stock  Searchbar |

|  |  |
| --- | --- |
| **Class:** UserDisplay  **Responsibilities**  - Displays the user and his portfolio | **Collaborators**  User  Portfolio |

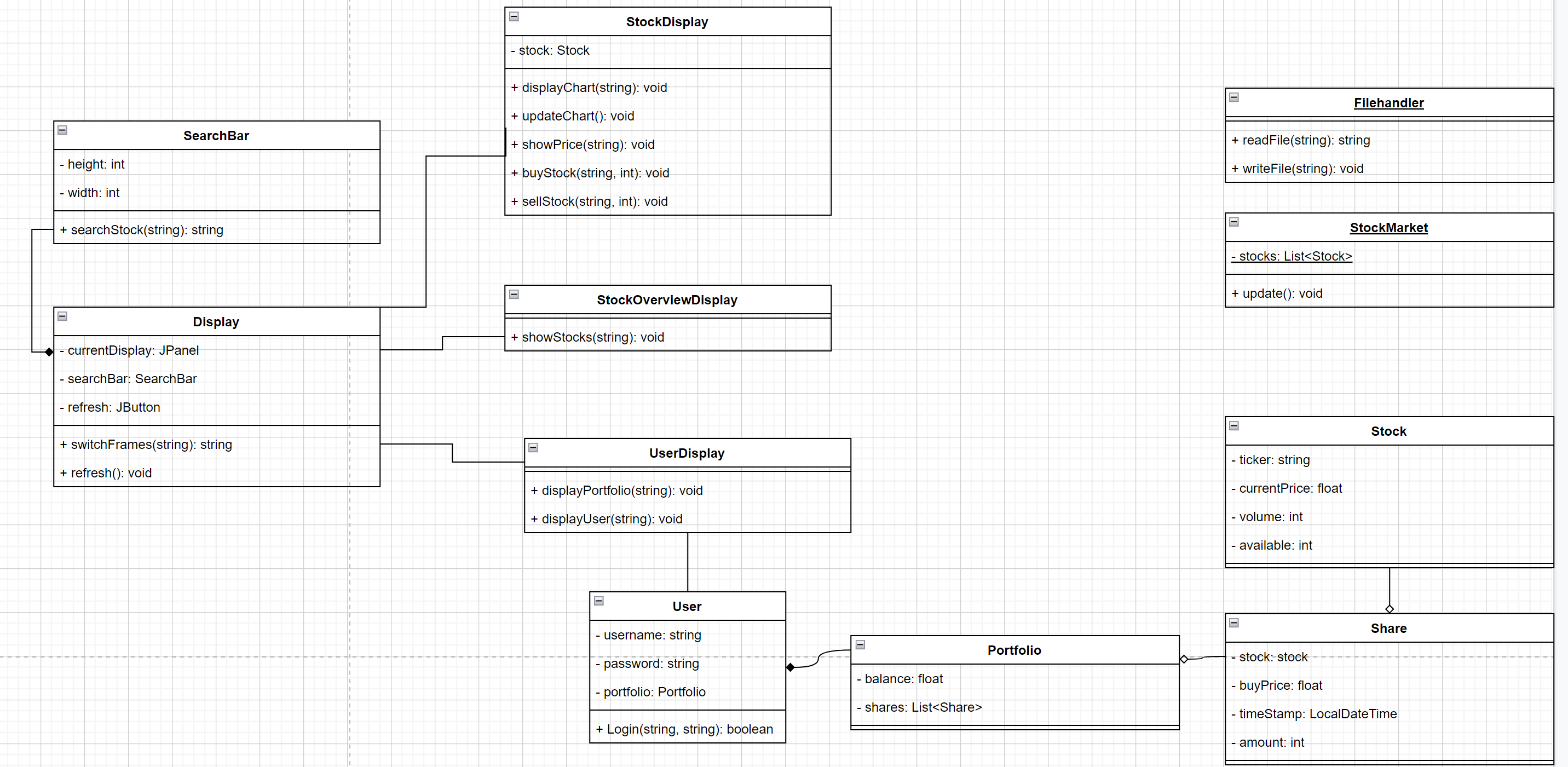
|  |  |
| --- | --- |
| **Class:** Searchbar  **Responsibilities**  - Option to search for a stock | **Collaborators**  Stock |

|  |  |
| --- | --- |
| **Class:** User  **Responsibilities**  - User credentials | **Collaborators**  Portfolio |

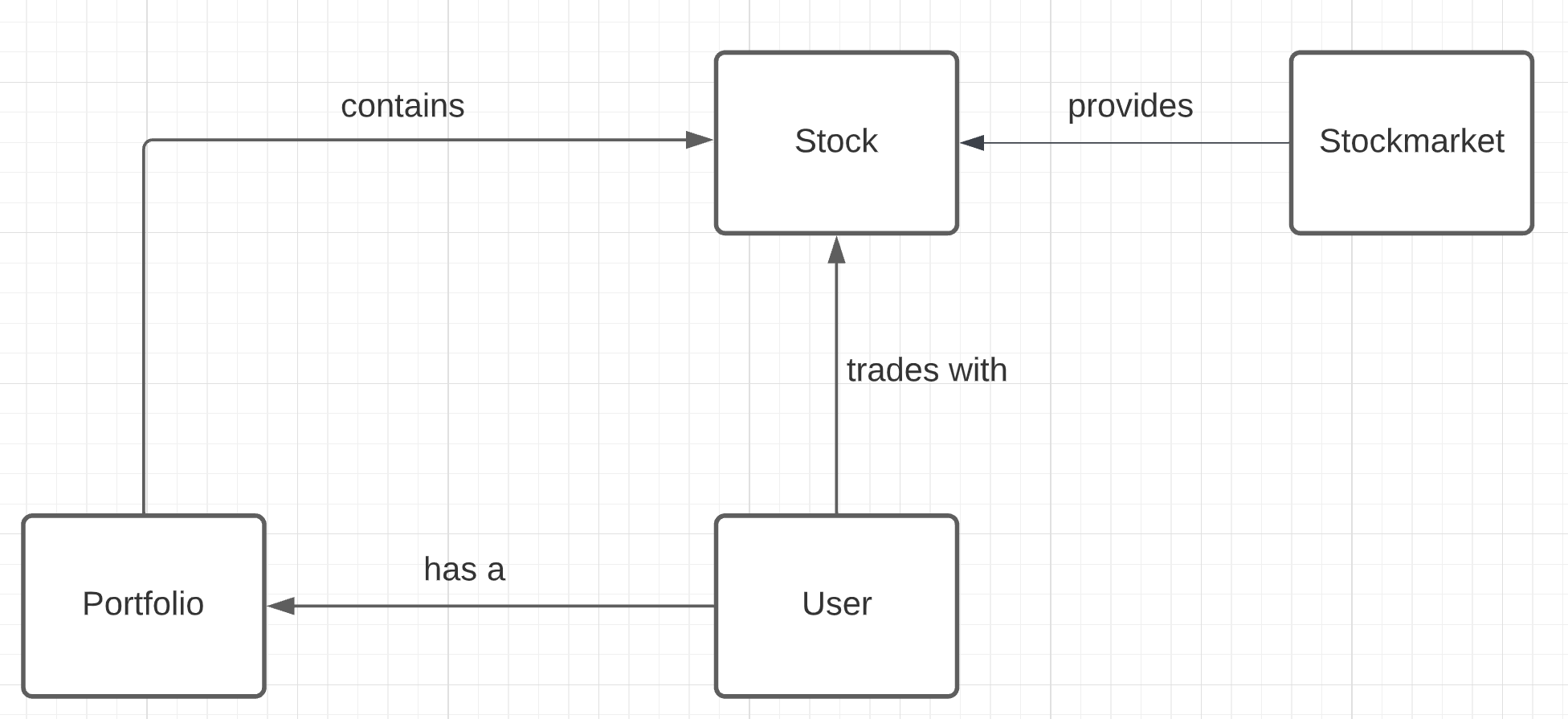
|  |  |
| --- | --- |
| **Class:** Share  **Responsibilities**  - holds stock, buying price and amount of held stock | **Collaborators**  Stock |

|  |  |
| --- | --- |
| **Class:** StockMarket  **Responsibilities**  - holds all stocks  - updates stocks | **Collaborators** |

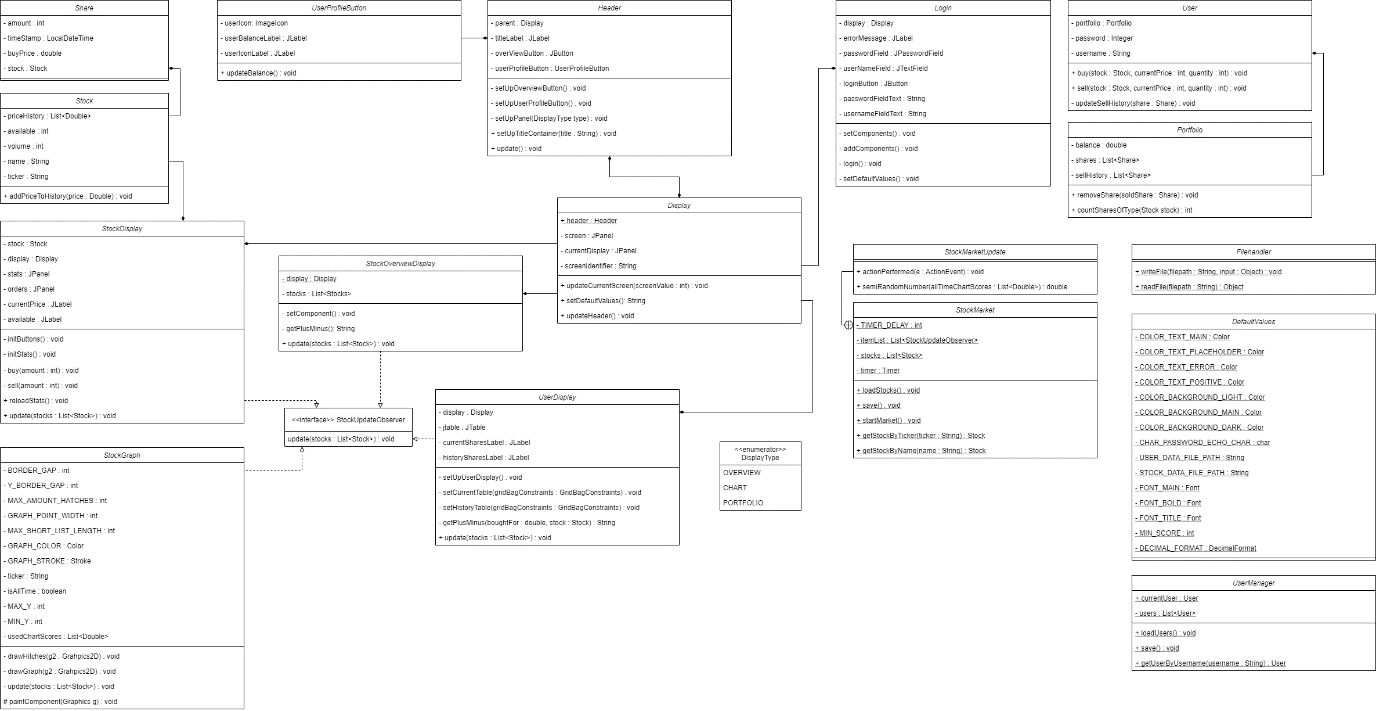
# Class Diagramm



# Domain Model



# Updated Class Diagramm



*Because of the size of the diagram, another image of it can be found in the folder.*

# Design Changes

The biggest change in the design is the use of the observer pattern, which we used to update all the displays with the new generated data. For the update method we added a list of all the displays using polymorphism. Next to the new pattern, we added a login page as a GUI, because of technical reasons, we could not add the search bar but instead added a user profile button, which displays the balance of the user. Is a transaction done, the balance gets updated. Instead of a save button, we replaced it with a hook, so that by closing the frame, the current data is stored. The use cases didn’t change since before.

# How to use our program

To log in you can either use the current users, which usernames are andy0 - andy9, whereby the password the same is as the username. Or you can comment out the code in the main file and a new User to the user manager list. If not already commented in, you must save the user manager before, as the code is loading the data from the file and overrides the list elements. Since this program adds another double each few seconds, the program could run out of memory if used to long. If the frame is white, you have to resize the window to see the actual program, since it is a bug from java swing and not from our program, we couldn’t prevent it initially. As soon as the overview page is displayed, you can click on a stock after which it will reroute you to the stock page where you can see the price history as a graph, whereby you can change it form all time to the last few. Also, you can buy and sell stocks, after which you can press the button on the top right, where you will be rerouted again, but this time you will be on the user page. This page shows you all your current stocks and all the stocks you sold.