

# Trading Simulation

## Inhalt

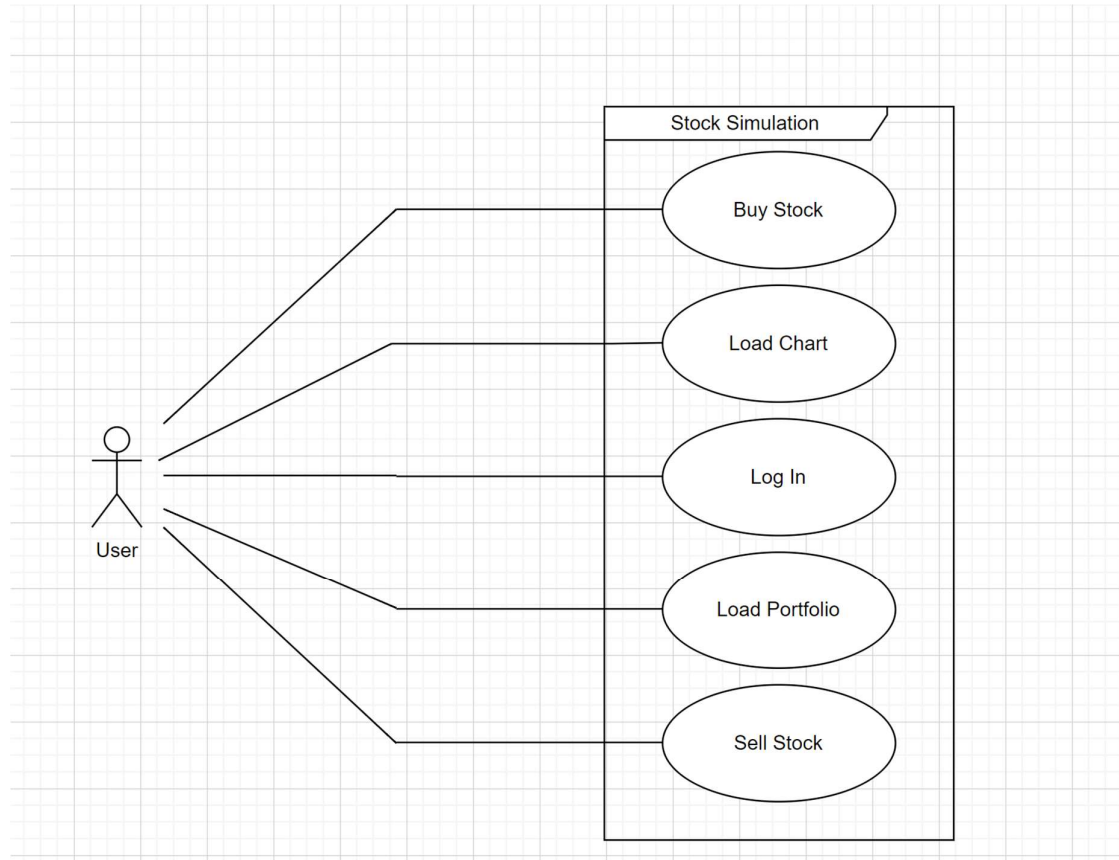
- Description ..... 2
- Use Cases ..... 3
  - Use Case-Diagramm ..... 3
  - Use Case-Description ..... 3
- CRC-Cards ..... 7
- Class Diagramm ..... 9
- Domain Model ..... 10

### 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

<b>Use Case</b>	Login
<b>Use Case ID</b>	1
<b>Description</b>	The user can log into his account
<b>Precondition</b>	User has an account
<b>Actors (Primary)</b>	user
<b>Procedure:</b> <ol style="list-style-type: none"><li>1. Enter username and password into text fields</li><li>2. Click "Login"</li></ol>	
<b>Postcondition:</b>	User has access to his account
<b>Alternative flows:</b> Wrong credentials	

<b>Use Case</b>	Wrong credentials
<b>Use Case ID</b>	1.1
<b>Description</b>	The user tries to log into an account with wrong credentials
<b>Precondition</b>	none
<b>Actors (Primary)</b>	user
<b>Procedure:</b> <ol style="list-style-type: none"> <li>1. Enter username and password into text fields</li> <li>2. Click "Login"</li> <li>3. User gets informed that his login data was wrong</li> </ol>	
<b>Postcondition:</b>	none

<b>Use Case</b>	Load Chart
<b>Use Case ID</b>	2
<b>Description</b>	The user loads a chart from a stock
<b>Precondition</b>	The user is authenticated, the stock is available in the system
<b>Actors (Primary)</b>	user
<b>Procedure:</b> <ol style="list-style-type: none"> <li>1. Search for a stock</li> <li>2. The chart of the stock should be displayed</li> </ol>	
<b>Postcondition:</b>	none

<b>Use Case</b>	Buy a stock
<b>Use Case ID</b>	3
<b>Description</b>	The user can buy a specific stock
<b>Precondition</b>	The user is authenticated
<b>Actors (Primary)</b>	user
<b>Procedure:</b> <ol style="list-style-type: none"> <li>3. Load chart of desired stock</li> <li>4. Click "Buy Stock"</li> <li>5. Money is withdrawn from user</li> <li>6. Share gets transferred to user's portfolio</li> </ol>	
<b>Postcondition:</b>	User owns shares

**Alternative flows:** Not enough money

<b>Use Case</b>	Not enough money
<b>Use Case ID</b>	3.1
<b>Description</b>	The user buys a specific stock
<b>Precondition</b>	The user is authenticated
<b>Actors (Primary)</b>	user
<b>Procedure:</b> <ol style="list-style-type: none"> <li>1. Alternative flow starts after step 2 in the Main flow</li> <li>2. Money cannot be withdrawn because user hasn't enough money</li> <li>3. Transaction stops</li> </ol>	
<b>Postcondition:</b>	none

<b>Use Case</b>	Sell a stock
<b>Use Case ID</b>	4
<b>Description</b>	The user sells a specific stock
<b>Precondition</b>	The user is authenticated, owns shares of stock to sell
<b>Actors (Primary)</b>	user
<b>Procedure:</b> <ol style="list-style-type: none"> <li>1. Load chart of desired stock</li> <li>2. Click "Sell Stock"</li> <li>3. Share gets withdrawn from user</li> <li>4. Money gets transferred to user's portfolio</li> </ol>	
<b>Postcondition:</b>	none
<b>Alternative flows:</b> none	

<b>Use Case</b>	Load Portfolio
<b>Use Case ID</b>	5
<b>Description</b>	The user retrieves his own portfolio to see his shares and his balance
<b>Precondition</b>	The user is authenticated

<b>Actors (Primary)</b>	user
<b>Procedure:</b> <ol style="list-style-type: none"> <li>1. Load Portfolio</li> <li>2. The balance and owned share of the user are displayed</li> </ol>	
<b>Postcondition:</b>	none
<b>Alternative flows:</b> none	

## CRC-Cards

<b>Class:</b> Stock	<b>Collaborators</b>
<hr/> <b>Responsibilities</b> - Blueprint of Stock	

<b>Class:</b> Portfolio	<b>Collaborators</b>  Share
<hr/> <b>Responsibilities</b> - Holds stocks	

<b>Class:</b> Display	<b>Collaborators</b>  UserDisplay StockOverviewDisplay StockDisplay Searchbar
<hr/> <b>Responsibilities</b> - UI	

<b>Class:</b> Filehandler	<b>Collaborators</b>
<hr/> <b>Responsibilities</b> - read and write to file	

<b>Class:</b> StockDisplay	<b>Collaborators</b>  Stock
<hr/> <b>Responsibilities</b> - Display a specific stock	

<b>Class:</b> StockOverviewDisplay	<b>Collaborators</b>
<hr/>	Stock
<b>Responsibilities</b>	Searchbar
- Displays a list of stocks	

<b>Class:</b> UserDisplay	<b>Collaborators</b>
<hr/>	User
<b>Responsibilities</b>	Portfolio
- Displays the user and his portfolio	

<b>Class:</b> Searchbar	<b>Collaborators</b>
<hr/>	Stock
<b>Responsibilities</b>	
- Option to search for a stock	

<b>Class:</b> User	<b>Collaborators</b>
<hr/>	Portfolio
<b>Responsibilities</b>	
- User credentials	

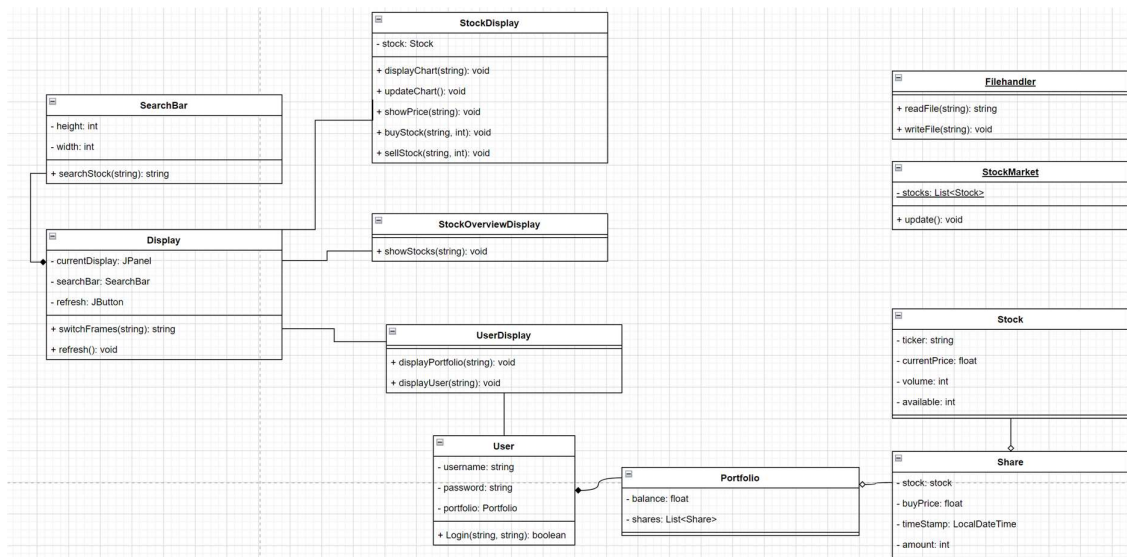


<b>Class:</b> Share	<b>Collaborators</b>
<b>Responsibilities</b> <ul style="list-style-type: none"> <li>- holds stock, buying price and amount of held stock</li> </ul>	Stock

<b>Class:</b> StockMarket	<b>Collaborators</b>
<b>Responsibilities</b> <ul style="list-style-type: none"> <li>- holds all stocks</li> <li>- updates stocks</li> </ul>	

<b>Class:</b> Notify	<b>Collaborators</b>
<b>Responsibilities</b> <ul style="list-style-type: none"> <li>- notifies the stockmarket to update the stocks</li> </ul>	

## Class Diagramm



Trading Simulation

Domain Model

