

GooGrade

Generated by Doxygen 1.5.5

Sun Mar 15 13:24:24 2009



# Contents

<b>1</b>	<b>Main Page</b>	<b>1</b>
1.1	Description . . . . .	1
1.2	Locations . . . . .	1
1.3	MVC . . . . .	1
1.4	Markup . . . . .	1
<b>2</b>	<b>Namespace Documentation</b>	<b>3</b>
2.1	Package blugoo . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	Account Class Reference . . . . .	5
3.2	AccountController Class Reference . . . . .	10
3.3	Announcement Class Reference . . . . .	11
3.4	AnnouncementController Class Reference . . . . .	14
3.5	Assignment Class Reference . . . . .	15
3.6	AssignmentController Class Reference . . . . .	18
3.7	Course Class Reference . . . . .	19
3.8	CourseController Class Reference . . . . .	23
3.9	Email Class Reference . . . . .	25
3.10	EmailAddress Class Reference . . . . .	27
3.11	EmailController Class Reference . . . . .	29
3.12	File Class Reference . . . . .	30
3.13	FileController Class Reference . . . . .	32
3.14	GradingRules Class Reference . . . . .	33
3.15	GradingRulesController Class Reference . . . . .	36
3.16	Login Class Reference . . . . .	37
3.17	Password Class Reference . . . . .	39
3.18	PasswordController Class Reference . . . . .	40
3.19	Permissions Class Reference . . . . .	41

3.20	PermissionsController Class Reference . . . . .	45
3.21	Serializable Class Reference . . . . .	47
3.22	StorageConnection Class Reference . . . . .	48
3.23	Student Class Reference . . . . .	49
3.24	StudentController Class Reference . . . . .	51
3.25	Teacher Class Reference . . . . .	52
3.26	TeacherAssistant Class Reference . . . . .	57
3.27	TeacherAssistantController Class Reference . . . . .	59
3.28	TeacherController Class Reference . . . . .	60
<b>4</b>	<b>File Documentation</b>	<b>65</b>
4.1	blugoo/WEB-INF/src/java/blugoo/Account.java File Reference . . . . .	65
4.2	blugoo/WEB-INF/src/java/blugoo/AccountController.java File Reference . . . . .	66
4.3	blugoo/WEB-INF/src/java/blugoo/Announcement.java File Reference . . . . .	67
4.4	blugoo/WEB-INF/src/java/blugoo/AnnouncementController.java File Reference . . . . .	68
4.5	blugoo/WEB-INF/src/java/blugoo/Assignment.java File Reference . . . . .	69
4.6	blugoo/WEB-INF/src/java/blugoo/AssignmentController.java File Reference . . . . .	70
4.7	blugoo/WEB-INF/src/java/blugoo/Course.java File Reference . . . . .	71
4.8	blugoo/WEB-INF/src/java/blugoo/CourseController.java File Reference . . . . .	72
4.9	blugoo/WEB-INF/src/java/blugoo/Email.java File Reference . . . . .	73
4.10	blugoo/WEB-INF/src/java/blugoo/EmailAddress.java File Reference . . . . .	74
4.11	blugoo/WEB-INF/src/java/blugoo/EmailController.java File Reference . . . . .	75
4.12	blugoo/WEB-INF/src/java/blugoo/File.java File Reference . . . . .	76
4.13	blugoo/WEB-INF/src/java/blugoo/FileController.java File Reference . . . . .	77
4.14	blugoo/WEB-INF/src/java/blugoo/GradingRules.java File Reference . . . . .	78
4.15	blugoo/WEB-INF/src/java/blugoo/GradingRulesController.java File Reference . . . . .	79
4.16	blugoo/WEB-INF/src/java/blugoo/Login.java File Reference . . . . .	80
4.17	blugoo/WEB-INF/src/java/blugoo/Password.java File Reference . . . . .	81
4.18	blugoo/WEB-INF/src/java/blugoo/PasswordController.java File Reference . . . . .	82
4.19	blugoo/WEB-INF/src/java/blugoo/Permissions.java File Reference . . . . .	83
4.20	blugoo/WEB-INF/src/java/blugoo/PermissionsController.java File Reference . . . . .	84
4.21	blugoo/WEB-INF/src/java/blugoo/StorageConnection.java File Reference . . . . .	85
4.22	blugoo/WEB-INF/src/java/blugoo/Student.java File Reference . . . . .	86
4.23	blugoo/WEB-INF/src/java/blugoo/StudentController.java File Reference . . . . .	87
4.24	blugoo/WEB-INF/src/java/blugoo/Teacher.java File Reference . . . . .	88
4.25	blugoo/WEB-INF/src/java/blugoo/TeacherAssistant.java File Reference . . . . .	89
4.26	blugoo/WEB-INF/src/java/blugoo/TeacherAssistantController.java File Reference . . . . .	90

---

4.27 blugoo/WEB-INF/src/java/blugoo/TeacherController.java File Reference . . . . .	91
4.28 front.txt File Reference . . . . .	92



# Chapter 1

## Main Page

### 1.1 Description

This is a project for CSC308, Winter 2009 with Clark Savage Turner. This is made by the bluGoo Team whose wiki can be found at <http://wiki.csc.calpoly.edu/blugoo/>

### 1.2 Locations

Interested in a printable version of this document? If you have the source code run 'make latex' otherwise check out the most recent published versions.

- PDF: <http://users.csc.calpoly.edu/~nwelch/blugoo/blugooarch.pdf>
- Web: <http://users.csc.calpoly.edu/~nwelch/blugoo/arch/>

### 1.3 MVC

This project follows the Model, View, Controller Philosophy. Views are Java Server Pages. Models are classes containing Get and Set methods and the ability to store data to the storage system. Controllers act with the models and the views to do actual specific work.

### 1.4 Markup

Doxygen generates [HTML 4.01 Transitional](#) documents with [Cascading Style Sheets \(CSS\)](#) and you can include markup tags that are compatible with those standards.





## Chapter 2

# Namespace Documentation

### 2.1 Package blugoo

#### Classes

- class [Account](#)  
*This is a class that keeps all of the information of a user.*
- class [AccountController](#)  
*This class does most of the work on the [Account](#).*
- class [Announcement](#)  
*An [Announcement](#) for a [Course](#).*
- class [AnnouncementController](#)  
*This is the controller for [Announcement](#).*
- class [Assignment](#)  
*This class keeps information about Assignments.*
- class [AssignmentController](#)  
*Does all of the work on the [Assignment](#), it is the controller.*
- class [Course](#)  
*This class keeps information about Courses.*
- class [CourseController](#)  
*Methods for modifying a course and it's related actions.*
- class [Email](#)  
*The class stores information about Emails.*
- class [EmailAddress](#)  
*A model for email addresses since they need to be validated and such.*

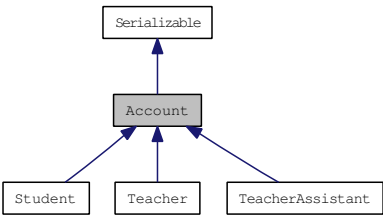
- class [EmailController](#)  
*This deals with sending emails.*
- class [File](#)  
*This class keeps information about Files.*
- class [FileController](#)  
*This class is to deal with files.*
- class [GradingRules](#)  
*The [GradingRules](#) class determines how the Total Grade is computed from mutiple [Assignment](#) Grades.*
- class [GradingRulesController](#)  
*The [GradingRulesController](#) class is the [GradingRules](#) controller.*
- class [Login](#)  
*This should deal with login I guess.*
- class [Password](#)  
*A model for passwords since they need to be validated and such.*
- class [PasswordController](#)  
*A controller for passwords since they need to be validated and such.*
- class [Permissions](#)  
*This class keeps information about permissions for each action.*
- class [PermissionsController](#)  
*This class keeps the permissions controller.*
- class [StorageConnection](#)  
*This is the class that deals with connecting to the storage device.*
- class [Student](#)  
*This class keeps information about Students.*
- class [StudentController](#)  
*This class deals with controlling Students.*
- class [Teacher](#)  
*A [Teacher](#) is an [Account](#) type that has the most permissions.*
- class [TeacherAssistant](#)  
*This class keeps information about [Teacher](#) Assistants.*
- class [TeacherAssistantController](#)  
*This class controlls [Teacher](#) Assistants.*
- class [TeacherController](#)  
*This is a controller for teacher.*

# Chapter 3

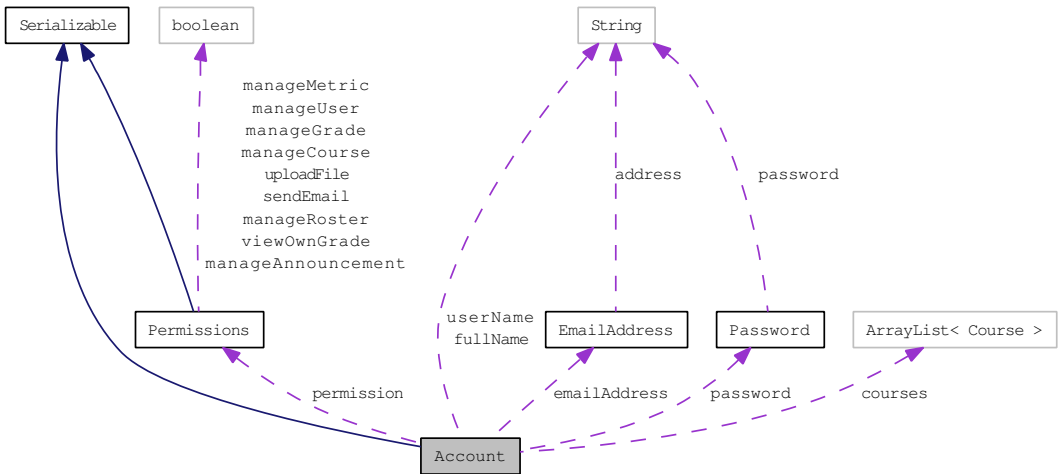
## Class Documentation

### 3.1 Account Class Reference

Inheritance diagram for Account:



Collaboration diagram for Account:



#### 3.1.1 Detailed Description

This is a class that keeps all of the information of a user.

A user is a synonym for an [Account](#). This will have lots of get and set methods that will handle getting and saving Accounts to the Storage system using the Storage Connection class.

**Author:**

bluGoo

**Version:**

0.42

**Public Member Functions**

- [EmailAddress](#) getEmailAddress ()  
*gets the [EmailAddress](#) of this account.*
- boolean setEmailAddress ()  
*sets the [EmailAddress](#) of this [Account](#).*

**Package Functions**

- String getUsername ()  
*get the accounts user name.*
- String getFullName ()  
*get the accounts full name.*
- boolean setPassword (String newpass)  
*Takes new password, hashes it, and saves to database.*
- String setUsername ()  
*set the Account's user name.*
- String setFullName ()  
*set the Account's full name.*
- boolean isPassword (String passwd)  
*Checks the provided password to see if they match.*

**Private Attributes**

- String [userName](#)  
*Username for logging in.*
- String [fullName](#)  
*Full name of the [Account](#) owner.*
- [EmailAddress](#) emailAddress

*Email address of the [Account](#) owner.*

- [Password](#) password

*Password for logging in.*

- `ArrayList< Course > courses`

*An array of Courses that the User is involved in.*

- [Permissions](#) permission

*Permissions for performing actions.*

### 3.1.2 Member Function Documentation

#### 3.1.2.1 `String getUsername ()` [package]

get the accounts user name.

**Returns:**

a string with the user name of the current account

#### 3.1.2.2 `String getFullName ()` [package]

get the accounts full name.

**Returns:**

a string with the full name of the current account

#### 3.1.2.3 `EmailAddress getEmailAddress ()`

gets the [EmailAddress](#) of this account.

**Returns:**

this user's email address

#### 3.1.2.4 `boolean setPassword (String newpass)` [package]

Takes new password, hashes it, and saves to database.

**Parameters:**

*newpass* the new password to be saved

**Returns:**

true if set, false if failure.

**3.1.2.5 String setUsername ()** [package]

set the Account's user name.

**Returns:**

true if set, false if error.

**3.1.2.6 String setFullName ()** [package]

set the Account's full name.

**Returns:**

true if set, false if error.

**3.1.2.7 boolean setEmailAddress ()**

sets the [EmailAddress](#) of this [Account](#).

**Returns:**

true if set, false if error.

**3.1.2.8 boolean isPassword (String *passwd*)** [package]

Checks the provided password to see if they match.

**Parameters:**

*passwd* the password entered by the user that will be compared with the stored value.

**Returns:**

true if the hashed passwd matches the password in the database, false otherwise.

**3.1.3 Member Data Documentation****3.1.3.1 String userName** [private]

Username for logging in.

**3.1.3.2 String fullName** [private]

Full name of the [Account](#) owner.

**3.1.3.3 EmailAddress emailAddress** [private]

[Email](#) address of the [Account](#) owner.

**3.1.3.4 Password password** [private]

[Password](#) for logging in.

This will not be stored in plain text due to privacy concerns.

**3.1.3.5 ArrayList<Course> courses** [private]

An array of Courses that the User is involved in.

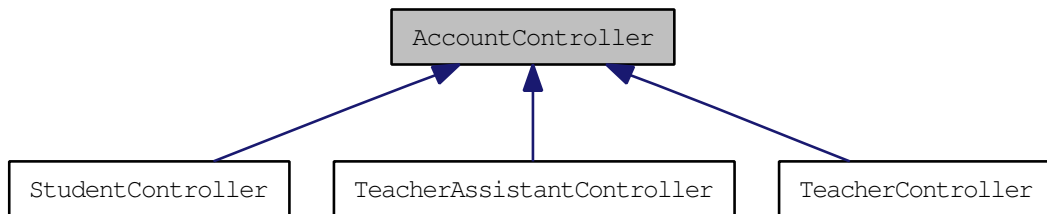
Reimplemented in [Teacher](#).

**3.1.3.6 Permissions permission** [private]

[Permissions](#) for performing actions.

## 3.2 AccountController Class Reference

Inheritance diagram for AccountController:



### 3.2.1 Detailed Description

This class does most of the work on the [Account](#).

It is the [Account](#) controller. Not really sure what it should be doing, but we'll get to that.

**Author:**

bluGoo

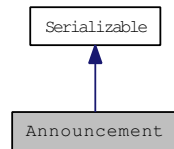
**Version:**

0.42

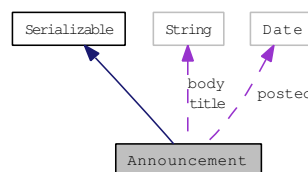


## 3.3 Announcement Class Reference

Inheritance diagram for Announcement:



Collaboration diagram for Announcement:



### 3.3.1 Detailed Description

An [Announcement](#) for a [Course](#).

Quite a simple class for saving and storing Announcements. There should be a method in [Course](#) to obtain an array of these associated with said [Course](#).

#### Author:

bluGoo

#### Version:

0.42

### Public Member Functions

- String [getBody](#) ()  
*Returns the body, aka the content, of an [Announcement](#).*
- Date [getDate](#) ()  
*Returns when this [Announcement](#) was posted.*
- String [getTitle](#) ()  
*Returns the title of this [Announcement](#).*
- boolean [setBody](#) ()  
*Sets the body of the [Announcement](#).*
- Date [setDate](#) ()  
*Sets the date of the [Announcement](#).*

- String `setTitle ()`  
*sets the title of the announcement.*

## Private Attributes

- String `body`
- String `title`
- Date `posted`

## 3.3.2 Member Function Documentation

### 3.3.2.1 String `getBody ()`

Returns the body, aka the content, of an [Announcement](#).

#### Returns:

the content of the post in a single string.

### 3.3.2.2 Date `getDate ()`

Returns when this [Announcement](#) was posted.

#### Returns:

the date this was posted.

### 3.3.2.3 String `getTitle ()`

Returns the title of this [Announcement](#).

#### Returns:

a string containing the title.

### 3.3.2.4 boolean `setBody ()`

Sets the body of the [Announcement](#).

#### Returns:

false if error. True if successful

#### 3.3.2.5 Date setDate ()

Sets the date of the [Announcement](#).

**Returns:**

false if error. True if successful

#### 3.3.2.6 String setTitle ()

sets the title of the announcement.

**Returns:**

false if error. True if successful

### 3.3.3 Member Data Documentation

**3.3.3.1 String body** [private]

**3.3.3.2 String title** [private]

**3.3.3.3 Date posted** [private]

## 3.4 AnnouncementController Class Reference

### 3.4.1 Detailed Description

This is the controller for [Announcement](#).

All it does is post things.

**Author:**

bluGoo

**Version:**

0.42

### Package Functions

- boolean [post](#) (String title, String post)

*Posts an announcement.*

### 3.4.2 Member Function Documentation

#### 3.4.2.1 boolean [post](#) (String *title*, String *post*) [package]

Posts an announcement.

This will call the functions needed to save to the database.

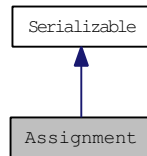
**Parameters:**

*title* The title

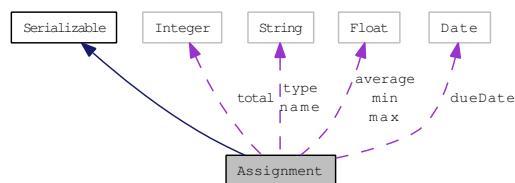
*post* The post body

## 3.5 Assignment Class Reference

Inheritance diagram for Assignment:



Collaboration diagram for Assignment:



### 3.5.1 Detailed Description

This class keeps information about Assignments.

**Author:**

bluGoo

**Version:**

0.42

### Public Member Functions

- Date [getDueDate](#) ()
- String [getName](#) ()
- Integer [getTotal](#) ()
- String [getType](#) ()
- Float [getAvg](#) ()
- Float [getMax](#) ()
- Float [getMin](#) ()

### Private Attributes

- Date [dueDate](#)  
*The due date of the [Assignment](#).*
- String [name](#)  
*The name of the [Assignment](#).*

- Integer [total](#)  
*The maximum possible score of the [Assignment](#).*
- String [type](#)  
*The type of the [Assignment](#); can be Test, Quiz, Homework, Participation, or other.*
- Float [average](#)  
*The average for all graded assignments for this assignment.*
- Float [max](#)  
*The maximum for all graded assignments for this assignment.*
- Float [min](#)  
*The minimum for all graded assignments for this assignment.*

## 3.5.2 Member Function Documentation

### 3.5.2.1 Date `getDueDate ()`

### 3.5.2.2 String `getName ()`

### 3.5.2.3 Integer `getTotal ()`

### 3.5.2.4 String `getType ()`

### 3.5.2.5 Float `getAvg ()`

### 3.5.2.6 Float `getMax ()`

### 3.5.2.7 Float `getMin ()`

## 3.5.3 Member Data Documentation

### 3.5.3.1 Date `dueDate` [private]

The due date of the [Assignment](#).

### 3.5.3.2 String `name` [private]

The name of the [Assignment](#).

### 3.5.3.3 Integer `total` [private]

The maximum possible score of the [Assignment](#).

### 3.5.3.4 String `type` [private]

The type of the [Assignment](#); can be Test, Quiz, Homework, Participation, or other.

**3.5.3.5 Float average** [private]

The average for all graded assignments for this assignment.

**3.5.3.6 Float max** [private]

The maximum for all graded assignments for this assignment.

**3.5.3.7 Float min** [private]

The minimum for all graded assignments for this assignment.

## 3.6 AssignmentController Class Reference

### 3.6.1 Detailed Description

Does all of the work on the [Assignment](#), it is the controller.

We haven't really speced out what this is going to do yet.

**Author:**

bluGoo

**Version:**

0.42

### Public Member Functions

- void [buildMetrics](#) ()

*More of a concept than anything.*

### 3.6.2 Member Function Documentation

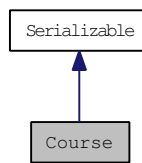
#### 3.6.2.1 void buildMetrics ()

More of a concept than anything.

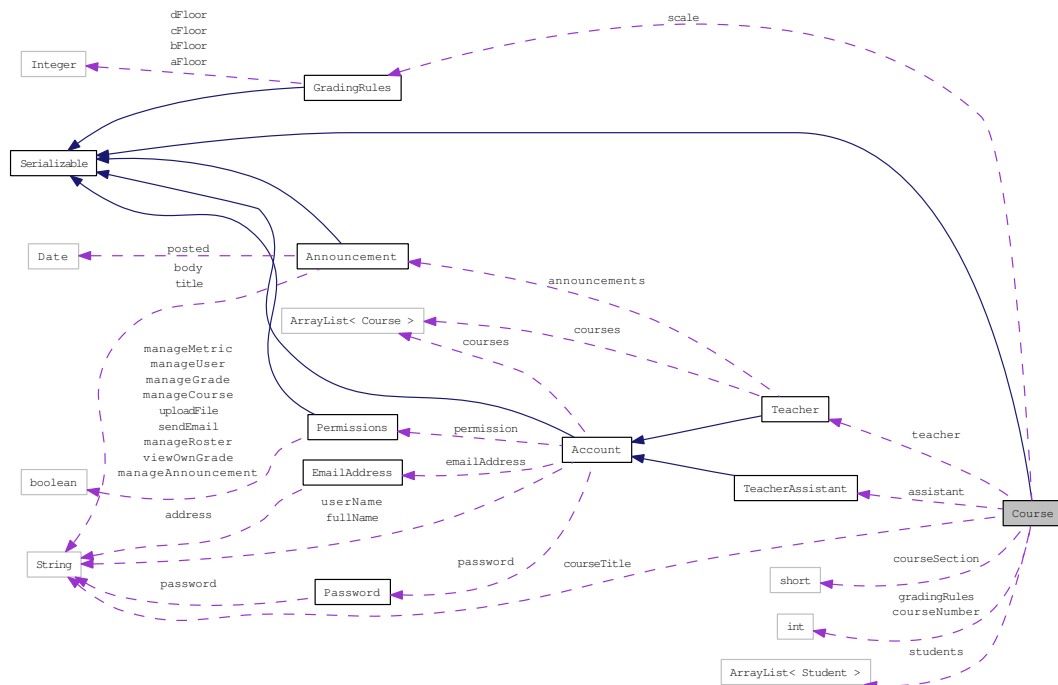


## 3.7 Course Class Reference

Inheritance diagram for Course:



Collaboration diagram for Course:



### 3.7.1 Detailed Description

This class keeps information about Courses.

It's a JavaBean since it implements `java.io.Serializable`. It too uses `StorageConnection` to store and get things out of the storage system.

**Author:**

bluGoo

**Version:**

0.42

## Public Member Functions

- [Course](#) ()  
*Constructor of [Course](#) class.*
- boolean [setTitle](#) (String newTitle)  
*setTitle saves a new Course title over the old.*
- boolean [setSection](#) (short newSection)  
*setSection saves a new section number over the old one*
- boolean [setNumber](#) (int newNumber)  
*setNumber replaces the course number with a new one*
- boolean [setTeacher](#) ([Teacher](#) t)  
*setTeacher sets the [Teacher](#) attribute for this course*
- [Teacher](#) [getTeacher](#) ()  
*getTeacher gets the [Teacher](#) attribute from this course*

## Private Attributes

- int [gradingRules](#)  
*The grading rules for the [Course](#).*
- String [courseTitle](#)  
*The Course's title.*
- short [courseSection](#)  
*The Course's section.*
- int [courseNumber](#)  
*The Course's number.*
- ArrayList< [Student](#) > [students](#)  
*An array of Students who are enrolled in the [Course](#).*
- [TeacherAssistant](#) [assistant](#)  
*An instance of [Teacher](#) Assistant who is associated with the [Course](#).*
- [Teacher](#) [teacher](#)  
*An instance of [Teacher](#) who is teaching the [Course](#).*
- [GradingRules](#) [scale](#)  
*The set of grade boundaries for this class.*

## 3.7.2 Constructor & Destructor Documentation

### 3.7.2.1 Course ()

Constructor of [Course](#) class.

## 3.7.3 Member Function Documentation

### 3.7.3.1 boolean setTitle (String *newTitle*)

setTitle saves a new Course title over the old.

**Parameters:**

*newTitle* the new title to save

**Returns:**

true if no errors

### 3.7.3.2 boolean setSection (short *newSection*)

setSection saves a new section number over the old one

**Parameters:**

*newSection* is the number of the new section to save

**Returns:**

true if no errors occurred

### 3.7.3.3 boolean setNumber (int *newNumber*)

setNumber replaces the course number with a new one

**Parameters:**

*newNumber* is the new course number to save

**Returns:**

true if no errors occurred.

### 3.7.3.4 boolean setTeacher (Teacher *t*)

setTeacher sets the [Teacher](#) attribute for this course

**Parameters:**

*t* the [Teacher](#) to be assigned to this course

**Returns:**

true if no error have occurred

### 3.7.3.5 Teacher getTeacher ()

getTeacher gets the [Teacher](#) attribute from this course

#### Returns:

teacher of this course

## 3.7.4 Member Data Documentation

### 3.7.4.1 int gradingRules [private]

The grading rules for the [Course](#).

### 3.7.4.2 String courseTitle [private]

The Course's title.

### 3.7.4.3 short courseSection [private]

The Course's section.

### 3.7.4.4 int courseNumber [private]

The Course's number.

### 3.7.4.5 ArrayList<Student> students [private]

An array of Students who are enrolled in the [Course](#).

### 3.7.4.6 TeacherAssistant assistant [private]

An instance of [Teacher](#) Assistant who is associated with the [Course](#).

### 3.7.4.7 Teacher teacher [private]

An instance of [Teacher](#) who is teaching the [Course](#).

### 3.7.4.8 GradingRules scale [private]

The set of grade boundaries for this class.

## 3.8 CourseController Class Reference

### 3.8.1 Detailed Description

Methods for modifying a course and it's related actions.

**Author:**

bluGoo

**Version:**

0.42

### Public Member Functions

- boolean [addStudent](#) ([Permissions](#) permission, [Student](#) student)  
*addStudent adds a new student to this course.*
- boolean [removeStudent](#) ([Permissions](#) permission, [Student](#) student)  
*removeStudent removes a particular student from the course.*
- boolean [addTA](#) ([Permissions](#) permission, [TeacherAssistant](#) ta)  
*addTA assigns a new [TeacherAssistant](#) to this course.*
- boolean [removeTA](#) ([Permissions](#) permission, [TeacherAssistant](#) ta)  
*removeTA removes a [TeacherAssistant](#) from this course*

### 3.8.2 Member Function Documentation

#### 3.8.2.1 boolean addStudent ([Permissions](#) *permission*, [Student](#) *student*)

[addStudent](#) adds a new student to this course.

**Parameters:**

*permission* the permission of the user who calls this function

*student* the student that is to be added to the course

**Returns:**

true if no errors were encountered

#### 3.8.2.2 boolean removeStudent ([Permissions](#) *permission*, [Student](#) *student*)

[removeStudent](#) removes a particular student from the course.

**Parameters:**

*permission* the permission of the user who calls this function

*student* the student to be removed from this course

**Returns:**

true if no errors were encountered in removal.

**3.8.2.3 boolean addTA (Permissions *permission*, TeacherAssistant *ta*)**

addTA assigns a new [TeacherAssistant](#) to this course.

**Parameters:**

*permission* the permission of the user who calls this function

*ta* the [TeacherAssistant](#) to be assigned to this course

**Returns:**

true if no errors have occurred

**3.8.2.4 boolean removeTA (Permissions *permission*, TeacherAssistant *ta*)**

removeTA removes a [TeacherAssistant](#) from this course

**Parameters:**

*permission* the permission of the user who calls this function

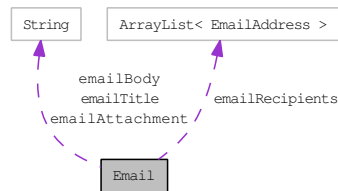
*ta* the [TeacherAssistant](#) to be removed to this course

**Returns:**

true if no errors have occurred

## 3.9 Email Class Reference

Collaboration diagram for Email:



### 3.9.1 Detailed Description

The class stores information about Emails.

These emails will not be stored in the database. This is purely a way to create a wrapper for validation and sending of emails.

#### Author:

bluGoo

#### Version:

0.42

### Public Member Functions

- [Email \(\)](#)

*Constructor for [Email](#) class.*

### Private Attributes

- String [emailTitle](#)

*The title of the [Email](#).*

- String [emailAttachment](#)

*The pathname of a file to be attached to the [Email](#).*

- ArrayList< [EmailAddress](#) > [emailRecipients](#)

*An array of email addresses to which the [Email](#) will be sent.*

- String [emailBody](#)

*The body of the [Email](#).*

## 3.9.2 Constructor & Destructor Documentation

### 3.9.2.1 Email ()

Constructor for [Email](#) class.

## 3.9.3 Member Data Documentation

### 3.9.3.1 String emailTitle [private]

The title of the [Email](#).

### 3.9.3.2 String emailAttachment [private]

The pathname of a file to be attached to the [Email](#).

### 3.9.3.3 ArrayList<EmailAddress> emailRecipients [private]

An array of email addresses to which the [Email](#) will be sent.

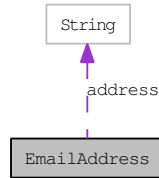
### 3.9.3.4 String emailBody [private]

The body of the [Email](#).



## 3.10 EmailAddress Class Reference

Collaboration diagram for EmailAddress:



### 3.10.1 Detailed Description

A model for email addresses since they need to be validated and such.

**Author:**

bluGoo

**Version:**

0.42

### Public Member Functions

- String [toString](#) ()  
*Obtain a string form of the [Email](#) Address.*

### Package Functions

- boolean [valid](#) (String in)  
*Basically a wrapper for validation.*

### Private Attributes

- String [address](#)

### 3.10.2 Member Function Documentation

#### 3.10.2.1 boolean [valid](#) (String *in*) [package]

Basically a wrapper for validation.

**Parameters:**

*in* [Email](#) address to validate

**Returns:**

boolean if is a valid email address or not.

### 3.10.2.2 String toString ()

Obtain a string form of the [Email](#) Address.

**Returns:**

a String of this [EmailAddress](#)

## 3.10.3 Member Data Documentation

### 3.10.3.1 String address [private]

## 3.11 EmailController Class Reference

### 3.11.1 Detailed Description

This deals with sending emails.

**Author:**

bluGoo

**Version:**

0.42

### Public Member Functions

- void [send](#) ()  
*Sends the email once it has been written.*

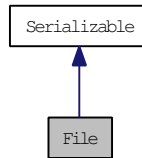
### 3.11.2 Member Function Documentation

#### 3.11.2.1 void send ()

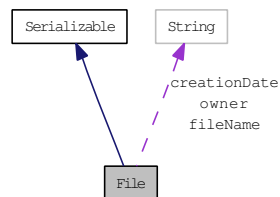
Sends the email once it has been written.

## 3.12 File Class Reference

Inheritance diagram for File:



Collaboration diagram for File:



### 3.12.1 Detailed Description

This class keeps information about Files.

**Author:**

bluGoo

**Version:**

0.42

### Private Attributes

- String [creationDate](#)  
*The time and date the [File](#) was created or uploaded.*
- String [owner](#)  
*The owner of the [File](#).*
- String [fileName](#)  
*The name of the [File](#).*

### 3.12.2 Member Data Documentation

#### 3.12.2.1 String [creationDate](#) [private]

The time and date the [File](#) was created or uploaded.

**3.12.2.2 String owner** [private]

The owner of the [File](#).

**3.12.2.3 String fileName** [private]

The name of the [File](#).

## 3.13 FileController Class Reference

### 3.13.1 Detailed Description

This class is to deal with files.

This is not a good implementation.

**Author:**

bluGoo

**Version:**

0.42

### Public Member Functions

- void [upload](#) ()  
*Upload to system.*
- void [download](#) ()  
*Download a file.*

### 3.13.2 Member Function Documentation

#### 3.13.2.1 void upload ()

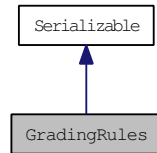
Upload to system.

#### 3.13.2.2 void download ()

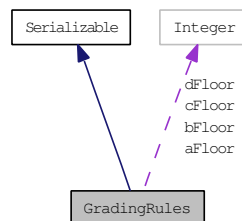
Download a file.

## 3.14 GradingRules Class Reference

Inheritance diagram for GradingRules:



Collaboration diagram for GradingRules:



### 3.14.1 Detailed Description

The [GradingRules](#) class determines how the Total Grade is computed from mutiple [Assignment](#) Grades.

**Author:**

Blugoo

**Version:**

0.42

### Package Functions

- boolean [setCurve](#) (Character letter, Integer grade)  
*Sets a the lowest percentage one can have to have a grade.*
- Integer [getCurve](#) (Character letter)  
*Gets a the lowest percentage one can have to have a grade.*

### Private Attributes

- Integer [aFloor](#)  
*The lowest percentage a student can have to obtain an A in the course.*
- Integer [bFloor](#)  
*The lowest percentage a student can have to obtain an B in the course.*

- Integer [cFloor](#)

*The lowest percentage a student can have to obtain an C in the course.*

- Integer [dFloor](#)

*The lowest percentage a student can have to obtain an D in the course.*

### 3.14.2 Member Function Documentation

#### 3.14.2.1 boolean setCurve (Character *letter*, Integer *grade*) [package]

Sets a the lowest percentage one can have to have a grade.

##### Parameters:

*letter* The Grade (A,B,C,D) you wish to change

*grade* The Percentage you wish to set the floor to.

##### Returns:

false if fail, true if set.

#### 3.14.2.2 Integer getCurve (Character *letter*) [package]

Gets a the lowest percentage one can have to have a grade.

##### Parameters:

*letter* The Grade (A,B,C,D) you wish to get

##### Returns:

false if fail, true if set.

### 3.14.3 Member Data Documentation

#### 3.14.3.1 Integer aFloor [private]

The lowest percentage a student can have to obtain an A in the course.

#### 3.14.3.2 Integer bFloor [private]

The lowest percentage a student can have to obtain an B in the course.

#### 3.14.3.3 Integer cFloor [private]

The lowest percentage a student can have to obtain an C in the course.



**3.14.3.4 Integer dFloor** [private]

The lowest percentage a student can have to obtain an D in the course.

## 3.15 GradingRulesController Class Reference

### 3.15.1 Detailed Description

The [GradingRulesController](#) class is the [GradingRules](#) controller.

**Author:**

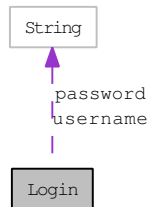
Blugoo

**Version:**

0.42

## 3.16 Login Class Reference

Collaboration diagram for Login:



### 3.16.1 Detailed Description

This should deal with login I guess.

Taken from the struts example.

#### Public Member Functions

- String `execute ()` throws Exception
- String `getUsername ()`
- void `setUsername (String username)`
- String `getPassword ()`
- void `setPassword (String password)`

#### Private Member Functions

- boolean `isInvalid (String value)`

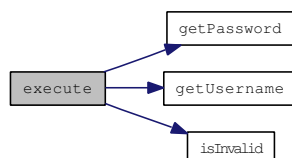
#### Private Attributes

- String `username`
- String `password`

### 3.16.2 Member Function Documentation

#### 3.16.2.1 String execute () throws Exception

Here is the call graph for this function:



### 3.16.2.2 boolean isInvalid (String *value*) [private]

Here is the caller graph for this function:



### 3.16.2.3 String getUsername ()

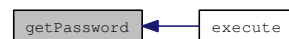
Here is the caller graph for this function:



### 3.16.2.4 void setUsername (String *username*)

### 3.16.2.5 String getPassword ()

Here is the caller graph for this function:



### 3.16.2.6 void setPassword (String *password*)

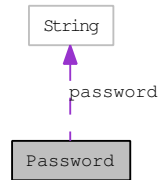
## 3.16.3 Member Data Documentation

### 3.16.3.1 String username [private]

### 3.16.3.2 String password [private]

## 3.17 Password Class Reference

Collaboration diagram for Password:



### 3.17.1 Detailed Description

A model for passwords since they need to be validated and such.

**Author:**

bluGoo

**Version:**

0.42

### Package Functions

- [Password](#) (String *newpass*)  
*Basically a wrapper for hashing.*

### Private Attributes

- String [password](#)

### 3.17.2 Constructor & Destructor Documentation

#### 3.17.2.1 Password (String *newpass*) [package]

Basically a wrapper for hashing.

### 3.17.3 Member Data Documentation

#### 3.17.3.1 String `password` [private]

## 3.18 PasswordController Class Reference

### 3.18.1 Detailed Description

A controller for passwords since they need to be validated and such.

**Author:**

bluGoo

**Version:**

0.42



## Private Attributes

- boolean [sendEmail](#)  
*Permission for sending emails.*
- boolean [uploadFile](#)  
*Permission for uploading files.*
- boolean [viewOwnGrade](#)  
*Permission for viewing personal grades.*
- boolean [manageCourse](#)  
*Permission for managing Courses.*
- boolean [manageMetric](#)  
*Permission for managing metrics.*
- boolean [manageGrade](#)  
*Permission for managing grades.*
- boolean [manageRoster](#)  
*Permission for managing the roster.*
- boolean [manageAnnouncement](#)  
*Permission for managing Assignments.*
- boolean [manageUser](#)  
*Permission for managing Users.*

## 3.19.2 Constructor & Destructor Documentation

### 3.19.2.1 Permissions ()

Constructor for [Permissions](#) class.

## 3.19.3 Member Function Documentation

### 3.19.3.1 boolean VerifyPermission (int *inquirer*)

VerifyPermission determines if an account has the permission required to perform another method.

#### Parameters:

*inquirer* indicates the specific permission that is being inquired about.

#### Returns:

permission is true if the account has permission to perform the inquired action.



### 3.19.3.2 void AddPermission (int *permission*)

AddPermission grants a new permission to a user.

It does nothing to permissions already granted.

**Parameters:**

*permission* to determine which permission is to be granted

### 3.19.3.3 void RemovePermission (int *permission*)

Remove Permission prevents a user from using a specific permission.

It does nothing to permissions previously not granted.

**Parameters:**

*permission* to determine which permission to be removed

## 3.19.4 Member Data Documentation

### 3.19.4.1 boolean sendEmail [private]

Permission for sending emails.

### 3.19.4.2 boolean uploadFile [private]

Permission for uploading files.

### 3.19.4.3 boolean viewOwnGrade [private]

Permission for viewing personal grades.

### 3.19.4.4 boolean manageCourse [private]

Permission for managing Courses.

### 3.19.4.5 boolean manageMetric [private]

Permission for managing metrics.

### 3.19.4.6 boolean manageGrade [private]

Permission for managing grades.

### 3.19.4.7 boolean manageRoster [private]

Permission for managing the roster.

**3.19.4.8 boolean manageAnnouncement** [private]

Permission for managing Assignments.

**3.19.4.9 boolean manageUser** [private]

Permission for managing Users.

## 3.20 PermissionsController Class Reference

### 3.20.1 Detailed Description

This class keeps the permissions controller.

**Author:**

bluGoo

**Version:**

0.42

### Public Member Functions

- boolean [VerifyPermission](#) (int inquirer)  
*VerifyPermission determines if an account has the permission required to perform another method.*
- void [AddPermission](#) (int permission)  
*AddPermission grants a new permission to a user.*
- void [RemovePermission](#) (int permission)  
*Remove Permission prevents a user from using a specific permission.*

### 3.20.2 Member Function Documentation

#### 3.20.2.1 boolean [VerifyPermission](#) (int *inquirer*)

[VerifyPermission](#) determines if an account has the permission required to perform another method.

**Parameters:**

*inquirer* indicates the specific permission that is being inquired about.

**Returns:**

permission is true if the account has permission to perform the inquired action.

#### 3.20.2.2 void [AddPermission](#) (int *permission*)

[AddPermission](#) grants a new permission to a user.

It does nothing to permissions already granted.

**Parameters:**

*permission* to determine which permission is to be granted

### 3.20.2.3 void RemovePermission (int *permission*)

Remove Permission prevents a user from using a specific permission.

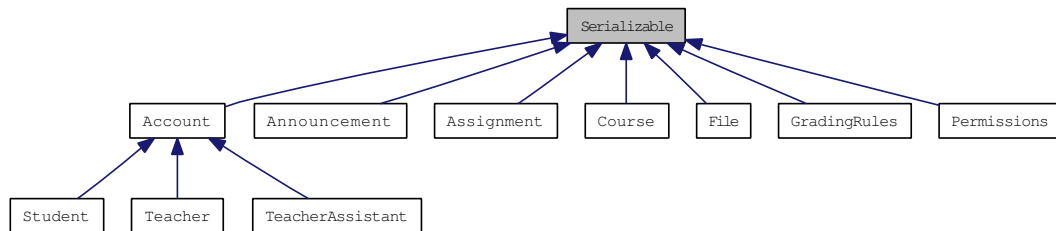
It does nothing to permissions previously not granted.

**Parameters:**

*permission* to determine which permission to be removed

## 3.21 Serializable Class Reference

Inheritance diagram for Serializable:



## 3.22 StorageConnection Class Reference

### 3.22.1 Detailed Description

This is the class that deals with connecting to the storage device.

It could also deal with querying the storage system. Look into Hibernate as a way to abstract this more.

**Author:**

[blugoo](#)

**Version:**

0.42

### Package Functions

- [StorageConnection \(\)](#)  
*Connect and be jolly.*

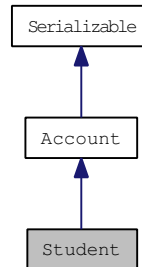
### 3.22.2 Constructor & Destructor Documentation

#### 3.22.2.1 StorageConnection () [package]

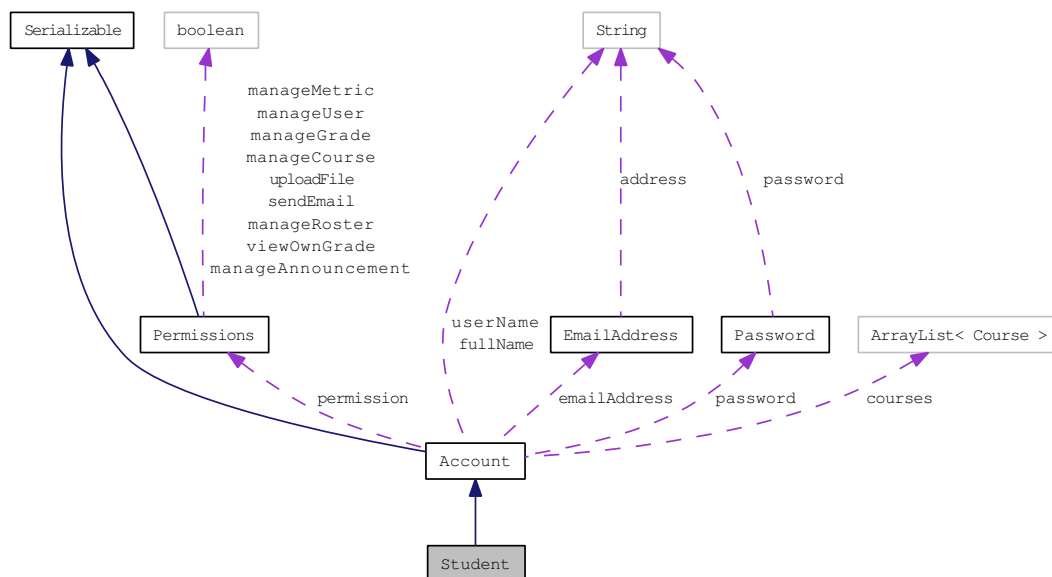
Connect and be jolly.

## 3.23 Student Class Reference

Inheritance diagram for Student:



Collaboration diagram for Student:



### 3.23.1 Detailed Description

This class keeps information about Students.

It is an extension of [Account](#). Students have files.

**Author:**

bluGoo

**Version:**

0.42

### Package Functions

- int [GetGrade](#) (String course)

*Gets the student grade in the course.*

- `int` [getGrade](#) (`String` *course*, `String` *assignment*)
- `void` [uploadFile](#) (`String` *assignment*, `String` *course*)

### 3.23.2 Member Function Documentation

#### 3.23.2.1 `int GetGrade (String course)` [package]

Gets the student grade in the course.

**Returns:**

percentage the student has in the course

**Parameters:**

*course* A string which matches a course name, or other identifier.

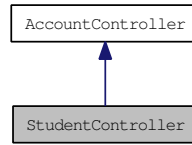
#### 3.23.2.2 `int getGrade (String course, String assignment)` [package]

#### 3.23.2.3 `void uploadFile (String assignment, String course)` [package]

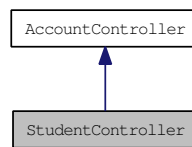


## 3.24 StudentController Class Reference

Inheritance diagram for StudentController:



Collaboration diagram for StudentController:



### 3.24.1 Detailed Description

This class deals with controlling Students.

It is an extension of [AccountController](#). Students have files.

**Author:**

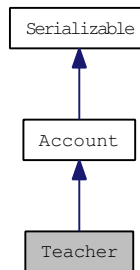
bluGoo

**Version:**

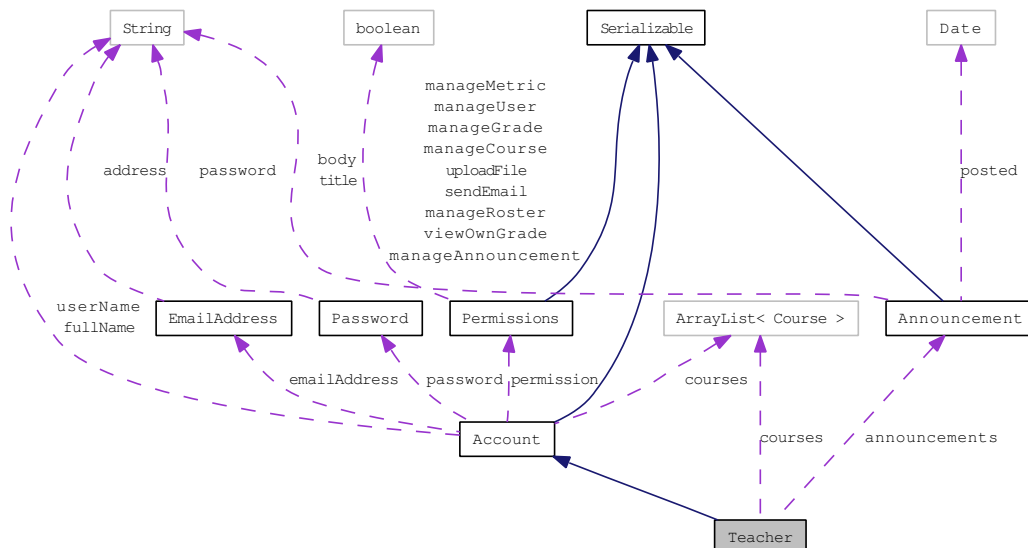
0.42

## 3.25 Teacher Class Reference

Inheritance diagram for Teacher:



Collaboration diagram for Teacher:



### 3.25.1 Detailed Description

A [Teacher](#) is an [Account](#) type that has the most permissions.

It acts as the administrator of its installation. There are a lot of methods in here that really should not be in here. I mean seriously. most of the create, edit, and remove methods SHOULD NOT be in here. It makes me cringe everytime I think about it.

**Author:**

bluGoo

**Version:**

0.42

## Public Member Functions

- [Teacher](#) ()  
*Constructor for [Teacher](#).*
- [Course](#) [createCourse](#) ()  
*[createCourse](#) provides the overhead for creating a new class.*
- [Course](#) [editCourse](#) ([Course](#) course)  
*[editCourse](#) asks for information from the user and updates the [Course](#) with it*
- boolean [removeCourse](#) ([Course](#) course)  
*[removeCourse](#) marks a course for deletion.*
- [Announcement](#) [createAnnouncement](#) ([Course](#) which)  
*[createAnnouncement](#) gathers information from the user to make a new [Announcement](#) with*
- boolean [gradeStudent](#) ([Student](#) who)  
*[gradeStudent](#) adjusts information in a [Student](#) specified by the user*
- [Account](#) [createUser](#) ()  
*[createUser](#) creates a new [Account](#) based on the user's preferences.*
- [Account](#) [removeUser](#) ([Account](#) user)  
*[removeUser](#) removes a User from a [Course](#).*
- [Account](#) [editUser](#) ()  
*[editUser](#) provides overhead for [Teacher](#) to edit User information*
- [Assignment](#) [createAssignment](#) ()  
*[createAssignment](#) creates a new [Assignment](#) with the Teacher's specifications.*
- [Assignment](#) [editAssignment](#) ()  
*[editAssignment](#) changes the information in the [Assignment](#) to fit the Teacher's preferences.*
- [Assignment](#) [deleteAssignment](#) ()  
*[deleteAssignment](#) marks an [Assignment](#) for removal from the system*
- [Email](#) [sendEmail](#) ()  
*[SendEmail](#) creates a new [Email](#) and send it to an address specified by the [Teacher](#).*
- [GradingRules](#) [adjustCurve](#) ()  
*[adjustCruve](#) modifies the [GradingRules](#) to a specific [Course](#)*
- boolean [TakeAttendance](#) ()  
*[takeAttendance](#) adjusts the Participation portion of a grade for Students*

## Private Attributes

- [Announcement](#) `announcements []`

*An array of [Announcements](#) owned by the [Teacher](#).*

- `ArrayList< Course > courses`

*A list of [courses](#) owned by the [Teacher](#).*

## 3.25.2 Constructor & Destructor Documentation

### 3.25.2.1 [Teacher](#) ()

Constructor for [Teacher](#).

All permissions are granted to the [Teacher](#) by default.

## 3.25.3 Member Function Documentation

### 3.25.3.1 [Course](#) `createCourse ()`

`createCourse` provides the overhead for creating a new class.

The method prompts the user for information.

#### Returns:

a new course as specified by the [Teacher](#)'s input.

### 3.25.3.2 [Course](#) `editCourse (Course course)`

`editCourse` asks for information from the user and updates the [Course](#) with it

#### Returns:

the new, edited [Course](#)

### 3.25.3.3 `boolean` `removeCourse (Course course)`

`removeCourse` marks a course for deletion.

#### Returns:

true if no errors occur

### 3.25.3.4 [Announcement](#) `createAnnouncement (Course which)`

`createAnnouncement` gathers information from the user to make a new [Announcement](#) with

**Parameters:**

*which* The course to add an announcement for.

**Returns:**

the new, created [Announcement](#).

**3.25.3.5 boolean gradeStudent (Student *who*)**

gradeStudent adjusts information in a [Student](#) specified by the user

**Parameters:**

*who* the [Student](#) to be graded.

**3.25.3.6 Account createUser ()**

createUser creates a new [Account](#) based on the user's preferences.

If a [Teacher](#) Assistant is being created, their permissions may be given. The new account will be stored in userStorage until they are given a [Course](#), if none specified.

**Returns:**

the created [Account](#)

**3.25.3.7 Account removeUser (Account *user*)**

removeUser removes a User from a [Course](#).

If the user is not on any remaining courses, the account is marked for deletion

**Returns:**

the [Account](#) removed

**3.25.3.8 Account editUser ()**

editUser provides overhead for [Teacher](#) to edit User information

**Returns:**

the [Account](#) that was edited

**3.25.3.9 Assignment createAssignment ()**

createAssignment creates a new [Assignment](#) with the Teacher's specifications.

**Returns:**

the new [Assignment](#)

**3.25.3.10 Assignment editAssignment ()**

editAssignment changes the information in the [Assignment](#) to fit the Teacher's preferences.

**Returns:**

the modified [Assignment](#)

**3.25.3.11 Assignment deleteAssignment ()**

deleteAssignment marks an [Assignment](#) for removal from the system

**Returns:**

the [Assignment](#) deleted

**3.25.3.12 Email sendEmail ()**

SendEmail creates a new [Email](#) and send it to an address specified by the [Teacher](#).

**Returns:**

the sent email

**3.25.3.13 GradingRules adjustCurve ()**

adjustCruve modifies the [GradingRules](#) to a specific [Course](#)

**Returns:**

the [GradingRules](#) that were modified

**3.25.3.14 boolean TakeAttendance ()**

takeAttendance adjusts the Participation portion of a grade for Students

**Returns:**

true if no errors occurred

**3.25.4 Member Data Documentation****3.25.4.1 Announcement announcements[] [private]**

An array of Announcements owned by the [Teacher](#).

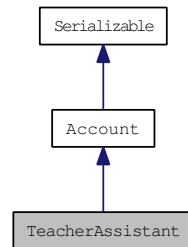
**3.25.4.2 ArrayList<Course> courses [private]**

A list of courses owned by the [Teacher](#).

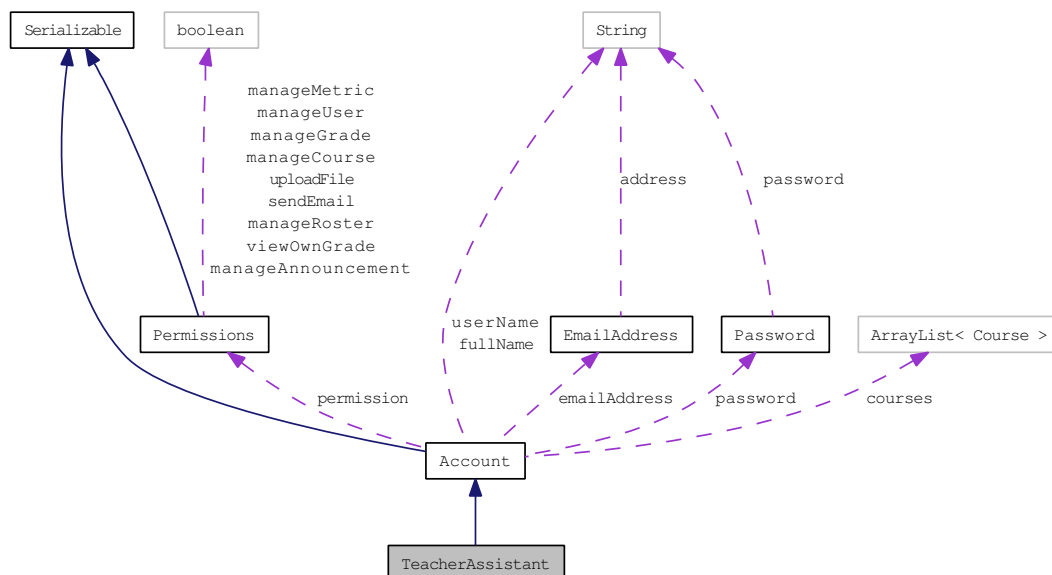
Reimplemented from [Account](#).

## 3.26 TeacherAssistant Class Reference

Inheritance diagram for TeacherAssistant:



Collaboration diagram for TeacherAssistant:



### 3.26.1 Detailed Description

This class keeps information about [Teacher](#) Assistants.

It is an extension of [Account](#).

#### Author:

bluGoo

#### Version:

0.42

### Public Member Functions

- [TeacherAssistant](#) ([Permissions](#) granted)

*Constructor for [Teacher](#) Assistant.*

- [TeacherAssistant](#) ()

## 3.26.2 Constructor & Destructor Documentation

### 3.26.2.1 TeacherAssistant (Permissions *granted*)

Constructor for [Teacher](#) Assistant.

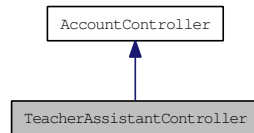
If no permissions are passed in, a default, minimal permission set will be initiated.

### 3.26.2.2 TeacherAssistant ()

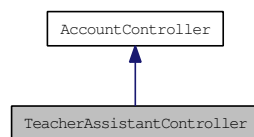


## 3.27 TeacherAssistantController Class Reference

Inheritance diagram for TeacherAssistantController:



Collaboration diagram for TeacherAssistantController:



### 3.27.1 Detailed Description

This class controls [Teacher](#) Assistants.

It is an extension of [AccountController](#).

**Author:**

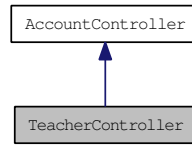
bluGoo

**Version:**

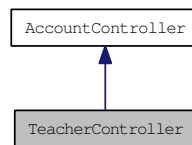
0.42

## 3.28 TeacherController Class Reference

Inheritance diagram for TeacherController:



Collaboration diagram for TeacherController:



### 3.28.1 Detailed Description

This is a controller for teacher.

**Author:**

bluGoo

**Version:**

0.42

### Public Member Functions

- [Course](#) `createCourse ()`  
*createCourse provides the overhead for creating a new class.*
- [Course](#) `editCourse (Course course)`  
*editCourse asks for information from the user and updates the [Course](#) with it*
- boolean `removeCourse (Course course)`  
*removeCourse marks a course for deletion.*
- [Announcement](#) `createAnnouncement (Course which)`  
*createAnnouncement gathers information from the user to make a new [Announcement](#) with*
- boolean `gradeStudent (Student who)`  
*gradeStudent adjusts information in a [Student](#) specified by the user*
- [Account](#) `createUser ()`  
*createUser creates a new [Account](#) based on the user's preferences.*

- [Account removeUser](#) ([Account](#) user)  
*removeUser* removes a [User](#) from a [Course](#).
- [Account editUser](#) ()  
*editUser* provides overhead for [Teacher](#) to edit [User](#) information
- [Assignment createAssignment](#) ()  
*createAssignment* creates a new [Assignment](#) with the [Teacher](#)'s specifications.
- [Assignment editAssignment](#) ()  
*editAssignment* changes the information in the [Assignment](#) to fit the [Teacher](#)'s preferences.
- [Assignment deleteAssignment](#) ()  
*deleteAssignment* marks an [Assignment](#) for removal from the system
- [GradingRules adjustCurve](#) ()  
*adjustCurve* modifies the [GradingRules](#) to a specific [Course](#)
- boolean [takeAttendance](#) ()  
*takeAttendance* adjusts the [Participation](#) portion of a grade for [Students](#)

## 3.28.2 Member Function Documentation

### 3.28.2.1 Course createCourse ()

*createCourse* provides the overhead for creating a new class.

The method prompts the user for information.

**Returns:**

a new course as specified by the [Teacher](#)'s input.

### 3.28.2.2 Course editCourse (Course course)

*editCourse* asks for information from the user and updates the [Course](#) with it

**Returns:**

the new, edited [Course](#)

### 3.28.2.3 boolean removeCourse (Course course)

*removeCourse* marks a course for deletion.

**Returns:**

true if no errors occur

#### 3.28.2.4 Announcement createAnnouncement (Course *which*)

createAnnouncement gathers information from the user to make a new [Announcement](#) with

**Parameters:**

*which* The course to add an announcement for.

**Returns:**

the new, created [Announcement](#).

#### 3.28.2.5 boolean gradeStudent (Student *who*)

gradeStudent adjusts information in a [Student](#) specified by the user

**Parameters:**

*who* the [Student](#) to be graded.

#### 3.28.2.6 Account createUser ()

createUser creates a new [Account](#) based on the user's preferences.

If a [Teacher](#) Assistant is being created, their permissions may be given. The new account will be stored in userStorage until they are given a [Course](#), if none specified.

**Returns:**

the created [Account](#)

#### 3.28.2.7 Account removeUser (Account *user*)

removeUser removes a User from a [Course](#).

If the user is not on any remaining courses, the account is marked for deletion

**Returns:**

the [Account](#) removed

#### 3.28.2.8 Account editUser ()

editUser provides overhead for [Teacher](#) to edit User information

**Returns:**

the [Account](#) that was edited

**3.28.2.9 Assignment createAssignment ()**

createAssignment creates a new [Assignment](#) with the Teacher's specifications.

**Returns:**

the new [Assignment](#)

**3.28.2.10 Assignment editAssignment ()**

editAssignment changes the information in the [Assignment](#) to fit the Teacher's preferences.

**Returns:**

the modified [Assignment](#)

**3.28.2.11 Assignment deleteAssignment ()**

deleteAssignment marks an [Assignment](#) for removal from the system

**Returns:**

the [Assignment](#) deleted

**3.28.2.12 GradingRules adjustCurve ()**

adjustCurve modifies the [GradingRules](#) to a specific [Course](#)

**Returns:**

the [GradingRules](#) that were modified

**3.28.2.13 boolean takeAttendance ()**

takeAttendance adjusts the Participation portion of a grade for Students

**Returns:**

true if no errors occurred



## Chapter 4

# File Documentation

### 4.1 blugoo/WEB-INF/src/java/blugoo/Account.java File Reference

#### Namespaces

- namespace [blugoo](#)

#### Classes

- class [Account](#)

*This is a class that keeps all of the information of a user.*

## 4.2 blugoo/WEB-INF/src/java/blugoo/AccountController.java File Reference

### Namespaces

- namespace [blugoo](#)

### Classes

- class [AccountController](#)

*This class does most of the work on the [Account](#).*



## 4.3 blugoo/WEB-INF/src/java/blugoo/Announcement.java File Reference

### Namespaces

- namespace [blugoo](#)

### Classes

- class [Announcement](#)  
*An [Announcement](#) for a [Course](#).*

## 4.4 blugoo/WEB-INF/src/java/blugoo/AnnouncementController.java

### File Reference

#### Namespaces

- namespace [blugoo](#)

#### Classes

- class [AnnouncementController](#)  
*This is the controller for [Announcement](#).*

## 4.5 blugoo/WEB-INF/src/java/blugoo/Assignment.java File Reference

### Namespaces

- namespace [blugoo](#)

### Classes

- class [Assignment](#)

*This class keeps information about Assignments.*

## 4.6 blugoo/WEB-INF/src/java/blugoo/AssignmentController.java

### File Reference

#### Namespaces

- namespace [blugoo](#)

#### Classes

- class [AssignmentController](#)

*Does all of the work on the [Assignment](#), it is the controller.*

## 4.7 blugoo/WEB-INF/src/java/blugoo/Course.java File Reference

### Namespaces

- namespace [blugoo](#)

### Classes

- class [Course](#)

*This class keeps information about Courses.*

## 4.8 blugoo/WEB-INF/src/java/blugoo/CourseController.java    File Reference

### Namespaces

- namespace [blugoo](#)

### Classes

- class [CourseController](#)

*Methods for modifying a course and it's related actions.*

## 4.9 blugoo/WEB-INF/src/java/blugoo/Email.java File Reference

### Namespaces

- namespace [blugoo](#)

### Classes

- class [Email](#)

*The class stores information about Emails.*

## 4.10 blugoo/WEB-INF/src/java/blugoo/EmailAddress.java File Reference

### Namespaces

- namespace [blugoo](#)

### Classes

- class [EmailAddress](#)

*A model for email addresses since they need to be validated and such.*



## 4.11 blugoo/WEB-INF/src/java/blugoo/EmailController.java File Reference

### Namespaces

- namespace [blugoo](#)

### Classes

- class [EmailController](#)  
*This deals with sending emails.*

## 4.12 blugoo/WEB-INF/src/java/blugoo/File.java File Reference

### Namespaces

- namespace [blugoo](#)

### Classes

- class [File](#)

*This class keeps information about Files.*

## 4.13 blugoo/WEB-INF/src/java/blugoo/FileController.java File Reference

### Namespaces

- namespace [blugoo](#)

### Classes

- class [FileController](#)

*This class is to deal with files.*

## 4.14 blugoo/WEB-INF/src/java/blugoo/GradingRules.java File Reference

### Namespaces

- namespace [blugoo](#)

### Classes

- class [GradingRules](#)

*The [GradingRules](#) class determines how the Total Grade is computed from mutiple [Assignment](#) Grades.*

## 4.15 blugoo/WEB-INF/src/java/blugoo/GradingRulesController.java File Reference

### Namespaces

- namespace [blugoo](#)

### Classes

- class [GradingRulesController](#)

*The [GradingRulesController](#) class is the [GradingRules](#) controller.*

## 4.16 blugoo/WEB-INF/src/java/blugoo/Login.java File Reference

### Namespaces

- namespace [blugoo](#)

### Classes

- class [Login](#)

*This should deal with login I guess.*

## 4.17 blugoo/WEB-INF/src/java/blugoo/Password.java File Reference

### Namespaces

- namespace [blugoo](#)

### Classes

- class [Password](#)

*A model for passwords since they need to be validated and such.*

## 4.18 blugoo/WEB-INF/src/java/blugoo/PasswordController.java

### File Reference

#### Namespaces

- namespace [blugoo](#)

#### Classes

- class [PasswordController](#)

*A controller for passwords since they need to be validated and such.*



## 4.19 blugoo/WEB-INF/src/java/blugoo/Permissions.java File Reference

### Namespaces

- namespace [blugoo](#)

### Classes

- class [Permissions](#)

*This class keeps information about permissions for each action.*

## 4.20 blugoo/WEB-INF/src/java/blugoo/PermissionsController.java

### File Reference

#### Namespaces

- namespace [blugoo](#)

#### Classes

- class [PermissionsController](#)

*This class keeps the permissions controller.*

## 4.21 blugoo/WEB-INF/src/java/blugoo/StorageConnection.java File Reference

### Namespaces

- namespace [blugoo](#)

### Classes

- class [StorageConnection](#)

*This is the class that deals with connecting to the storage device.*

## 4.22 blugoo/WEB-INF/src/java/blugoo/Student.java File Reference

### Namespaces

- namespace [blugoo](#)

### Classes

- class [Student](#)

*This class keeps information about Students.*

## 4.23 blugoo/WEB-INF/src/java/blugoo/StudentController.java File Reference

### Namespaces

- namespace [blugoo](#)

### Classes

- class [StudentController](#)

*This class deals with controlling Students.*

## 4.24 blugoo/WEB-INF/src/java/blugoo/Teacher.java File Reference

### Namespaces

- namespace [blugoo](#)

### Classes

- class [Teacher](#)  
*A [Teacher](#) is an [Account](#) type that has the most permissions.*

## 4.25 blugoo/WEB-INF/src/java/blugoo/TeacherAssistant.java File Reference

### Namespaces

- namespace [blugoo](#)

### Classes

- class [TeacherAssistant](#)

*This class keeps information about [Teacher](#) Assistants.*

## 4.26 blugoo/WEB-INF/src/java/blugoo/TeacherAssistantController.java

### File Reference

#### Namespaces

- namespace [blugoo](#)

#### Classes

- class [TeacherAssistantController](#)  
*This class controls [Teacher](#) Assistants.*



## 4.27 blugoo/WEB-INF/src/java/blugoo/TeacherController.java File Reference

### Namespaces

- namespace [blugoo](#)

### Classes

- class [TeacherController](#)

*This is a controller for teacher.*

## 4.28 front.txt File Reference

# Index

- AddPermission
  - blugoo::Permissions, 42
  - blugoo::PermissionsController, 45
- address
  - blugoo::EmailAddress, 28
- addStudent
  - blugoo::CourseController, 23
- addTA
  - blugoo::CourseController, 24
- adjustCurve
  - blugoo::Teacher, 56
  - blugoo::TeacherController, 63
- aFloor
  - blugoo::GradingRules, 34
- announcements
  - blugoo::Teacher, 56
- assistant
  - blugoo::Course, 22
- average
  - blugoo::Assignment, 16
- bFloor
  - blugoo::GradingRules, 34
- blugoo, 3
- blugoo/WEB-INF/src/java/blugoo/Account.java, 65
- blugoo/WEB-INF/src/java/blugoo/AccountController.java, 66
- blugoo/WEB-INF/src/java/blugoo/Announcement.java, 67
- blugoo/WEB-INF/src/java/blugoo/AnnouncementController.java, 68
- blugoo/WEB-INF/src/java/blugoo/Assignment.java, 69
- blugoo/WEB-INF/src/java/blugoo/AssignmentController.java, 70
- blugoo/WEB-INF/src/java/blugoo/Course.java, 71
- blugoo/WEB-INF/src/java/blugoo/CourseController.java, 72
- blugoo/WEB-INF/src/java/blugoo/Email.java, 73
- blugoo/WEB-INF/src/java/blugoo/EmailAddress.java, 74
- blugoo/WEB-INF/src/java/blugoo/EmailController.java, 75
- blugoo/WEB-INF/src/java/blugoo/File.java, 76
- blugoo/WEB-INF/src/java/blugoo/FileController.java, 77
- blugoo/WEB-INF/src/java/blugoo/GradingRules.java, 78
- blugoo/WEB-INF/src/java/blugoo/GradingRulesController.java, 79
- blugoo/WEB-INF/src/java/blugoo/Login.java, 80
- blugoo/WEB-INF/src/java/blugoo/Password.java, 81
- blugoo/WEB-INF/src/java/blugoo/PasswordController.java, 82
- blugoo/WEB-INF/src/java/blugoo/Permissions.java, 83
- blugoo/WEB-INF/src/java/blugoo/PermissionsController.java, 84
- blugoo/WEB-INF/src/java/blugoo/StorageConnection.java, 85
- blugoo/WEB-INF/src/java/blugoo/Student.java, 86
- blugoo/WEB-INF/src/java/blugoo/StudentController.java, 87
- blugoo/WEB-INF/src/java/blugoo/Teacher.java, 88
- blugoo/WEB-INF/src/java/blugoo/TeacherAssistant.java, 89
- blugoo/WEB-INF/src/java/blugoo/TeacherAssistantController.java, 90
- blugoo/WEB-INF/src/java/blugoo/TeacherController.java, 91
- blugoo::Account, 5
  - courses, 9
  - emailAddress, 8
  - fullName, 8
  - getEmailAddress, 7
  - getFullName, 7
  - getUserName, 7
  - password, 8
  - permission, 9
  - setEmailAddress, 8
  - setFullName, 8
  - setPassword, 7
  - setUserName, 7
  - userName, 8
- blugoo::AccountController, 10
- blugoo::Announcement, 11
  - body, 13

- getBody, 12
  - getDate, 12
  - getTitle, 12
  - posted, 13
  - setBody, 12
  - setDate, 12
  - setTitle, 13
  - title, 13
- blugoo::AnnouncementController, 14
  - post, 14
- blugoo::Assignment, 15
  - average, 16
  - dueDate, 16
  - getAvg, 16
  - getDueDate, 16
  - getMax, 16
  - getMin, 16
  - getName, 16
  - getTotal, 16
  - getType, 16
  - max, 17
  - min, 17
  - name, 16
  - total, 16
  - type, 16
- blugoo::AssignmentController, 18
  - buildMetrics, 18
- blugoo::Course, 19
  - assistant, 22
  - Course, 21
  - courseNumber, 22
  - courseSection, 22
  - courseTitle, 22
  - getTeacher, 21
  - gradingRules, 22
  - scale, 22
  - setNumber, 21
  - setSection, 21
  - setTeacher, 21
  - setTitle, 21
  - students, 22
  - teacher, 22
- blugoo::CourseController, 23
  - addStudent, 23
  - addTA, 24
  - removeStudent, 23
  - removeTA, 24
- blugoo::Email, 25
  - Email, 26
  - emailAttachment, 26
  - emailBody, 26
  - emailRecipients, 26
  - emailTitle, 26
- blugoo::EmailAddress, 27
  - address, 28
  - toString, 27
  - valid, 27
- blugoo::EmailController, 29
  - send, 29
- blugoo::File, 30
  - creationDate, 30
  - fileName, 31
  - owner, 30
- blugoo::FileController, 32
  - download, 32
  - upload, 32
- blugoo::GradingRules, 33
  - aFloor, 34
  - bFloor, 34
  - cFloor, 34
  - dFloor, 34
  - getCurve, 34
  - setCurve, 34
- blugoo::GradingRulesController, 36
- blugoo::Login, 37
  - execute, 37
  - getPassword, 38
  - getUsername, 38
  - isInvalid, 37
  - password, 38
  - setPassword, 38
  - setUsername, 38
  - username, 38
- blugoo::Password, 39
  - Password, 39
  - password, 39
- blugoo::PasswordController, 40
- blugoo::Permissions, 41
  - AddPermission, 42
  - manageAnnouncement, 43
  - manageCourse, 43
  - manageGrade, 43
  - manageMetric, 43
  - manageRoster, 43
  - manageUser, 44
  - Permissions, 42
  - RemovePermission, 43
  - sendEmail, 43
  - uploadFile, 43
  - VerifyPermission, 42
  - viewOwnGrade, 43
- blugoo::PermissionsController, 45
  - AddPermission, 45
  - RemovePermission, 45
  - VerifyPermission, 45
- blugoo::StorageConnection, 48
  - StorageConnection, 48
- blugoo::Student, 49

- GetGrade, 50
- getGrade, 50
- uploadFile, 50
- blugoo::StudentController, 51
- blugoo::Teacher, 52
  - adjustCurve, 56
  - announcements, 56
  - courses, 56
  - createAnnouncement, 54
  - createAssignment, 55
  - createCourse, 54
  - createUser, 55
  - deleteAssignment, 56
  - editAssignment, 55
  - editCourse, 54
  - editUser, 55
  - gradeStudent, 55
  - removeCourse, 54
  - removeUser, 55
  - sendEmail, 56
  - TakeAttendance, 56
  - Teacher, 54
- blugoo::TeacherAssistant, 57
  - TeacherAssistant, 58
- blugoo::TeacherAssistantController, 59
- blugoo::TeacherController, 60
  - adjustCurve, 63
  - createAnnouncement, 61
  - createAssignment, 62
  - createCourse, 61
  - createUser, 62
  - deleteAssignment, 63
  - editAssignment, 63
  - editCourse, 61
  - editUser, 62
  - gradeStudent, 62
  - removeCourse, 61
  - removeUser, 62
  - takeAttendance, 63
- body
  - blugoo::Announcement, 13
- buildMetrics
  - blugoo::AssignmentController, 18
- cFloor
  - blugoo::GradingRules, 34
- Course
  - blugoo::Course, 21
- courseNumber
  - blugoo::Course, 22
- courses
  - blugoo::Account, 9
  - blugoo::Teacher, 56
- courseSection
  - blugoo::Course, 22
- courseTitle
  - blugoo::Course, 22
- createAnnouncement
  - blugoo::Teacher, 54
  - blugoo::TeacherController, 61
- createAssignment
  - blugoo::Teacher, 55
  - blugoo::TeacherController, 62
- createCourse
  - blugoo::Teacher, 54
  - blugoo::TeacherController, 61
- createUser
  - blugoo::Teacher, 55
  - blugoo::TeacherController, 62
- creationDate
  - blugoo::File, 30
- deleteAssignment
  - blugoo::Teacher, 56
  - blugoo::TeacherController, 63
- dFloor
  - blugoo::GradingRules, 34
- download
  - blugoo::FileController, 32
- dueDate
  - blugoo::Assignment, 16
- editAssignment
  - blugoo::Teacher, 55
  - blugoo::TeacherController, 63
- editCourse
  - blugoo::Teacher, 54
  - blugoo::TeacherController, 61
- editUser
  - blugoo::Teacher, 55
  - blugoo::TeacherController, 62
- Email
  - blugoo::Email, 26
- emailAddress
  - blugoo::Account, 8
- emailAttachment
  - blugoo::Email, 26
- emailBody
  - blugoo::Email, 26
- emailRecipients
  - blugoo::Email, 26
- emailTitle
  - blugoo::Email, 26
- execute
  - blugoo::Login, 37
- fileName
  - blugoo::File, 31

- front.txt, [92](#)
- fullName
  - blugoo::Account, [8](#)
- getAvg
  - blugoo::Assignment, [16](#)
- getBody
  - blugoo::Announcement, [12](#)
- getCurve
  - blugoo::GradingRules, [34](#)
- getDate
  - blugoo::Announcement, [12](#)
- getDueDate
  - blugoo::Assignment, [16](#)
- getEmailAddress
  - blugoo::Account, [7](#)
- getFullName
  - blugoo::Account, [7](#)
- GetGrade
  - blugoo::Student, [50](#)
- getGrade
  - blugoo::Student, [50](#)
- getMax
  - blugoo::Assignment, [16](#)
- getMin
  - blugoo::Assignment, [16](#)
- getName
  - blugoo::Assignment, [16](#)
- getPassword
  - blugoo::Login, [38](#)
- getTeacher
  - blugoo::Course, [21](#)
- getTitle
  - blugoo::Announcement, [12](#)
- getTotal
  - blugoo::Assignment, [16](#)
- getType
  - blugoo::Assignment, [16](#)
- getUserName
  - blugoo::Account, [7](#)
- getUsername
  - blugoo::Login, [38](#)
- gradeStudent
  - blugoo::Teacher, [55](#)
  - blugoo::TeacherController, [62](#)
- gradingRules
  - blugoo::Course, [22](#)
- isInvalid
  - blugoo::Login, [37](#)
- isPassword
  - blugoo::Account, [8](#)
- java::io::Serializable, [47](#)
- manageAnnouncement
  - blugoo::Permissions, [43](#)
- manageCourse
  - blugoo::Permissions, [43](#)
- manageGrade
  - blugoo::Permissions, [43](#)
- manageMetric
  - blugoo::Permissions, [43](#)
- manageRoster
  - blugoo::Permissions, [43](#)
- manageUser
  - blugoo::Permissions, [44](#)
- max
  - blugoo::Assignment, [17](#)
- min
  - blugoo::Assignment, [17](#)
- name
  - blugoo::Assignment, [16](#)
- owner
  - blugoo::File, [30](#)
- Password
  - blugoo::Password, [39](#)
- password
  - blugoo::Account, [8](#)
  - blugoo::Login, [38](#)
  - blugoo::Password, [39](#)
- permission
  - blugoo::Account, [9](#)
- Permissions
  - blugoo::Permissions, [42](#)
- post
  - blugoo::AnnouncementController, [14](#)
- posted
  - blugoo::Announcement, [13](#)
- removeCourse
  - blugoo::Teacher, [54](#)
  - blugoo::TeacherController, [61](#)
- RemovePermission
  - blugoo::Permissions, [43](#)
  - blugoo::PermissionsController, [45](#)
- removeStudent
  - blugoo::CourseController, [23](#)
- removeTA
  - blugoo::CourseController, [24](#)
- removeUser
  - blugoo::Teacher, [55](#)
  - blugoo::TeacherController, [62](#)
- scale
  - blugoo::Course, [22](#)
- send

- blugoo::EmailController, 29
- sendEmail
  - blugoo::Permissions, 43
  - blugoo::Teacher, 56
- setBody
  - blugoo::Announcement, 12
- setCurve
  - blugoo::GradingRules, 34
- setDate
  - blugoo::Announcement, 12
- setEmailAddress
  - blugoo::Account, 8
- setFullName
  - blugoo::Account, 8
- setNumber
  - blugoo::Course, 21
- setPassword
  - blugoo::Account, 7
  - blugoo::Login, 38
- setSection
  - blugoo::Course, 21
- setTeacher
  - blugoo::Course, 21
- setTitle
  - blugoo::Announcement, 13
  - blugoo::Course, 21
- setUserName
  - blugoo::Account, 7
- setUsername
  - blugoo::Login, 38
- StorageConnection
  - blugoo::StorageConnection, 48
- students
  - blugoo::Course, 22
- TakeAttendance
  - blugoo::Teacher, 56
- takeAttendance
  - blugoo::TeacherController, 63
- Teacher
  - blugoo::Teacher, 54
- teacher
  - blugoo::Course, 22
- TeacherAssistant
  - blugoo::TeacherAssistant, 58
- title
  - blugoo::Announcement, 13
- toString
  - blugoo::EmailAddress, 27
- total
  - blugoo::Assignment, 16
- type
  - blugoo::Assignment, 16
- upload
  - blugoo::FileController, 32
- uploadFile
  - blugoo::Permissions, 43
  - blugoo::Student, 50
- userName
  - blugoo::Account, 8
- username
  - blugoo::Login, 38
- valid
  - blugoo::EmailAddress, 27
- VerifyPermission
  - blugoo::Permissions, 42
  - blugoo::PermissionsController, 45
- viewOwnGrade
  - blugoo::Permissions, 43