IT UNIVERSITY OF COPENHAGEN

BDSA 2014

Calendar System Documentation

ASSIGNMENT 39

Anders Wind Steffensen - awis@itu.dk
Christopher Blundell - cnbl@itu.dk
Pierre Mandas - ppma@itu.dk

Generated by Doxygen 1.8.8

Fri Sep 19 2014 22:00

Contents

Chapter 1

Namespace Index

1.1 Packages

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

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

Namespace Documentation

4.1 Package CalendarSystem

Namespaces

- package Controller
- package DataStorage
- package Model
- package View

Classes

- class **Startup**

4.2 Package CalendarSystem.Controller

Classes

- class InputController

  The inputcontroller handles all incoming user interaction, such as button presses and so on.
- class NotificationController

  The Notification controller has a timer for the notifications and will create a popup when a notification's time is exceeded.
- class ViewController

  The view controller handles all calls and creations of the view. It implements the IObserver interface and therefore can get notified when changes in the model happens.

4.3 Package CalendarSystem.DataStorage

Classes

- class DatabaseStorage

  A storage class which implements the IStorage interface. The class is meant to have a connection to a database where events will be added when they are created and put in the local Calendar class.
- class FakeStorage

A storage class which implements the IStorage interface. The class is a fake storage in that sense that the fields and methods return components that are created at runtime. Therefore no events will be accessible from between runs of the system.

- interface IStorage

  An interface for a storage class. The interface has methods which will make it possible to get and save events into the calendar, without knowing the actual implementation.

4.4 Package CalendarSystem.Model

Classes

- class Calendar

  A class containing a datastructure with IEvents and provides methods to get the events from the datastructure.
- class Event

  The event class implements the IEvent interface and represents a period of time at a given time, with a couple of other fields.
- interface IEvent

  An interface for event classes, used for given the minimum an event class must be able to do, and have.
- interface IObservable

  The interface for observable objects. Used to implement the observer pattern, such that the model can notify the view(controller)
- interface IObserver

  The interface for observing objects. Used to implement the observer pattern, such that the model can notify the view(controller)
- class Notification

  A class which has a date for when the notification enters an alarmstate and a description.
- class Tag

  Implementation is unclear for now

4.5 Package CalendarSystem.View

Classes

- class CalendarView

  The calendar view visually represents the calendar of events in the storage
- class EventView

  A class that visually represents an event object, or the creation thereof.
- class LoginView

  The login view is the view which is prompted to the user first. It contains two textboxes and a login button.
- class MainView

  The main view is the container of the other views as well as search bars, properties and so on.
- class NotificationView

  A class that visually represents an notification object, or the creation thereof.

4.6 Package CalendarSystemTests

Classes

- class InputControllerTest

Chapter 5

Class Documentation

5.1 CalendarSystem.Model.Calendar Class Reference

A class containing a datastructure with IEvents and provides methods to get the events from the datastructure.

Public Member Functions

- void **createCalenderEntry** (IEvent newEvent)
- void **updateCalenderEntry** (IEvent newEvent)
- IEvent **GetEvent** (int ID)

Public Attributes

- IList< IEvent > **_Events** = new List<IEvent>()

5.1.1 Detailed Description

A class containing a datastructure with IEvents and provides methods to get the events from the datastructure.

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

- CalendarSystem/Model/Calendar.cs

5.2 CalendarSystem.View.CalendarView Class Reference

The calendar view visually represents the calendar of events in the storage

Public Types

- enum OverviewType { **montly**, **weekly**, **daily** }

  An enum type of the different kinds of views.

Public Member Functions

- void changeOverviewType (OverviewType overviewType)

  Change the way the events are shown on screen

5.2.1 Detailed Description

The calendar view visually represents the calendar of events in the storage

5.2.2 Member Enumeration Documentation

5.2.2.1 enum **CalendarSystem.View.CalendarView.OverviewType**

An enum type of the different kinds of views.

5.2.3 Member Function Documentation

5.2.3.1 void CalendarSystem.View.CalendarView.changeOverviewType (**OverviewType** *overviewType*)

Change the way the events are shown on screen

Parameters

overviewType	

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

- CalendarSystem/View/CalendarView.cs

5.3 Class1 Class Reference

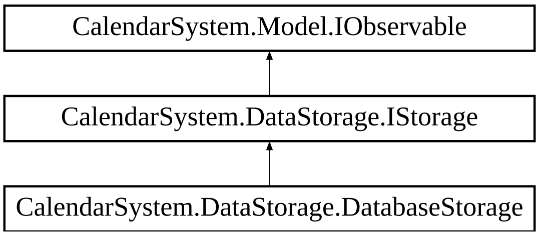
The documentation for this class was generated from the following file:

- Calendar.cs

5.4 CalendarSystem.DataStorage.DatabaseStorage Class Reference

A storage class which implements the IStorage interface. The class is meant to have a connection to a database where events will be added when they are created and put in the local Calendar class.

Inheritance diagram for CalendarSystem.DataStorage.DatabaseStorage:


CalendarSystem.Model.IObservable
CalendarSystem.DataStorage.IStorage
CalendarSystem.DataStorage.DatabaseStorage

Public Member Functions

- void loginAuthentication (string userName, string password)

  Authenticate and download Calendar and events belonging to that user.
- void SaveEvent (string description, DateTime dateTime, TimeSpan timeSpan, Notification notification)

  Save an event to the storage

- void UpdateEvent (int ID, string description, DateTime dateTime, TimeSpan timeSpan, Notification notification)

  Update an event to the storage
- IList< IEvent > GetAllEvents ()

  Returns all events in the active calendar
- IEvent GetEvent (int ID)

  Get a single event with a given ID if possible.
- IList< IEvent > GetEventsBetweenDates (DateTime beginDateTime, DateTime enDateTime)

  Return all events between to given dates.
- void CreateTag (Tag tag)

  Create a tag and save it in the storage.
- void NotifyObservers ()

  Notify observers that a change has happened.
- void BeObserved (IObserver observer)

  Become observed by an object which class has implemented IObserver
- void BeObserved (IList< IObserver > observers)

  Be observed by a list of IObserver(s).

5.4.1 Detailed Description

A storage class which implements the IStorage interface. The class is meant to have a connection to a database where events will be added when they are created and put in the local Calendar class.

5.4.2 Member Function Documentation

5.4.2.1 void CalendarSystem.DataStorage.DatabaseStorage.BeObserved (IObserver *observer*)

Become observed by an object which class has implemented IObserver

Parameters

observer	

Implements CalendarSystem.Model.IObservable.

5.4.2.2 void CalendarSystem.DataStorage.DatabaseStorage.BeObserved (IList< IObserver > *observers*)

Be observed by a list of IObserver(s).

Parameters

observers	

Implements CalendarSystem.Model.IObservable.

5.4.2.3 void CalendarSystem.DataStorage.DatabaseStorage.CreateTag (Tag *tag*)

Create a tag and save it in the storage.

Parameters

tag	

Implements CalendarSystem.DataStorage.IStorage.

5.4.2.4 IList< **IEvent** > CalendarSystem.DataStorage.DatabaseStorage.GetAllEvents ( )

Returns all events in the active calendar

Returns

Implements CalendarSystem.DataStorage.IStorage.

5.4.2.5 **IEvent** CalendarSystem.DataStorage.DatabaseStorage.GetEvent (int *ID*)

Get a single event with a given ID if possible.

Parameters

ID	

Returns

Implements CalendarSystem.DataStorage.IStorage.

5.4.2.6 IList< **IEvent** > CalendarSystem.DataStorage.DatabaseStorage.GetEventsBetweenDates (DateTime *beginDateTime,* DateTime *enDateTime*)

Return all events between to given dates.

Parameters

beginDateTime	
enDateTime	

Returns

Implements CalendarSystem.DataStorage.IStorage.

5.4.2.7 void CalendarSystem.DataStorage.DatabaseStorage.loginAuthentication (string *userName,* string *password*)

Authenticate and download Calendar and events belonging to that user.

Parameters

userName	
password	

Returns

Implements CalendarSystem.DataStorage.IStorage.

5.4.2.8 void CalendarSystem.DataStorage.DatabaseStorage.NotifyObservers ( )

Notify observers that a change has happened.

Implements CalendarSystem.Model.IObservable.

5.4.2.9 void CalendarSystem.DataStorage.DatabaseStorage.SaveEvent (string *description,* DateTime *dateTime,* TimeSpan *timeSpan,* Notification *notification*)

Save an event to the storage

Parameters

description	
dateTime	
timeSpan	
notification	

Implements CalendarSystem.DataStorage.IStorage.

5.4.2.10 void CalendarSystem.DataStorage.DatabaseStorage.UpdateEvent (int *ID,* string *description,* DateTime *dateTime,* TimeSpan *timeSpan,* Notification *notification*)

Update an event to the storage

Parameters

description	
dateTime	
timeSpan	
notification	

Implements CalendarSystem.DataStorage.IStorage.

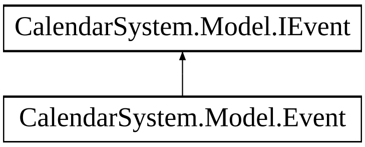
The documentation for this class was generated from the following file:

- CalendarSystem/DataStorage/DatabaseStorage.cs

5.5 CalendarSystem.Model.Event Class Reference

The event class implements the IEvent interface and represents a period of time at a given time, with a couple of other fields.

Inheritance diagram for CalendarSystem.Model.Event:



Public Member Functions

- **Event** (string description, TimeSpan timespan, DateTime date, Notification notification, int ID)
- void changeTag (Tag tag)

  Change the tag of the

Properties

- Notification **_notification** `[get, set]`
- DateTime **_date** `[get, set]`
- TimeSpan **_timeSpan** `[get, set]`
- string **_description** `[get, set]`
- int **_ID** `[get]`

5.5.1 Detailed Description

The event class implements the IEvent interface and represents a period of time at a given time, with a couple of other fields.

5.5.2 Member Function Documentation

5.5.2.1 void CalendarSystem.Model.Event.changeTag (Tag *tag*)

Change the tag of the

Parameters

tag	

Implements CalendarSystem.Model.IEvent.

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

- CalendarSystem/Model/Event.cs

5.6 CalendarSystem.View.EventView Class Reference

A class that visually represents an event object, or the creation thereof.

5.6.1 Detailed Description

A class that visually represents an event object, or the creation thereof.

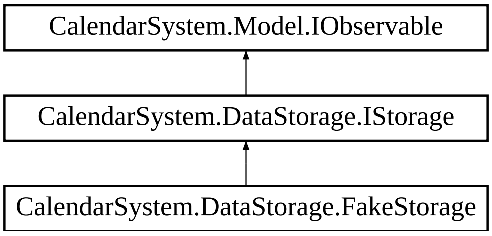
The documentation for this class was generated from the following file:

- CalendarSystem/View/EventView.cs

5.7 CalendarSystem.DataStorage.FakeStorage Class Reference

A storage class which implements the IStorage interface. The class is a fake storage in that sense that the fields and methods return components that are created at runtime. Therefore no events will be accessible from between runs of the system.

Inheritance diagram for CalendarSystem.DataStorage.FakeStorage:


CalendarSystem.Model.IObservable
CalendarSystem.DataStorage.IStorage
CalendarSystem.DataStorage.FakeStorage

Public Member Functions

- void loginAuthentication (string userName, string password)

  Authenticate and download Calendar and events belonging to that user.
- void SaveEvent (string description, DateTime dateTime, TimeSpan timeSpan, Notification notification)

  Save an event to the storage
- void UpdateEvent (int ID, string description, DateTime dateTime, TimeSpan timeSpan, Notification notification)

  Update an event to the storage
- IList< IEvent > GetAllEvents ()

  Returns all events in the active calendar
- IEvent GetEvent (int ID)

  Get a single event with a given ID if possible.
- IList< IEvent > GetEventsBetweenDates (DateTime beginDateTime, DateTime enDateTime)

  Return all events between to given dates.
- void CreateTag (Tag tag)

  Create a tag and save it in the storage.
- void NotifyObservers ()

  Notify observers that a change has happened.
- void BeObserved (IObserver observer)

  Become observed by an object which class has implemented IObserver
- void BeObserved (IList< IObserver > observers)

  Be observed by a list of IObserver(s).

5.7.1 Detailed Description

A storage class which implements the IStorage interface. The class is a fake storage in that sense that the fields and methods return components that are created at runtime. Therefore no events will be accessible from between runs of the system.

5.7.2 Member Function Documentation

5.7.2.1 void CalendarSystem.DataStorage.FakeStorage.BeObserved (**IObserver** *observer*)

Become observed by an object which class has implemented IObserver

Parameters

observer	

Implements CalendarSystem.Model.IObservable.

5.7.2.2 void CalendarSystem.DataStorage.FakeStorage.BeObserved (IList< **IObserver** > *observers*)

Be observed by a list of IObserver(s).

Parameters

observers	

Implements CalendarSystem.Model.IObservable.

5.7.2.3 void CalendarSystem.DataStorage.FakeStorage.CreateTag (Tag *tag*)

Create a tag and save it in the storage.

Parameters

tag	

Implements CalendarSystem.DataStorage.IStorage.

5.7.2.4 IList<IEvent> CalendarSystem.DataStorage.FakeStorage.GetAllEvents ( )

Returns all events in the active calendar

Returns

Implements CalendarSystem.DataStorage.IStorage.

5.7.2.5 IEvent CalendarSystem.DataStorage.FakeStorage.GetEvent (int *ID*)

Get a single event with a given ID if possible.

Parameters

ID	

Returns

Implements CalendarSystem.DataStorage.IStorage.

5.7.2.6 IList<IEvent> CalendarSystem.DataStorage.FakeStorage.GetEventsBetweenDates (DateTime *beginDateTime,* DateTime *enDateTime*)

Return all events between to given dates.

Parameters

beginDateTime	
enDateTime	

Returns

Implements CalendarSystem.DataStorage.IStorage.

5.7.2.7 void CalendarSystem.DataStorage.FakeStorage.loginAuthentication (string *userName,* string *password*)

Authenticate and download Calendar and events belonging to that user.

Parameters

userName	
password	

Returns

Implements CalendarSystem.DataStorage.IStorage.

5.7.2.8 void CalendarSystem.DataStorage.FakeStorage.NotifyObservers ( )

Notify observers that a change has happened.

Implements CalendarSystem.Model.IObservable.

5.7.2.9 void CalendarSystem.DataStorage.FakeStorage.SaveEvent (string *description,* DateTime *dateTime,* TimeSpan *timeSpan,* Notification *notification*)

Save an event to the storage

Parameters

description	
dateTime	
timeSpan	
notification	

Implements CalendarSystem.DataStorage.IStorage.

5.7.2.10 void CalendarSystem.DataStorage.FakeStorage.UpdateEvent (int *ID,* string *description,* DateTime *dateTime,* TimeSpan *timeSpan,* Notification *notification*)

Update an event to the storage

Parameters

description	
dateTime	
timeSpan	
notification	

Implements CalendarSystem.DataStorage.IStorage.

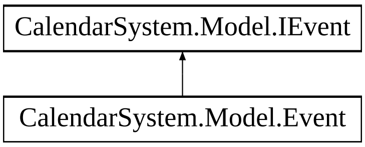
The documentation for this class was generated from the following file:

- CalendarSystem/DataStorage/FakeStorage.cs

5.8 CalendarSystem.Model.IEvent Interface Reference

An interface for event classes, used for given the minimum an event class must be able to do, and have.

Inheritance diagram for CalendarSystem.Model.IEvent:



Public Member Functions

- void **changeTag** (Tag tag)

Properties

- Notification **_notification** `[get, set]`
- DateTime **_date** `[get, set]`
- TimeSpan **_timeSpan** `[get, set]`
- string **_description** `[get, set]`
- int **_ID** `[get]`

5.8.1 Detailed Description

An interface for event classes, used for given the minimum an event class must be able to do, and have.

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

- CalendarSystem/Model/IEvent.cs

5.9 CalendarSystem.Controller.InputController Class Reference

The inputcontroller handles all incoming user interaction, such as button presses and so on.

Public Member Functions

- void CreateCalendarEntry (string description, int year, int month, int day, int startHour, int endHour)

  A method which will often be called from the view. The method takes some parameters and gives the to the storage, where an event will be created and uploaded.
- void UpdateCalendarEntry (int ID, string description, int year, int month, int day, int startHour, int endHour)

  A method which will often be called from the view. The method takes some parameters and gives the to the storage, where an event will be updated and uploaded.
- void CreateTag (string newTag)

  A method to create a tag and put it in the datastorage
- void ChangeOverviewType (string overviewType)

  Send the request for a new view type to the viewController.

Static Public Member Functions

- static InputController getInstance ()

  Singleton pattern. Makes sure only one instance can exist at a given time, and give classes easy access to the controller.

Properties

- IStorage **_storage** [get]

5.9.1 Detailed Description

The inputcontroller handles all incoming user interaction, such as button presses and so on.

5.9.2 Member Function Documentation

5.9.2.1 void CalendarSystem.Controller.InputController.ChangeOverviewType (string *overviewType*)

Send the request for a new view type to the viewController.

Parameters

overviewType	A string which represents a overview type

5.9.2.2 void CalendarSystem.Controller.InputController.CreateCalendarEntry (string *description,* int *year,* int *month,* int *day,* int *startHour,* int *endHour*)

A method which will often be called from the view. The method takes some parameters and gives the to the storage, where an event will be created and uploaded.

Parameters

description	The description of an event
month	The month of the event
day	The day of the event
startHour	The start hour of the event
endHour	The end hour of the event

5.9.2.3 void CalendarSystem.Controller.InputController.CreateTag (string *newTag*)

A method to create a tag and put it in the datastorage

Parameters

newTag	

5.9.2.4 static InputController CalendarSystem.Controller.InputController.getInstance ( ) [static]

Singleton pattern. Makes sure only one instance can exist at a given time, and give classes easy access to the controller.

Returns

The only instance of the class

5.9.2.5 void CalendarSystem.Controller.InputController.UpdateCalendarEntry (int *ID,* string *description,* int *year,* int *month,* int *day,* int *startHour,* int *endHour*)

A method which will often be called from the view. The method takes some parameters and gives the to the storage, where an event will be updated and uploaded.

Parameters

ID	The ID of the
description	The description of an event
month	The month of the event
day	The day of the event
startHour	The start hour of the event
endHour	The end hour of the event

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

- CalendarSystem/Controller/InputController.cs

5.10 CalendarSystemTests.InputControllerTest Class Reference

Public Member Functions

- void **Setup** ()
- void **placeholder** ()

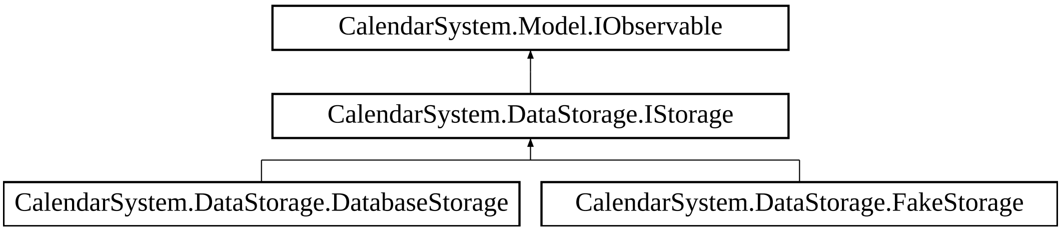
The documentation for this class was generated from the following file:

- CalendarSystemTests/InputControllerTest.cs

5.11 CalendarSystem.Model.IObservable Interface Reference

The interface for observable objects. Used to implement the observer pattern, such that the model can notify the view(controller)

Inheritance diagram for CalendarSystem.Model.IObservable:


CalendarSystem.Model.IObservable
CalendarSystem.DataStorage.IStorage
CalendarSystem.DataStorage.DatabaseStorage
CalendarSystem.DataStorage.FakeStorage

Public Member Functions

- void NotifyObservers ()

  Notify observers that a change has happened.
- void BeObserved (IObserver observer)

  Become observed by an object which class has implemented IObserver
- void BeObserved (IList< IObserver > observers)

  Be observed by a list of IObserver(s).

5.11.1 Detailed Description

The interface for observable objects. Used to implement the observer pattern, such that the model can notify the view(controller)

5.11.2 Member Function Documentation

5.11.2.1 void CalendarSystem.Model.IObservable.BeObserved (**IObserver** *observer*)

Become observed by an object which class has implemented IObserver

Parameters

observer	

Implemented in CalendarSystem.DataStorage.DatabaseStorage, and CalendarSystem.DataStorage.FakeStorage.

5.11.2.2 void CalendarSystem.Model.IObservable.BeObserved (IList< **IObserver** > *observers*)

Be observed by a list of IObserver(s).

Parameters

observers	

Implemented in CalendarSystem.DataStorage.DatabaseStorage, and CalendarSystem.DataStorage.FakeStorage.

5.11.2.3 void CalendarSystem.Model.IObservable.NotifyObservers ( )

Notify observers that a change has happened.

Implemented in CalendarSystem.DataStorage.DatabaseStorage, and CalendarSystem.DataStorage.FakeStorage.

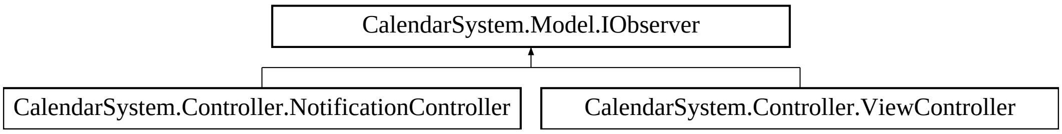
The documentation for this interface was generated from the following file:

- CalendarSystem/Model/IObservable.cs

5.12 CalendarSystem.Model.IObserver Interface Reference

The interface for observing objects. Used to implement the observer pattern, such that the model can notify the view(controller)

Inheritance diagram for CalendarSystem.Model.IObserver:


CalendarSystem.Model.IObserver
CalendarSystem.Controller.NotificationController
CalendarSystem.Controller.ViewController

Public Member Functions

- void BeNotifiedByObserved ()

  A method which is called from the IObservable objects which the IObserver has observed. The method notifies that the IObservable has been updated.

5.12.1 Detailed Description

The interface for observing objects. Used to implement the observer pattern, such that the model can notify the view(controller)

5.12.2 Member Function Documentation

5.12.2.1 void CalendarSystem.Model.IObserver.BeNotifiedByObserved ( )

A method which is called from the IObservable objects which the IObserver has observed. The method notifies that the IObservable has been updated.

Implemented in CalendarSystem.Controller.ViewController, and CalendarSystem.Controller.NotificationController.

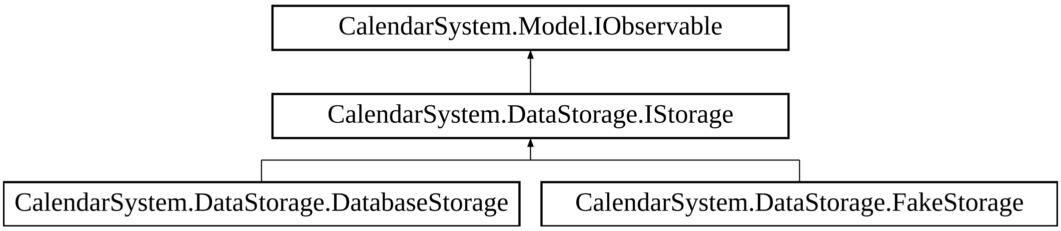
The documentation for this interface was generated from the following file:

- CalendarSystem/Model/IObserver.cs

5.13 CalendarSystem.DataStorage.IStorage Interface Reference

An interface for a storage class. The interface has methods which will make it possible to get and save events into the calendar, without knowing the actual implementation.

Inheritance diagram for CalendarSystem.DataStorage.IStorage:


CalendarSystem.Model.IObservable
CalendarSystem.DataStorage.IStorage
CalendarSystem.DataStorage.DatabaseStorage
CalendarSystem.DataStorage.FakeStorage

Public Member Functions

- void loginAuthentication (string userName, string password)

  Authenticate and download Calendar and events belonging to that user.
- void SaveEvent (string description, DateTime dateTime, TimeSpan timeSpan, Notification notification)

  Save an event to the storage
- void UpdateEvent (int ID, string description, DateTime dateTime, TimeSpan timeSpan, Notification notification)

  Update an event to the storage
- IList< IEvent > GetAllEvents ()

  Returns all events in the active calendar
- IEvent GetEvent (int ID)

  Get a single event with a given ID if possible.
- IList< IEvent > GetEventsBetweenDates (DateTime beginDateTime, DateTime enDateTime)

  Return all events between to given dates.
- void CreateTag (Tag tag)

  Create a tag and save it in the storage.

5.13.1 Detailed Description

An interface for a storage class. The interface has methods which will make it possible to get and save events into the calendar, without knowing the actual implementation.

5.13.2 Member Function Documentation

5.13.2.1 void CalendarSystem.DataStorage.IStorage.CreateTag (Tag *tag*)

Create a tag and save it in the storage.

Parameters

tag	

Implemented in CalendarSystem.DataStorage.DatabaseStorage, and CalendarSystem.DataStorage.FakeStorage.

5.13.2.2 IList< IEvent > CalendarSystem.DataStorage.IStorage.GetAllEvents ( )

Returns all events in the active calendar

Returns

Implemented in CalendarSystem.DataStorage.DatabaseStorage, and CalendarSystem.DataStorage.FakeStorage.

5.13.2.3 IEvent CalendarSystem.DataStorage.IStorage.GetEvent (int *ID*)

Get a single event with a given ID if possible.

Parameters

ID	

Returns

Implemented in CalendarSystem.DataStorage.DatabaseStorage, and CalendarSystem.DataStorage.FakeStorage.

5.13.2.4 IList< IEvent > CalendarSystem.DataStorage.IStorage.GetEventsBetweenDates (DateTime *beginDateTime,* DateTime *enDateTime*)

Return all events between to given dates.

Parameters

beginDateTime	
enDateTime	

Returns

Implemented in CalendarSystem.DataStorage.DatabaseStorage, and CalendarSystem.DataStorage.FakeStorage.

5.13.2.5 void CalendarSystem.DataStorage.IStorage.loginAuthentication (string *userName,* string *password*)

Authenticate and download Calendar and events belonging to that user.

Parameters

userName	
password	

Returns

Implemented in CalendarSystem.DataStorage.DatabaseStorage, and CalendarSystem.DataStorage.FakeStorage.

5.13.2.6 void CalendarSystem.DataStorage.IStorage.SaveEvent (string *description,* DateTime *dateTime,* TimeSpan *timeSpan,* Notification *notification*)

Save an event to the storage

Parameters

description	
dateTime	
timeSpan	
notification	

Implemented in CalendarSystem.DataStorage.DatabaseStorage, and CalendarSystem.DataStorage.FakeStorage.

5.13.2.7 void CalendarSystem.DataStorage.IStorage.UpdateEvent (int *ID,* string *description,* DateTime *dateTime,* TimeSpan *timeSpan,* Notification *notification*)

Update an event to the storage

Parameters

description	
dateTime	
timeSpan	
notification	

Implemented in CalendarSystem.DataStorage.DatabaseStorage, and CalendarSystem.DataStorage.FakeStorage.

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

- CalendarSystem/DataStorage/IStorage.cs

5.14 CalendarSystem.View.LoginView Class Reference

The login view is the view which is prompted to the user first. It contains two textboxes and a login button.

5.14.1 Detailed Description

The login view is the view which is prompted to the user first. It contains two textboxes and a login button.

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

- CalendarSystem/View/LoginView.cs

5.15 CalendarSystem.View.MainView Class Reference

The main view is the container of the other views as well as search bars, properties and so on.

Public Member Functions

- **MainView** (CalendarView calendarView)

5.15.1 Detailed Description

The main view is the container of the other views as well as search bars, properties and so on.

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

- CalendarSystem/View/MainView.cs

5.16 CalendarSystem.Model.Notification Class Reference

A class which has a date for when the notification enters an alarmstate and a description.

Public Member Functions

- **Notification** (DateTime date, string description)
- bool **isInAlarmState** ()

5.16.1 Detailed Description

A class which has a date for when the notification enters an alarmstate and a description.

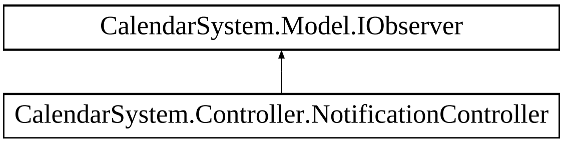
The documentation for this class was generated from the following file:

- CalendarSystem/Model/Notification.cs

5.17 CalendarSystem.Controller.NotificationController Class Reference

The Notification controller has a timer for the notifications and will create a popup when a notification's time is exceeded.

Inheritance diagram for CalendarSystem.Controller.NotificationController:


CalendarSystem.Model.IObserver
CalendarSystem.Controller.NotificationController

Public Member Functions

- void BeNotifiedByObserved ()

  A change in the model has happened.

Static Public Member Functions

- static NotificationController getInstance ()

  Singleton pattern. Makes sure only one instance can exist at a given time, and give classes easy access to the controller.

5.17.1 Detailed Description

The Notification controller has a timer for the notifications and will create a popup when a notification's time is exceeded.

5.17.2 Member Function Documentation

5.17.2.1 void CalendarSystem.Controller.NotificationController.BeNotifiedByObserved ( )

A change in the model has happened.

Implements CalendarSystem.Model.IObserver.

5.17.2.2 static NotificationController CalendarSystem.Controller.NotificationController.getInstance ( ) [static]

Singleton pattern. Makes sure only one instance can exist at a given time, and give classes easy access to the controller.

Returns

The only instance of the class

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

- CalendarSystem/Controller/NotificationController.cs

5.18 CalendarSystem.View.NotificationView Class Reference

A class that visually represents an notification object, or the creation thereof.

5.18.1 Detailed Description

A class that visually represents an notification object, or the creation thereof.

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

- CalendarSystem/View/NotificationView.cs

5.19 CalendarSystem.Model.Tag Class Reference

Implementation is unclear for now

Public Member Functions

- void **createTags** (string newTag, Color newColor)

5.19.1 Detailed Description

Implementation is unclear for now

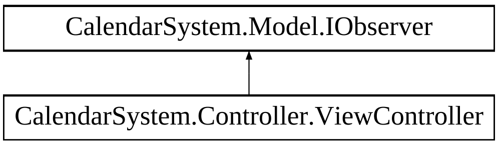
The documentation for this class was generated from the following file:

- CalendarSystem/Model/Tag.cs

5.20 CalendarSystem.Controller.ViewController Class Reference

The view controller handles all calls and creations of the view. It implements the IObserver interface and therefore can get notified when changes in the model happens.

Inheritance diagram for CalendarSystem.Controller.ViewController:


CalendarSystem.Model.IObserver
CalendarSystem.Controller.ViewController

Public Member Functions

- void startMainView ()

  Create the main view
- void createEventView ()

  Create a new event view
- void **createNotificationView** (int ID, Notification notification)
- void createEventView (IEvent iEvent)

  Create an Event view with the data from a already existing event (update event)
- void UpdateCalenderOverview (string overviewType)

  Change the overviewtype of the calendarview. Right now it does not use the enum type correctly. Must be updated to properly use the enum type.
- void BeNotifiedByObserved ()

  The observable pattern method of the observers. When a change in the model has happened (the model is the observable) it has to update the calendar view.

Static Public Member Functions

- static ViewController getInstance ()

  Singleton pattern. Makes sure only one instance can exist at a given time, and give classes easy access to the controller.

5.20.1 Detailed Description

The view controller handles all calls and creations of the view. It implements the IObserver interface and therefore can get notified when changes in the model happens.

5.20.2 Member Function Documentation

5.20.2.1 void CalendarSystem.Controller.ViewController.BeNotifiedByObserved ( )

The observable pattern method of the observers. When a change in the model has happened (the model is the observable) it has to update the calendar view.

Implements CalendarSystem.Model.IObserver.

5.20.2.2 void CalendarSystem.Controller.ViewController.createEventView ( )

Create a new event view

5.20.2.3 void CalendarSystem.Controller.ViewController.createEventView (IEvent *iEvent*)

Create an Event view with the data from a already existing event (update event)

Parameters

iEvent	

5.20.2.4 static ViewController CalendarSystem.Controller.ViewController.getInstance ( ) [static]

Singleton pattern. Makes sure only one instance can exist at a given time, and give classes easy access to the controller.

Returns

The only instance of the class

5.20.2.5 void CalendarSystem.Controller.ViewController.startMainView ( )

Create the main view

5.20.2.6 void CalendarSystem.Controller.ViewController.UpdateCalenderOverview (string *overviewType*)

Change the overviewtype of the calendarview. Right now it does not use the enum type correctly. Must be updated to properly use the enum type.

Parameters

overviewType	

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

- CalendarSystem/Controller/ViewController.cs