

# Entity Classes

User	
	Child: Admin
<ul style="list-style-type: none"><li>- Stores name, portfolio, id</li><li>- callTransaction()</li></ul>	<ul style="list-style-type: none"><li>- Portfolio</li><li>- Admin</li></ul>

Admin	
	Parent: User
<ul style="list-style-type: none"><li>- Regulate and ban users</li><li>- Control transaction flow</li></ul>	<ul style="list-style-type: none"><li>- Portfolio</li><li>- Admin</li></ul>

Transaction	
<ul style="list-style-type: none"><li>- stores sender and receiver</li><li>- stores id</li><li>- asset being traded</li><li>- stores voter name and total votes for and against</li></ul>	<ul style="list-style-type: none"><li>- User and Admin</li><li>- Asset</li><li>- Portfolio</li></ul>

Portfolio	
<ul style="list-style-type: none"><li>- Stores array of asset, historical values</li><li>- get set assets</li><li>- return total value</li></ul>	<ul style="list-style-type: none"><li>- Asset</li></ul>

Asset	
<ul style="list-style-type: none"><li>- Asset symbol</li><li>- price</li><li>- quantity</li><li>- update price</li><li>- calculate value</li></ul>	

# Use Case Classes

UserManager	
<ul style="list-style-type: none"><li>- Ban, promote, or accept user</li><li>- Calculate voting power of user</li></ul>	<ul style="list-style-type: none"><li>- User</li><li>- portfolioManager</li></ul>

TransactionManager	
<ul style="list-style-type: none"><li>- Control transactions</li><li>- Determine if transaction can be carried out directly or are votes required</li><li>- Store total vote for and against an asset.</li><li>- Decide the voting result whether the transaction passes.</li></ul>	<ul style="list-style-type: none"><li>- Transaction</li><li>- User</li></ul>

PortfolioManager	
<ul style="list-style-type: none"><li>- Update portfolio</li><li>- Determine the net worth</li><li>- Stores performance history</li></ul>	<ul style="list-style-type: none"><li>- Portfolio</li><li>- transactionManager</li></ul>

<I> DataAccessInterface	
<ul style="list-style-type: none"><li>- Load data from API (Used for dependency inversion)</li></ul>	

## Controller Classes

commandParser	
<ul style="list-style-type: none"><li>• Parses the input command and orchestrate managers</li></ul>	<ul style="list-style-type: none"><li>- UserManager</li><li>- portfolioManager</li><li>- TransactionManager</li><li>- assetManager</li></ul>

graphicsPresenter	
<ul style="list-style-type: none"><li>• Generate graph of portfolio</li></ul>	<ul style="list-style-type: none"><li>- portfolioManager</li></ul>

## Interface Classes

CLI	
<ul style="list-style-type: none"><li>Controls the system through command lines</li></ul>	CommandParser

GUI	
<ul style="list-style-type: none"><li>Displays graphs generated</li></ul>	GraphicsPresenter

YahooFinance	
<ul style="list-style-type: none"><li>Inputs real-time financial data into the system</li></ul>	DataAccessInterface