

Assessment One

1.0

Generated by Doxygen 1.7.2

Mon Oct 18 2010 00:00:28

Contents

1	Namespace Index	1
1.1	Package List	1
2	Class Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	Namespace Documentation	7
4.1	Package ApplicationLogic.Interfaces	7
4.2	Package ApplicationLogic.Model	7
4.3	Package ApplicationLogic.Presenter	8
4.4	Package Assessment_One	8
4.5	Package Assessment_One.Properties	8
4.6	Package NUnit_Tests.ApplicationLogic	9
5	Class Documentation	11
5.1	ApplicationLogic.Model.AppDataManager Class Reference	11
5.1.1	Detailed Description	12
5.1.2	Member Function Documentation	12
5.1.2.1	CreateNewBankAccount	12
5.1.2.2	CreateNewStockItem	12
5.1.2.3	DeleteBankAccount	12
5.1.2.4	DeleteStockItem	12
5.1.2.5	OrderItem	13
5.2	ApplicationLogic.Model.BankAccount Class Reference	13
5.2.1	Detailed Description	14
5.2.2	Constructor & Destructor Documentation	14
5.2.2.1	BankAccount	14
5.2.2.2	BankAccount	15
5.2.3	Member Function Documentation	15
5.2.3.1	CsvRepresentation	15
5.2.3.2	Deposit	15
5.2.3.3	ParseFromString	15
5.2.3.4	Transfer	16
5.2.3.5	Withdraw	16
5.2.4	Property Documentation	16
5.2.4.1	AccountNumber	16
5.2.4.2	Balance	16

5.2.4.3	Surname	17
5.3	NUnit.Tests.ApplicationLogic.BankAccountTest Class Reference	17
5.3.1	Detailed Description	17
5.4	ApplicationLogic.Presenter.CongregatePresenter Class Reference	17
5.4.1	Detailed Description	19
5.4.2	Constructor & Destructor Documentation	19
5.4.2.1	CongregatePresenter	19
5.4.3	Member Function Documentation	19
5.4.3.1	CloseApplication	19
5.4.3.2	CreateNewBankAccount	20
5.4.3.3	CreateNewStockItem	20
5.4.3.4	DeleteBankAccount	20
5.4.3.5	DeleteStockItem	20
5.4.3.6	Deposit	20
5.4.3.7	EditBankAccount	20
5.4.3.8	EditStockItem	20
5.4.3.9	LoadBankAccountsFromFile	21
5.4.3.10	LoadStockItemsFromFile	21
5.4.3.11	OrderItem	21
5.4.3.12	SaveBankAccountsToFile	21
5.4.3.13	SaveBankAccountsToFile	21
5.4.3.14	SaveStockItemsToFile	21
5.4.3.15	SaveStockItemsToFile	22
5.4.3.16	SetUpBankAccountsFilePath	22
5.4.3.17	SetUpStockItemFilePath	22
5.4.3.18	Withdraw	22
5.5	ApplicationLogic.Model.ErrorMessage Class Reference	23
5.5.1	Detailed Description	23
5.6	ApplicationLogic.Model.ErrorMessageCollection Class Reference	23
5.6.1	Detailed Description	23
5.7	ApplicationLogic.FileHandler< T > Class Template Reference	23
5.7.1	Detailed Description	24
5.7.2	Member Function Documentation	24
5.7.2.1	LoadFromFile	24
5.7.2.2	SaveToFile	25
5.8	ApplicationLogic.Interfaces.IBankAccountView Interface Reference	25
5.8.1	Detailed Description	25
5.9	ApplicationLogic.Interfaces.ICongregateView Interface Reference	25
5.9.1	Detailed Description	26
5.10	ApplicationLogic.Interfaces.ICSVSerializable< T > Interface Template Reference	26
5.10.1	Detailed Description	26
5.11	ApplicationLogic.Interfaces.IStockItemView Interface Reference	27
5.11.1	Detailed Description	27
5.12	ApplicationLogic.Interfaces.IViewModel Interface Reference	27
5.12.1	Detailed Description	27
5.13	Assessment_One.MainWindow Class Reference	28
5.13.1	Detailed Description	28
5.13.2	Member Function Documentation	28
5.13.2.1	Dispose	28

5.14	ApplicationLogic.Model.NoFilePathSetException Class Reference	29
5.14.1	Detailed Description	29
5.15	ApplicationLogic.Model.NotEnoughFundsException Class Reference . .	29
5.15.1	Detailed Description	29
5.16	Assessment_One.Properties.Resources Class Reference	30
5.16.1	Detailed Description	30
5.17	Assessment_One.Properties.Settings Class Reference	30
5.17.1	Detailed Description	30
5.18	Assessment_One.SettingsWindow Class Reference	30
5.18.1	Detailed Description	30
5.18.2	Member Function Documentation	31
5.18.2.1	Dispose	31
5.19	ApplicationLogic.Model.StockItem Class Reference	31
5.19.1	Detailed Description	32
5.19.2	Member Function Documentation	32
5.19.2.1	CsvRepresentation	32
5.19.2.2	IsValidStockCode	32
5.19.2.3	ParseFromString	33
5.19.2.4	Validate	33
5.20	NUnit.Tests.StockItemTest Class Reference	34
5.20.1	Detailed Description	34

Chapter 1

Namespace Index

1.1 Package List

Here are the packages with brief descriptions (if available):

ApplicationLogic.Interfaces	7
ApplicationLogic.Model	7
ApplicationLogic.Presenter	8
Assessment_One	8
Assessment_One.Properties	8
NUnit.Tests.ApplicationLogic	9

Chapter 2

Class Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

NUnit.Tests.ApplicationLogic.BankAccountTest	17
ApplicationLogic.Presenter.CongregatePresenter	17
ApplicationLogic.Model.ErrorMessage	23
ApplicationLogic.Model.ErrorMessageCollection	23
ApplicationLogic.FileHandler< T >	23
ApplicationLogic.Interfaces.IBankAccountView	25
ApplicationLogic.Interfaces.ICongregateView	25
ApplicationLogic.Interfaces.ICSVSerializable< T >	26
ApplicationLogic.Model.BankAccount	13
ApplicationLogic.Model.StockItem	31
ApplicationLogic.Interfaces.IStockItemView	27
ApplicationLogic.Interfaces.IViewModel	27
ApplicationLogic.Model.AppDataManager	11
Assessment_One.MainWindow	28
ApplicationLogic.Model.NoFilePathSetException	29
ApplicationLogic.Model.NotEnoughFundsException	29
Assessment_One.Properties.Resources	30
Assessment_One.Properties.Settings	30
Assessment_One.SettingsWindow	30
NUnit.Tests.StockItemTest	34

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

ApplicationLogic.Model.AppDataManager (Handles data and persistent) . . .	11
ApplicationLogic.Model.BankAccount (Pseudo bank account to allow the placement of an order)	13
NUnit.Tests.ApplicationLogic.BankAccountTest	17
ApplicationLogic.Presenter.CongregatePresenter (Presenter for the MainWindow: handles events and GUI-related part of the application logic)	17
ApplicationLogic.Model.ErrorMessage	23
ApplicationLogic.Model.ErrorMessageCollection (Used to store meaningful error messages for further use)	23
ApplicationLogic.FileHandler< T > (Handles reading and writing from files) .	23
ApplicationLogic.Interfaces.IBankAccountView (Utilized by the presenter to get the necessary values from a view)	25
ApplicationLogic.Interfaces.ICongregateView (Utilized by the presenter to get the necessary values from a view)	25
ApplicationLogic.Interfaces.ICSVSerializable< T > (Used to ensure all objects will be able to persisted via FileHandler class)	26
ApplicationLogic.Interfaces.IStockItemView (Utilized by the presenter to get the necessary values from a view)	27
ApplicationLogic.Interfaces.IViewModel (Utilized by the View to set up the data binding to the lists in the model)	27
Assessment_One.MainWindow	28
ApplicationLogic.Model.NoFilePathSetException	29
ApplicationLogic.Model.NotEnoughFundsException (Description of NotEnoughFundsException)	29
Assessment_One.Properties.Resources (Eine stark typisierte Ressourcenklasse zum Suchen von lokalisierten Zeichenfolgen usw)	30
Assessment_One.Properties.Settings	30
Assessment_One.SettingsWindow	30

ApplicationLogic.Model.StockItem (Stores all necessary data for a StockItem)	31
NUnit.Tests.StockItemTest	34

Chapter 4

Namespace Documentation

4.1 Package ApplicationLogic.Interfaces

Classes

- interface [IBankAccountView](#)
Utilized by the presenter to get the necessary values from a view.
- interface [ICongregateView](#)
Utilized by the presenter to get the necessary values from a view.
- interface [ICSVSerializable](#)
Used to ensure all objects will be able to persited via [FileHandler](#) class.
- interface [IStockItemView](#)
Utilized by the presenter to get the necessary values from a view.
- interface [IViewModel](#)
Utilized by the View to set up the data binding to the lists in the model.

4.2 Package ApplicationLogic.Model

Classes

- class [AppDataManager](#)
Handles data and persistent.
- class [BankAccount](#)
Pseudo bank account to allow the placement of an order.

- class [ErrorMessage](#)
- class [ErrorMessageCollection](#)
Used to store meaningful error messages for further use.
- class [NoFilePathSetException](#)
- class [NotEnoughFundsException](#)
Description of [NotEnoughFundsException](#).
- class [StockItem](#)
Stores all necessary data for a [StockItem](#).

4.3 Package ApplicationLogic.Presenter

Classes

- class [CongregatePresenter](#)
[Presenter](#) for the MainWindow: handles events and GUI-related part of the application logic.

4.4 Package Assessment_One

Packages

- package [Properties](#)

Classes

- class [MainWindow](#)
- class [Program](#)
- class [SettingsWindow](#)

4.5 Package Assessment_One.Properties

Classes

- class [Resources](#)
Eine stark typisierte Ressourcenklasse zum Suchen von lokalisierten Zeichenfolgen usw.
- class [Settings](#)

4.6 Package NUnit_Tests.ApplicationLogic

Classes

- class [BankAccountTest](#)

Chapter 5

Class Documentation

5.1 ApplicationLogic.Model.AppDataManager Class Reference

Handles data and persistent.

Inheritance diagram for ApplicationLogic.Model.AppDataManager:

Public Member Functions

- void [CreateNewStockItem](#) ()
Creates a new [StockItem](#) and initializes it with dummy values. Adds the item to the [StockItem](#) collection.
- void [DeleteStockItem](#) ([StockItem](#) si)
Deletes a [StockItem](#) from the [StockItem](#) collection. Throws an [ArgumentException](#) if the item can not be found.
- void [CreateNewBankAccount](#) ()
Creates a new [BankAccount](#) and initializes it with dummy values. Adds the item to the [BankAccount](#) collection.
- void [DeleteBankAccount](#) ([BankAccount](#) ba)
Deletes a [BankAccount](#) from the [BankAccount](#) collection. Throws an [ArgumentException](#) if the account can not be found.
- void [OrderItem](#) ([StockItem](#) si, [BankAccount](#) ba, int quantity)
Attempts to order an item. If no quantity was provided the required quantity will be ordered. Otherwise the provided quantity will be ordered.

Properties

- BindingList< [StockItem](#) > **StockItems** [get, set]
- BindingList< [BankAccount](#) > **BankAccounts** [get, set]
- FileHandler< [StockItem](#) > **StockItemHandler** [get, set]
- FileHandler< [BankAccount](#) > **BankAccountHandler** [get, set]

5.1.1 Detailed Description

Handles data and persistent.

Definition at line 13 of file AppDataManager.cs.

5.1.2 Member Function Documentation

5.1.2.1 void ApplicationLogic.Model.AppDataManager.CreateNewBankAccount ()

Creates a new [BankAccount](#) and initializes it with dummy values. Adds the item to the [BankAccount](#) collection.

Definition at line 70 of file AppDataManager.cs.

5.1.2.2 void ApplicationLogic.Model.AppDataManager.CreateNewStockItem ()

Creates a new [StockItem](#) and initializes it with dummy values. Adds the item to the [StockItem](#) collection.

Definition at line 44 of file AppDataManager.cs.

5.1.2.3 void ApplicationLogic.Model.AppDataManager.DeleteBankAccount (BankAccount ba)

Deletes a [BankAccount](#) from the [BankAccount](#) collection. Throws an ArgumentException if the account can not be found.

Definition at line 80 of file AppDataManager.cs.

5.1.2.4 void ApplicationLogic.Model.AppDataManager.DeleteStockItem (StockItem si)

Deletes a [StockItem](#) from the [StockItem](#) collection. Throws an ArgumentException if the item can not be found.

Definition at line 54 of file AppDataManager.cs.

5.1.2.5 void ApplicationLogic.Model.AppDataManager.OrderItem (StockItem si, BankAccount ba, int quantity)

Attempts to order an item. If no quantity was provided the required quantity will be ordered. Otherwise the provided quantity will be ordered.

Parameters

<i>indexStock-Item</i>	Index of the stock item in the stock items list.
<i>indexBankAccount</i>	Index of the bank account in the bank account list.
<i>quantity</i>	The quantity to be ordered. 0 orders the required quantity.

Definition at line 202 of file AppDataManager.cs.

The documentation for this class was generated from the following file:

- Application_Logic/Model/AppDataManager.cs

5.2 ApplicationLogic.Model.BankAccount Class Reference

Pseudo bank account to allow the placement of an order.

Inheritance diagram for ApplicationLogic.Model.BankAccount:

Public Member Functions

- [BankAccount](#) ()
Initializes a new instance of the [BankAccount](#) class.
- [BankAccount](#) (int acc, string name, double balance)
Initializes a new instance of the [BankAccount](#) class.
- void [Withdraw](#) (double amount)
Allows the withdrawal of money from this account. Credit is not granted.
- void [Deposit](#) (double amount)
Allows the deposit of money to this account.
- void [Transfer](#) (int accountNumber, double amount)
Amount to be transfered to another account. NOTE: This method is a fake to simulate "real" banking. The money will not be transfered to any account.

- string [CsvRepresentation](#) ()
Returns the current [BankAccount](#) object as a CSV-String.
- [BankAccount ParseFromString](#) (string stringRepresentation)
Attempts to create a [BankAccount](#) object from a string.

Static Public Attributes

- static [ErrorMessageCollection](#) **ErrorMessages** = new [ErrorMessageCollection](#)()

Properties

- int [AccountNumber](#) [get, set]
Gets or sets the account number.
- String [Surname](#) [get, set]
Gets or sets the surname.
- double [Balance](#) [get, set]
Gets or sets the balance.

Events

- PropertyChangedEventHandler **PropertyChanged**

5.2.1 Detailed Description

Pseudo bank account to allow the placement of an order.

Definition at line 13 of file BankAccount.cs.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 ApplicationLogic.Model.BankAccount.BankAccount ()

Initializes a new instance of the [BankAccount](#) class.

Definition at line 82 of file BankAccount.cs.

5.2.2.2 ApplicationLogic.Model.BankAccount.BankAccount (int *acc*, string *name*, double *balance*)

Initializes a new instance of the [BankAccount](#) class.

Parameters

<i>acc</i>	The acc.
<i>name</i>	The name.
<i>balance</i>	The balance.

Definition at line 92 of file BankAccount.cs.

5.2.3 Member Function Documentation

5.2.3.1 string ApplicationLogic.Model.BankAccount.CsvRepresentation ()

Returns the current [BankAccount](#) object as a CSV-String.

Returns

Representation of the current object as CSV-String.

Implements [ApplicationLogic.Interfaces.ICSVSerializable< T >](#).

Definition at line 204 of file BankAccount.cs.

5.2.3.2 void ApplicationLogic.Model.BankAccount.Deposit (double *amount*)

Allows the deposit of money to this account.

Parameters

<i>amount</i>	Amount to be deposited - must be greater than 0.
---------------	--

Definition at line 127 of file BankAccount.cs.

5.2.3.3 BankAccount ApplicationLogic.Model.BankAccount.ParseFromString (string *stringRepresentation*)

Attempts to create a [BankAccount](#) object from a string.

Parameters

<i>stringRepresentation</i>	The String to be parsed to bank account.
-----------------------------	--

Returns

[BankAccount](#) object.

Definition at line 214 of file BankAccount.cs.

5.2.3.4 void ApplicationLogic.Model.BankAccount.Transfer (int *accountNumber*, double *amount*)

Amount to be transfered to another account. NOTE: This method is a fake to simulate "real" banking. The money will not be transfered to any account.

Parameters

<i>amount</i>	Amount to be transfered - must be greater than 0.
<i>account-Number</i>	Account number to transfer the money to.

Definition at line 145 of file BankAccount.cs.

5.2.3.5 void ApplicationLogic.Model.BankAccount.Withdraw (double *amount*)

Allows the withdrawal of money from this account. Credit is not granted.

Parameters

<i>amount</i>	Amount to be withdrawn - must be greater than 0.
---------------	--

Definition at line 104 of file BankAccount.cs.

5.2.4 Property Documentation**5.2.4.1 int ApplicationLogic.Model.BankAccount.AccountNumber [get, set]**

Gets or sets the account number.

The account number.

Definition at line 21 of file BankAccount.cs.

5.2.4.2 double ApplicationLogic.Model.BankAccount.Balance [get, set]

Gets or sets the balance.

The balance.

Definition at line 57 of file BankAccount.cs.

5.2.4.3 String ApplicationLogic.Model.BankAccount.Surname [get, set]

Gets or sets the surname.

The surname.

Definition at line 39 of file BankAccount.cs.

The documentation for this class was generated from the following file:

- Application_Logic/Model/BankAccount.cs

5.3 NUnit_Tests.ApplicationLogic.BankAccountTest Class Reference

Public Member Functions

- void **SetUp** ()
- void **TestBalanceUnder0** ()
- void **TestWithdrawalWithTooHighValues** ()
- void **TestWithdrawalWithTooSmallValues** ()
- void **TestDepositWithTooSmallValue** ()
- void **TestTransferWithTooSmallValue** ()
- void **TestOrdinaryFunctions** ()

5.3.1 Detailed Description

Definition at line 11 of file BankAccountTest.cs.

The documentation for this class was generated from the following file:

- NUnit_Tests/ApplicationLogic/BankAccountTest.cs

5.4 ApplicationLogic.Presenter.CongregatePresenter Class Reference

[Presenter](#) for the MainWindow: handles events and GUI-related part of the application logic.

Public Member Functions

- [CongregatePresenter](#) ([ICongregateView](#) view, [IStockItemView](#) stockItemView, [IBankAccountView](#) bankAccountView, [IViewModel](#) model)
Initializes a new instance of the [CongregatePresenter](#) class.
- void [CreateNewStockItem](#) ()

ApplicationLogic.Model.AppDataManagerClass

- void [DeleteStockItem](#) ()
Deletes the stock item.
- void [CreateNewBankAccount](#) ()
Creates the new bank account.
- void [DeleteBankAccount](#) ()
Deletes the bank account.
- void [EditStockItem](#) ()
Edits the stock item.
- void [EditBankAccount](#) ()
Edits the bank account.
- void [OrderItem](#) ()
Orders the item.
- void [Deposit](#) ()
Deposits this instance.
- void [Withdraw](#) ()
Withdraws this instance.
- void [CloseApplication](#) ()
Closes the application.
- void [LoadStockItemsFromFile](#) (String filePath)
Loads the stock items from file.
- void [LoadBankAccountsFromFile](#) (String filePath)
Loads the bank accounts from file.
- void [SaveStockItemsToFile](#) (String filePath)
Saves the stock items to file.
- void [SaveBankAccountsToFile](#) (String filePath)
Saves the bank accounts to file.
- void [SetUpStockItemFilePath](#) (string filePathStockItems)
Sets up stock item file path.
- void [SetUpBankAccountsFilePath](#) (string filePathBankAccounts)
Sets up bank accounts file path.

- void [SaveBankAccountsToFile](#) ()
Saves the bank accounts to file.
- void [SaveStockItemsToFile](#) ()
Saves the stock items to file.

Public Attributes

- [ICongregateView](#) _View
- [IStockItemView](#) _StockItemView
- [IBankAccountView](#) _BankAccountView
- [AppDataManager](#) _Model

5.4.1 Detailed Description

[Presenter](#) for the MainWindow: handles events and GUI-related part of the application logic.

Definition at line 14 of file CongregatePresenter.cs.

5.4.2 Constructor & Destructor Documentation

- 5.4.2.1 [ApplicationLogic.Presenter.CongregatePresenter.CongregatePresenter](#) (
[ICongregateView](#) *view*, [IStockItemView](#) *stockItemView*, [IBankAccountView](#) *bankAccountView*, [IViewModel](#) *model*)

Initializes a new instance of the [CongregatePresenter](#) class.

Parameters

<i>view</i>	The view.
<i>stock-ItemView</i>	The stock item view.
<i>bankAccountView</i>	The bank account view.
<i>model</i>	The model.

Definition at line 29 of file CongregatePresenter.cs.

5.4.3 Member Function Documentation

- 5.4.3.1 void [ApplicationLogic.Presenter.CongregatePresenter.CloseApplication](#) ()

Closes the application.

Definition at line 207 of file CongregatePresenter.cs.

5.4.3.2 void ApplicationLogic.Presenter.CongregatePresenter.CreateNewBankAccount ()

Creates the new bank account.

Definition at line 60 of file CongregatePresenter.cs.

5.4.3.3 void ApplicationLogic.Presenter.CongregatePresenter.CreateNewStockItem ()

ApplicationLogic.Model.AppDataManagerClass

Definition at line 40 of file CongregatePresenter.cs.

5.4.3.4 void ApplicationLogic.Presenter.CongregatePresenter.DeleteBankAccount ()

Deletes the bank account.

Definition at line 68 of file CongregatePresenter.cs.

5.4.3.5 void ApplicationLogic.Presenter.CongregatePresenter.DeleteStockItem ()

Deletes the stock item.

Definition at line 48 of file CongregatePresenter.cs.

5.4.3.6 void ApplicationLogic.Presenter.CongregatePresenter.Deposit ()

Deposits this instance.

Definition at line 162 of file CongregatePresenter.cs.

5.4.3.7 void ApplicationLogic.Presenter.CongregatePresenter.EditBankAccount ()

Edits the bank account.

Definition at line 111 of file CongregatePresenter.cs.

5.4.3.8 void ApplicationLogic.Presenter.CongregatePresenter.EditStockItem ()

Edits the stock item.

Definition at line 79 of file CongregatePresenter.cs.

5.4.3.9 void ApplicationLogic.Presenter.CongregatePresenter.LoadBankAccountsFromFile (String filePath)

Loads the bank accounts from file.

Parameters

<i>filePath</i>	The file path.
-----------------	----------------

Definition at line 226 of file CongregatePresenter.cs.

5.4.3.10 void ApplicationLogic.Presenter.CongregatePresenter.LoadStockItemsFromFile (String filePath)

Loads the stock items from file.

Parameters

<i>filePath</i>	The file path.
-----------------	----------------

Definition at line 217 of file CongregatePresenter.cs.

5.4.3.11 void ApplicationLogic.Presenter.CongregatePresenter.OrderItem ()

Orders the item.

Definition at line 139 of file CongregatePresenter.cs.

5.4.3.12 void ApplicationLogic.Presenter.CongregatePresenter.SaveBankAccountsToFile (String filePath)

Saves the bank accounts to file.

Parameters

<i>filePath</i>	The file path.
-----------------	----------------

Definition at line 244 of file CongregatePresenter.cs.

5.4.3.13 void ApplicationLogic.Presenter.CongregatePresenter.SaveBankAccountsToFile ()

Saves the bank accounts to file.

Definition at line 276 of file CongregatePresenter.cs.

5.4.3.14 void ApplicationLogic.Presenter.CongregatePresenter.SaveStockItemsToFile ()

Saves the stock items to file.

Definition at line 291 of file CongregatePresenter.cs.

5.4.3.15 void ApplicationLogic.Presenter.CongregatePresenter.SaveStockItemsToFile (String *filePath*)

Saves the stock items to file.

Parameters

<i>filePath</i>	The file path.
-----------------	----------------

Definition at line 235 of file CongregatePresenter.cs.

5.4.3.16 void ApplicationLogic.Presenter.CongregatePresenter.SetupBankAccountsFilePath (string *filePathBankAccounts*)

Sets up bank accounts file path.

Parameters

<i>filePath-BankAccounts</i>	The file path bank accounts.
------------------------------	------------------------------

Definition at line 265 of file CongregatePresenter.cs.

5.4.3.17 void ApplicationLogic.Presenter.CongregatePresenter.SetupStockItemFilePath (string *filePathStockItems*)

Sets up stock item file path.

Parameters

<i>filePath-StockItems</i>	The file path stock items.
----------------------------	----------------------------

Definition at line 253 of file CongregatePresenter.cs.

5.4.3.18 void ApplicationLogic.Presenter.CongregatePresenter.Withdraw ()

Withdraws this instance.

Definition at line 179 of file CongregatePresenter.cs.

The documentation for this class was generated from the following file:

- Application_Logic/Presenter/CongregatePresenter.cs

5.5 ApplicationLogic.Model.ErrorMessage Class Reference

Public Member Functions

- **ErrorMessage** (string p)
- override string **ToString** ()

Properties

- String **Message** [get, set]
- String **Source** [get, set]

5.5.1 Detailed Description

Definition at line 8 of file ErrorMessage.cs.

The documentation for this class was generated from the following file:

- Application_Logic/Model/ErrorMessage.cs

5.6 ApplicationLogic.Model.ErrorMessageCollection Class Reference

Used to store meaningful errormessages for further use.

Public Member Functions

- override string **ToString** ()

5.6.1 Detailed Description

Used to store meaningful errormessages for further use.

Definition at line 11 of file ErrorMessageCollection.cs.

The documentation for this class was generated from the following file:

- Application_Logic/Model/ErrorMessageCollection.cs

5.7 ApplicationLogic.FileHandler< T > Class Template Reference

Handles reading and writing from files.

Public Member Functions

- **FileHandler** ()
- **FileHandler** (String readFilePath, String writeFilePath)
- void **SaveToFile** (IList< T > elements)

Attempts to write the specified collection to the file specified in the Handler. Throws NoFilePathSetException if no file has yet been set.

- IList< T > **LoadFromFile** (T item)

Attempts to read a collection of items from the specified file. Throws NoFilePathSetException if no file has yet been set.

Public Attributes

- String **ReadFilePath**
- String **WriteFilePath**

5.7.1 Detailed Description

`template<T> class ApplicationLogic::FileHandler< T >`

Handles reading and writing from files.

Type Constraints

T : ICSVSerializable<T>

Definition at line 13 of file FileHandler.cs.

5.7.2 Member Function Documentation

5.7.2.1 `template<T> IList<T> ApplicationLogic.FileHandler< T >.LoadFromFile (T item)`

Attempts to read a collection of items from the specified file. Throws NoFilePathSetException if no file has yet been set.

Parameters

<i>item</i>	Parameter needed to construct the objects.
-------------	--

Returns

Collection of items.

Definition at line 61 of file FileHandler.cs.

5.7.2.2 `template<T> void ApplicationLogic.FileHandler< T >.SaveToFile (IList< T > elements)`

Attempts to write the specified collection to the file specified in the Handler. Throws `NoFilePathSetException` if no file has yet been set.

Parameters

<i>elements</i>	Collection to be saved to file.
-----------------	---------------------------------

Definition at line 34 of file `FileHandler.cs`.

The documentation for this class was generated from the following file:

- `Application_Logic/Model/FileHandler.cs`

5.8 ApplicationLogic.Interfaces.IBankAccountView Interface Reference

Utilized by the presenter to get the necessary values from a view.

Properties

- `int AccountNumber` [get]
- `String Surname` [get]
- `double Balance` [get]

5.8.1 Detailed Description

Utilized by the presenter to get the necessary values from a view.

Definition at line 11 of file `IBankAccountView.cs`.

The documentation for this interface was generated from the following file:

- `Application_Logic/Interfaces/IBankAccountView.cs`

5.9 ApplicationLogic.Interfaces.ICongregateView Interface Reference

Utilized by the presenter to get the necessary values from a view.

Public Member Functions

- `bool ConfirmDelete ()`
- `bool ConfirmClose ()`
- `void DisplayValidationErrors (ErrorMessageCollection errorCollection)`

Properties

- [StockItem](#) **StockItem** [get]
- [BankAccount](#) **BankAccount** [get]
- int **Quantity** [get]
- double **Deposit** [get]
- double **Withdraw** [get]

5.9.1 Detailed Description

Utilized by the presenter to get the necessary values from a view.

Definition at line 12 of file ICongregateView.cs.

The documentation for this interface was generated from the following file:

- Application_Logic/Interfaces/ICongregateView.cs

5.10 ApplicationLogic.Interfaces.ICSVSerializable< T > Interface Template Reference

Used to ensure all objects will be able to persited via [FileHandler](#) class.

Inheritance diagram for ApplicationLogic.Interfaces.ICSVSerializable< T >:

Public Member Functions

- String **CsvRepresentation** ()
- T **ParseFromString** (String stringRepresentation)

5.10.1 Detailed Description

template<T> interface ApplicationLogic::Interfaces::ICSVSerializable< T >

Used to ensure all objects will be able to persited via [FileHandler](#) class.

Template Parameters

<i>T</i>

Definition at line 12 of file ICSVSerializable.cs.

The documentation for this interface was generated from the following file:

- Application_Logic/Interfaces/ICSVSerializable.cs

5.11 ApplicationLogic.Interfaces.IStockItemView Interface Reference

Utilized by the presenter to get the necessary values from a view.

Properties

- int **CurrentStock** [get]
- string **ItemName** [get]
- int **RequiredStock** [get]
- string **StockCode** [get]
- string **SupplierName** [get]
- double **UnitCost** [get]

5.11.1 Detailed Description

Utilized by the presenter to get the necessary values from a view.

Definition at line 7 of file IStockItemView.cs.

The documentation for this interface was generated from the following file:

- Application_Logic/Interfaces/IStockItemView.cs

5.12 ApplicationLogic.Interfaces.IViewModel Interface Reference

Utilized by the View to set up the data binding to the lists in the model.

Inheritance diagram for ApplicationLogic.Interfaces.IViewModel:

Properties

- BindingList< [StockItem](#) > **StockItems** [get]
- BindingList< [BankAccount](#) > **BankAccounts** [get]

5.12.1 Detailed Description

Utilized by the View to set up the data binding to the lists in the model.

Definition at line 10 of file IViewModel.cs.

The documentation for this interface was generated from the following file:

- Application_Logic/Interfaces/IViewModel.cs

5.13 Assessment_One.MainWindow Class Reference

Public Member Functions

- void **DisplayValidationErrors** ([ErrorMessageCollection](#) errorCollection)
- bool **ConfirmDelete** ()
- bool **ConfirmClose** ()

Protected Member Functions

- override void [Dispose](#) (bool disposing)
Verwendete Ressourcen bereinigen.

Properties

- int **CurrentStock** [get]
- int **RequiredStock** [get]
- string **StockCode** [get]
- string **SupplierName** [get]
- double **UnitCost** [get]
- string **ItemName** [get]
- int **AccountNumber** [get]
- string **Surname** [get]
- double **Balance** [get]
- [StockItem](#) **StockItem** [get]
- [BankAccount](#) **BankAccount** [get]
- int **Quantity** [get]
- double **Deposit** [get]
- double **Withdraw** [get]

5.13.1 Detailed Description

Definition at line 18 of file MainWindow.cs.

5.13.2 Member Function Documentation

5.13.2.1 override void Assessment_One.MainWindow.Dispose (bool *disposing*) [protected]

Verwendete Ressourcen bereinigen.

Parameters

<i>disposing</i>	True, wenn verwaltete Ressourcen gelöscht werden sollen; andernfalls False.
------------------	---

Definition at line 14 of file MainWindow.Designer.cs.

The documentation for this class was generated from the following files:

- Assessment_One/MainWindow.cs
- Assessment_One/MainWindow.Designer.cs

5.14 ApplicationLogic.Model.NoFilePathSetException Class Reference

Public Member Functions

- **NoFilePathSetException** (String msg)

5.14.1 Detailed Description

Definition at line 8 of file NoFilePathSetException.cs.

The documentation for this class was generated from the following file:

- Application_Logic/Model/NoFilePathSetException.cs

5.15 ApplicationLogic.Model.NotEnoughFundsException Class Reference

Description of [NotEnoughFundsException](#).

Public Member Functions

- **NotEnoughFundsException** (String message)

5.15.1 Detailed Description

Description of [NotEnoughFundsException](#).

Definition at line 8 of file NotEnoughFundsException.cs.

The documentation for this class was generated from the following file:

- Application_Logic/Model/NotEnoughFundsException.cs

5.16 Assessment_One.Properties.Resources Class Reference

Eine stark typisierte Ressourcenklasse zum Suchen von lokalisierten Zeichenfolgen usw.

5.16.1 Detailed Description

Eine stark typisierte Ressourcenklasse zum Suchen von lokalisierten Zeichenfolgen usw.

Definition at line 25 of file Resources.Designer.cs.

The documentation for this class was generated from the following file:

- Assessment_One/Properties/Resources.Designer.cs

5.17 Assessment_One.Properties.Settings Class Reference

Properties

- static [Settings](#) **Default** [get]
- string **StockItemFilePath** [get, set]
- string **BankAccountFilePath** [get, set]

5.17.1 Detailed Description

Definition at line 16 of file Settings.Designer.cs.

The documentation for this class was generated from the following files:

- Assessment_One/Properties/Settings.Designer.cs
- Assessment_One/Settings.cs

5.18 Assessment_One.SettingsWindow Class Reference

Protected Member Functions

- override void [Dispose](#) (bool disposing)
Clean up any resources being used.

5.18.1 Detailed Description

Definition at line 15 of file SettingsWindow.cs.

5.18.2 Member Function Documentation

5.18.2.1 `override void Assessment_One.SettingsWindow.Dispose (bool disposing)`
[protected]

Clean up any resources being used.

Parameters

<i>disposing</i>	true if managed resources should be disposed; otherwise, false.
------------------	---

Definition at line 14 of file SettingsWindow.Designer.cs.

The documentation for this class was generated from the following files:

- Assessment_One/SettingsWindow.cs
- Assessment_One/SettingsWindow.Designer.cs

5.19 ApplicationLogic.Model.StockItem Class Reference

Stores all necessary data for a [StockItem](#).

Inheritance diagram for ApplicationLogic.Model.StockItem:

Public Member Functions

- **StockItem** (String stockCode, String name, String supplierName, double unitCost, int required, int currentStock)
- void **EditStockItem** (String stockCode, String name, String supplierName, double unitCost, int required, int currentStock)
- String **CsvRepresentation** ()
Returns the current [StockItem](#) object as a CSV-String.
- **StockItem ParseFromString** (string stringRepresentation)
Attempts to create a [StockItem](#) object from a string.

Static Public Member Functions

- static bool **IsValidStockCode** (string value)
Checks if the stock code conforms to a certain format. Current format is exactly four numbers, with leading 0 allowed.
- static bool **Validate** (String stockCode, String name, String supplierName, double unitCost, int required, int currentStock)

Validates a set of possible changes to a [StockItem](#).

Static Public Attributes

- static [ErrorMessageCollection](#) **ErrorMessages** = new [ErrorMessageCollection](#)()

Properties

- String **StockCode** [get, set]
- String **Name** [get, set]
- String **SupplierName** [get, set]
- double **UnitCost** [get, set]
- int **RequiredStock** [get, set]
- int **CurrentStock** [get, set]

Events

- PropertyChangedEventHandler **PropertyChanged**

5.19.1 Detailed Description

Stores all necessary data for a [StockItem](#).

Definition at line 14 of file StockItem.cs.

5.19.2 Member Function Documentation

5.19.2.1 String ApplicationLogic.Model.StockItem.CsvRepresentation ()

Returns the current [StockItem](#) object as a CSV-String.

Returns

Representation of the current object as CSV-String.

Implements [ApplicationLogic.Interfaces.ICSVSerializable< T >](#).

Definition at line 230 of file StockItem.cs.

5.19.2.2 static bool ApplicationLogic.Model.StockItem.IsValidStockCode (string value) [static]

Checks if the stock code conforms to a certain format. Current format is exactly four numbers, with leading 0 allowed.

Parameters

<i>value</i>	The string that must be checked against the schema.
--------------	---

Returns

True if the string conforms to the schema, false otherwise.

Definition at line 152 of file StockItem.cs.

5.19.2.3 StockItem ApplicationLogic.Model.StockItem.ParseFromString (string stringRepresentation)

Attempts to create a [StockItem](#) object from a string.

Parameters

<i>stringRepresentation</i>	The String to be parsed to StockItem .
-----------------------------	--

Returns

[StockItem](#) object.

Definition at line 240 of file StockItem.cs.

**5.19.2.4 static bool ApplicationLogic.Model.StockItem.Validate (String stockCode, String name, String supplierName, double unitCost, int required, int currentStock)
[static]**

Validates a set of possible changes to a [StockItem](#).

Parameters

<i>stockCode</i>	StockCodew to be verified
<i>name</i>	Name to be verified
<i>supplier-Name</i>	SupplierName to be verified
<i>unitCost</i>	UnitCost to be verified
<i>required</i>	RequiredStock to be verified
<i>currentStock</i>	CurrentStock to be verified

Returns

True if the values would be valid, false otherwise.

Definition at line 187 of file StockItem.cs.

The documentation for this class was generated from the following file:

- Application_Logic/Model/StockItem.cs

5.20 NUnit.Tests.StockItemTest Class Reference

Public Member Functions

- void **SetUp** ()
- void **TestLessThan0Cost** ()
- void **TestLessThan0CurrentStock** ()
- void **TestLessThan0RequiredStock** ()
- void **TestIsValidStockCode** ()
- void **TestIsValidStockCodeRaiseException** ()

5.20.1 Detailed Description

Definition at line 11 of file StockItemTest.cs.

The documentation for this class was generated from the following file:

- NUnit_Tests/ApplicationLogic/StockItemTest.cs