

<<Interface>> <b>ILoginViewModel</b>
+ UserName: string //text box for username input + IsLoggedIn: bool //true:false is user logged in + LoginCommand: ICommand //Button press for logging in + CreateUser: ICommand //Button press for create user
+ UserName(): string + IsLoggedIn(): bool + LoginCommand(): ICommand + CreateUser(): ICommand

<<Enumeration>> <b>MarketCapUnitsEnum</b>
+ Millions, + Billions

<<Interface>> <b>ISettings</b>
+ PriceMin: float //filter by dollar minimum + PriceMax: float //filter by dollar maximum + MarketCapMin: float //Market cap min + MarketCapMax: float //Market cap max + MarketCapEnumUnits: enum //Units for market cap to filter by + VolumeMin: float //Minimum volume shares for the current day in mill + VolumeMax: float //Maximum volume shares for the current day in mill
+ PriceMin(): float + PriceMax(): float + MarketCapMin(): float + MarketCapMax(): float + MarketCapEnumUnits(): enum + VolumeMin(): float + VolumeMax(): float

<<Interface>> <b>IStock</b>
+ Ticker: string //Ticker symbol + MarketCap: float //Market cap in mill + CurrentPrice: float //Current Price + LastClosePrice: //Last trading close price + CurrentVolume: float //Current volume of shares traded
+ Ticker(): string + MarketCap(): float + CurrentPrice(): float + LastClosePrice(): float + CurrentVolume(): float + UpdateFromStock(IStock, stock): float

<<Interface>> <b>IStockScreenerViewModel</b>
+ IUserInfoService: IUserInfoService + ObservableCollection<IStock>: FilteredStocks //property for the view to see a collection of stocks from filter + ObservableCollection<IStock>: WatchedStocks //property for the view to see a collection of stocks to be watched + ICommand: SaveFavorite //Button for loading saved favorite + ICommand: Apply //Button for applying settings
+ IUserInfoService: IUserInfoService()

+ ObservableCollection<IStock>: FilteredStocks()  
+ ObservableCollection<IStock>: WatchedStocks()  
+ ICommand: SaveFavorite()  
+ ICommand: Apply()

<<Interface>> IStockService: INotifyPropertyChanged	
+ List<IStock>: Stocks	//List of all stocks that are constantly updated
+ List<IStock>: Stocks()	

<<Interface>> IUser	
+ Name: string	//Name of the user
+ List<string>: WatchedStocks	//List of stocks to watch
+ ISettings: Settings	
//User settings, updated on apply	
+ Name(): string	
+ List<string>: WatchedStocks()	
+ ISettings: Settings()	

<<Interface>> IUserInfoService	
+ IUser: LoggedInUser	
+ List<IUser>: Users	
+ IUser: LoggedInUser()	
+ LogInUser: bool(string user): name= user	
+ CreateUser: bool(string user): name= user	
+ LogOffUser(void)	
+ List<IUser>: Users()	
+ LoadUsersFromFile: bool(string filePath): name= filePath	
+ SaveUsersToFile: bool(string filePath): name= filePath	