

Student Grading System

Generated by Doxygen 1.13.2

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 edu.nu.sgm.controllers.AddCourseController Class Reference	7
4.1.1 Detailed Description	7
4.1.2 Member Function Documentation	7
4.1.2.1 addCourse()	7
4.1.2.2 setDialog()	8
4.2 edu.nu.sgm.controllers.AddStudentController Class Reference	8
4.2.1 Detailed Description	8
4.2.2 Member Function Documentation	8
4.2.2.1 setDialog()	8
4.2.2.2 setStudentData()	8
4.3 edu.nu.sgm.models.Course Class Reference	9
4.3.1 Detailed Description	9
4.3.2 Constructor & Destructor Documentation	10
4.3.2.1 Course()	10
4.3.3 Member Function Documentation	10
4.3.3.1 getCourseCode()	10
4.3.3.2 getCreditHours()	10
4.3.3.3 getId()	10
4.3.3.4 getInstructor()	11
4.3.3.5 getTitle()	11
4.3.3.6 getTotalStudents()	11
4.3.3.7 setCourseCode()	11
4.3.3.8 setCreditHours()	11
4.3.3.9 setId()	12
4.3.3.10 setInstructor()	12
4.3.3.11 setTitle()	12
4.3.3.12 setTotalStudents()	13
4.4 edu.nu.sgm.CourseCreationTest Class Reference	13
4.5 edu.nu.sgm.services.CourseService Class Reference	13
4.5.1 Detailed Description	14
4.5.2 Member Function Documentation	14
4.5.2.1 addCourse()	14
4.5.2.2 courseExists()	14

4.5.2.3 displayDetails()	15
4.5.2.4 exportCourses()	15
4.5.2.5 getAllCourses()	15
4.5.2.6 getCourseById()	15
4.5.2.7 getStudents()	16
4.5.2.8 getTotalStudents()	16
4.5.2.9 importCourses()	16
4.5.2.10 removeCourse()	17
4.6 edu.nu.sgm.controllers.CourseViewController Class Reference	17
4.6.1 Detailed Description	18
4.6.2 Member Function Documentation	18
4.6.2.1 setCourse()	18
4.7 edu.nu.sgm.utils.DatabaseManager Class Reference	18
4.7.1 Member Function Documentation	19
4.7.1.1 createCourse()	19
4.7.1.2 createEnrollment()	19
4.7.1.3 createGrade()	19
4.7.1.4 createStudent()	19
4.7.1.5 deleteCourse()	20
4.7.1.6 deleteEnrollment() [1/3]	20
4.7.1.7 deleteEnrollment() [2/3]	20
4.7.1.8 deleteEnrollment() [3/3]	21
4.7.1.9 deleteGrade()	21
4.7.1.10 deleteStudent()	21
4.7.1.11 fetchCourse()	22
4.7.1.12 fetchCourses() [1/2]	22
4.7.1.13 fetchCourses() [2/2]	22
4.7.1.14 fetchEnrollment() [1/3]	22
4.7.1.15 fetchEnrollment() [2/3]	23
4.7.1.16 fetchEnrollment() [3/3]	23
4.7.1.17 fetchGrades()	23
4.7.1.18 fetchStudents() [1/2]	24
4.7.1.19 fetchStudents() [2/2]	24
4.7.1.20 updateCourse()	24
4.7.1.21 updateGrades()	24
4.7.1.22 updateStudent()	25
4.8 edu.nu.sgm.DatabaseManagerTest Class Reference	25
4.9 edu.nu.sgm.controllers.EnrollCourse Class Reference	25
4.10 edu.nu.sgm.models.Enrollment Class Reference	26
4.10.1 Detailed Description	26
4.10.2 Constructor & Destructor Documentation	26
4.10.2.1 Enrollment()	26

4.10.3 Member Function Documentation	26
4.10.3.1 <code>getCourse()</code>	26
4.10.3.2 <code>getId()</code>	27
4.10.3.3 <code>getStudent()</code>	27
4.10.3.4 <code>setId()</code>	27
4.11 <code>edu.nu.sgm.services.EnrollmentService</code> Class Reference	27
4.11.1 Detailed Description	28
4.11.2 Member Function Documentation	28
4.11.2.1 <code>dropCourse()</code>	28
4.11.2.2 <code>enrollStudent()</code>	28
4.11.2.3 <code>generateReportCard()</code>	28
4.12 <code>edu.nu.sgm.EnrollmentTest</code> Class Reference	29
4.13 <code>edu.nu.sgm.controllers.EnrollStudents</code> Class Reference	29
4.13.1 Detailed Description	29
4.13.2 Member Function Documentation	29
4.13.2.1 <code>setCourse()</code>	29
4.13.2.2 <code>setDialog()</code>	30
4.14 <code>edu.nu.sgm.models.GradeItem</code> Class Reference	30
4.14.1 Detailed Description	31
4.14.2 Constructor & Destructor Documentation	31
4.14.2.1 <code>GradeItem()</code>	31
4.14.3 Member Function Documentation	31
4.14.3.1 <code>getCategory()</code>	31
4.14.3.2 <code>getFeedback()</code>	32
4.14.3.3 <code>getId()</code>	32
4.14.3.4 <code>getMaxScore()</code>	32
4.14.3.5 <code>getScore()</code>	32
4.14.3.6 <code>getTitle()</code>	32
4.14.3.7 <code>getWeight()</code>	33
4.14.3.8 <code>setCategory()</code>	33
4.14.3.9 <code>setFeedback()</code>	33
4.14.3.10 <code>setId()</code>	33
4.14.3.11 <code>setMaxScore()</code>	34
4.14.3.12 <code>setScore()</code>	34
4.14.3.13 <code>setTitle()</code>	34
4.14.3.14 <code>setWeight()</code>	35
4.15 <code>edu.nu.sgm.services.GradeItemService</code> Class Reference	35
4.15.1 Detailed Description	36
4.15.2 Member Function Documentation	36
4.15.2.1 <code>addGradeItem()</code>	36
4.15.2.2 <code>calculateTotalGrade()</code>	36
4.15.2.3 <code>exportGradeItems()</code>	36

4.15.2.4	getGrades()	37
4.15.2.5	impoGradelItems()	37
4.15.2.6	removeGradelItem()	37
4.16	edu.nu.sgm.GradelItemServiceTest Class Reference	38
4.17	edu.nu.sgm.Main Class Reference	38
4.18	edu.nu.sgm.controllers.MainViewController Class Reference	39
4.19	edu.nu.sgm.utils.Reader Class Reference	39
4.19.1	Detailed Description	39
4.19.2	Member Function Documentation	39
4.19.2.1	parseCourseExport()	39
4.19.2.2	parseCourseImport()	40
4.19.2.3	parseGradelItemsExport()	40
4.19.2.4	parseGradelItemsImport()	40
4.19.2.5	parseStudentExport()	41
4.19.2.6	parseStudentImport()	41
4.19.2.7	readCSV()	41
4.19.2.8	writeCSV()	42
4.20	edu.nu.sgm.utils.DatabaseManager.ResultSetMapper< T > Interface Template Reference	42
4.21	edu.nu.sgm.models.Student Class Reference	43
4.21.1	Detailed Description	43
4.21.2	Constructor & Destructor Documentation	43
4.21.2.1	Student()	43
4.21.3	Member Function Documentation	44
4.21.3.1	generateEmail()	44
4.22	edu.nu.sgm.services.StudentService Class Reference	44
4.22.1	Detailed Description	44
4.22.2	Member Function Documentation	44
4.22.2.1	addstudent()	44
4.22.2.2	displayDetails()	45
4.22.2.3	getCourses()	45
4.22.2.4	getStudentById()	45
4.22.2.5	getStudents()	46
4.22.2.6	getTotalCourses()	46
4.22.2.7	removeStudent()	46
4.22.2.8	updateStudent()	46
4.23	edu.nu.sgm.controllers.StudentViewController Class Reference	47
4.23.1	Detailed Description	47
4.23.2	Member Function Documentation	47
4.23.2.1	setStudent()	47
5 File Documentation		49
5.1	src/main/java/edu/nu/sgm/controllers/AddCourseController.java File Reference	49

5.1.1 Detailed Description	49
5.2 src/main/java/edu/nu/sgm/controllers/AddStudentController.java File Reference	49
5.2.1 Detailed Description	50
5.3 src/main/java/edu/nu/sgm/controllers/CourseViewController.java File Reference	50
5.3.1 Detailed Description	50
5.4 src/main/java/edu/nu/sgm/controllers/EnrollStudents.java File Reference	51
5.4.1 Detailed Description	51
5.5 src/main/java/edu/nu/sgm/controllers/StudentViewController.java File Reference	51
5.5.1 Detailed Description	52
5.6 src/main/java/edu/nu/sgm/models/Course.java File Reference	52
5.6.1 Detailed Description	52
5.7 src/main/java/edu/nu/sgm/models/Enrollment.java File Reference	52
5.7.1 Detailed Description	52
5.8 src/main/java/edu/nu/sgm/models/GradeItem.java File Reference	52
5.8.1 Detailed Description	53
5.9 src/main/java/edu/nu/sgm/models/Student.java File Reference	53
5.9.1 Detailed Description	53
5.10 src/main/java/edu/nu/sgm/services/CourseService.java File Reference	53
5.10.1 Detailed Description	54
5.11 src/main/java/edu/nu/sgm/services/EnrollmentService.java File Reference	54
5.11.1 Detailed Description	54
5.12 src/main/java/edu/nu/sgm/services/GradeItemService.java File Reference	54
5.12.1 Detailed Description	54
5.13 src/main/java/edu/nu/sgm/services/StudentService.java File Reference	55
5.13.1 Detailed Description	55
5.14 src/main/java/edu/nu/sgm/utils/Reader.java File Reference	55
5.14.1 Detailed Description	55
Index	57

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

edu.nu.sgm.controllers.AddCourseController	7
edu.nu.sgm.controllers.AddStudentController	8
Application	
edu.nu.sgm.Main	38
edu.nu.sgm.models.Course	9
edu.nu.sgm.CourseCreationTest	13
edu.nu.sgm.services.CourseService	13
edu.nu.sgm.utils.DatabaseManager	18
edu.nu.sgm.DatabaseManagerTest	25
edu.nu.sgm.controllers.EnrollCourse	25
edu.nu.sgm.models.Enrollment	26
edu.nu.sgm.services.EnrollmentService	27
edu.nu.sgm.EnrollmentTest	29
edu.nu.sgm.controllers.EnrollStudents	29
edu.nu.sgm.models.GradeItem	30
edu.nu.sgm.services.GradeItemService	35
edu.nu.sgm.GradeItemServiceTest	38
Initializable	
edu.nu.sgm.controllers.CourseViewController	17
edu.nu.sgm.controllers.StudentViewController	47
edu.nu.sgm.controllers.MainViewController	39
edu.nu.sgm.utils.Reader	39
edu.nu.sgm.utils.DatabaseManager.ResultSetMapper< T >	42
edu.nu.sgm.models.Student	43
edu.nu.sgm.services.StudentService	44

Chapter 2

Class Index

2.1 Class List

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

edu.nu.sgm.controllers.AddCourseController	7
Handles the logic for adding a new course from the dialog	
edu.nu.sgm.controllers.AddStudentController	8
Handles logic for adding or editing a student from the dialog	
edu.nu.sgm.models.Course	9
Represents a course with its details and enrollment count	
edu.nu.sgm.CourseCreationTest	13
edu.nu.sgm.services.CourseService	13
Provides methods to add, remove, retrieve, and manage courses	
edu.nu.sgm.controllers.CourseViewController	17
Handles the course detail view, including enrolled students	
edu.nu.sgm.utils.DatabaseManager	18
edu.nu.sgm.DatabaseManagerTest	25
edu.nu.sgm.controllers.EnrollCourse	25
edu.nu.sgm.models.Enrollment	26
Represents the enrollment relationship between a student and a course	
edu.nu.sgm.services.EnrollmentService	27
Provides methods to enroll students, drop courses, and generate report cards	
edu.nu.sgm.EnrollmentTest	29
edu.nu.sgm.controllers.EnrollStudents	29
Handles the dialog for enrolling a student in a course	
edu.nu.sgm.models.GradeItem	30
Represents a grade item (e.g., assignment, exam) for a student	
edu.nu.sgm.services.GradeItemService	35
Provides methods to create, retrieve, delete, and process grade items	
edu.nu.sgm.GradeItemServiceTest	38
edu.nu.sgm.Main	38
edu.nu.sgm.controllers.MainViewController	39
edu.nu.sgm.utils.Reader	39
Provides static methods for reading and writing CSV files and parsing/exporting model objects	
edu.nu.sgm.utils.DatabaseManager.ResultSetMapper< T >	42
edu.nu.sgm.models.Student	43
Represents a student with ID, name, and email	
edu.nu.sgm.services.StudentService	44
Provides methods to add, remove, update, and retrieve students and their courses	
edu.nu.sgm.controllers.StudentViewController	47
Handles the student detail view, including enrolled courses	

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

src/main/java/edu/nu/sgm/controllers/ AddCourseController.java	
Controller for adding a new course via the add-course dialog	49
src/main/java/edu/nu/sgm/controllers/ AddStudentController.java	
Controller for adding or editing a student via the add-student dialog	49
src/main/java/edu/nu/sgm/controllers/ CourseViewController.java	
Controller for displaying and managing course details and enrolled students	50
src/main/java/edu/nu/sgm/controllers/ EnrollStudents.java	
Controller for enrolling a student in a course from the course view dialog	51
src/main/java/edu/nu/sgm/controllers/ StudentViewController.java	
Controller for displaying and managing student details and enrolled courses	51
src/main/java/edu/nu/sgm/models/ Course.java	
Represents a course in the student grading system	52
src/main/java/edu/nu/sgm/models/ Enrollment.java	
Represents an enrollment of a student in a course	52
src/main/java/edu/nu/sgm/models/ GradeItem.java	
Represents a single grade item for a student	52
src/main/java/edu/nu/sgm/models/ Student.java	
Represents a student in the grading system	53
src/main/java/edu/nu/sgm/services/ CourseService.java	
Service class for managing courses in the grading system	53
src/main/java/edu/nu/sgm/services/ EnrollmentService.java	
Service class for managing student enrollments in courses	54
src/main/java/edu/nu/sgm/services/ GradeItemService.java	
Service class for managing grade items related to student enrollments	54
src/main/java/edu/nu/sgm/services/ StudentService.java	
Service class for managing students in the grading system	55
src/main/java/edu/nu/sgm/utils/ Reader.java	
Utility class for reading and writing CSV files for courses, students, and grade items	55

Chapter 4

Class Documentation

4.1 edu.nu.sgm.controllers.AddCourseController Class Reference

Handles the logic for adding a new course from the dialog.

Public Member Functions

- void [setDialog](#) (Dialog< ButtonType > dialog)
Sets the dialog reference for this controller.
- boolean [addCourse](#) (String name, String code, String instructor, int credits)
Adds a course using the provided details.

4.1.1 Detailed Description

Handles the logic for adding a new course from the dialog.

4.1.2 Member Function Documentation

4.1.2.1 addCourse()

```
boolean edu.nu.sgm.controllers.AddCourseController.addCourse (  
    String name,  
    String code,  
    String instructor,  
    int credits) [inline]
```

Adds a course using the provided details.

Parameters

<i>name</i>	Course name.
<i>code</i>	Course code.
<i>instructor</i>	Instructor name.
<i>credits</i>	Credit hours.

Returns

true if course was added successfully.

4.1.2.2 setDialog()

```
void edu.nu.sgm.controllers.AddCourseController.setDialog (
    Dialog< ButtonType > dialog) [inline]
```

Sets the dialog reference for this controller.

Parameters

<i>dialog</i>	The dialog instance.
---------------	----------------------

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

- [src/main/java/edu/nu/sgm/controllers/AddCourseController.java](#)

4.2 edu.nu.sgm.controllers.AddStudentController Class Reference

Handles logic for adding or editing a student from the dialog.

Public Member Functions

- void [setDialog](#) (Dialog< ButtonType > dialog)
Sets the dialog reference for this controller.
- void [setStudentData](#) (Student student)
Sets the student data for editing.

4.2.1 Detailed Description

Handles logic for adding or editing a student from the dialog.

4.2.2 Member Function Documentation

4.2.2.1 setDialog()

```
void edu.nu.sgm.controllers.AddStudentController.setDialog (
    Dialog< ButtonType > dialog) [inline]
```

Sets the dialog reference for this controller.

Parameters

<i>dialog</i>	The dialog instance.
---------------	----------------------

4.2.2.2 setStudentData()

```
void edu.nu.sgm.controllers.AddStudentController.setStudentData (
    Student student) [inline]
```

Sets the student data for editing.

Parameters

<i>student</i>	The student to edit.
----------------	----------------------

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

- [src/main/java/edu/nu/sgm/controllers/AddStudentController.java](#)

4.3 edu.nu.sgm.models.Course Class Reference

Represents a course with its details and enrollment count.

Public Member Functions

- [Course](#) (int id, String course_code, String title, String instructor, int credit_hours)
Constructs a [Course](#) object with the specified details.
- boolean [setCourseCode](#) (String course_code)
Sets the code of the course.
- String [getCourseCode](#) ()
Retrieves the code of the course.
- void [setId](#) (int id)
Sets the unique identifier for the course.
- int [getId](#) ()
Retrieves the unique identifier for the course.
- boolean [setTitle](#) (String title)
Sets the name of the course.
- String [getTitle](#) ()
Retrieves the name of the course.
- boolean [setInstructor](#) (String instructor)
Sets the name of the instructor teaching the course.
- String [getInstructor](#) ()
Retrieves the name of the instructor teaching the course.
- boolean [setCreditHours](#) (int credit_hours)
Sets the number of credit hours assigned to the course.
- int [getCreditHours](#) ()
Retrieves the number of credit hours assigned to the course.
- void [setTotalStudents](#) (int total_students)
Sets the total number of students enrolled in the course.
- int [getTotalStudents](#) ()
Retrieves the total number of students enrolled in the course.

4.3.1 Detailed Description

Represents a course with its details and enrollment count.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 Course()

```
edu.nu.sgm.models.Course.Course (
    int id,
    String course_code,
    String title,
    String instructor,
    int credit_hours) [inline]
```

Constructs a [Course](#) object with the specified details.

Parameters

<i>id</i>	The unique identifier for the course.
<i>course_code</i>	The code of the course.
<i>title</i>	The name of the course.
<i>instructor</i>	The name of the instructor teaching the course.
<i>credit_hours</i>	The number of credit hours assigned to the course.

4.3.3 Member Function Documentation

4.3.3.1 getCourseCode()

```
String edu.nu.sgm.models.Course.getCourseCode () [inline]
```

Retrieves the code of the course.

Returns

The code of the course.

4.3.3.2 getCreditHours()

```
int edu.nu.sgm.models.Course.getCreditHours () [inline]
```

Retrieves the number of credit hours assigned to the course.

Returns

The number of credit hours.

4.3.3.3 getId()

```
int edu.nu.sgm.models.Course.getId () [inline]
```

Retrieves the unique identifier for the course.

Returns

The unique identifier for the course.

4.3.3.4 getInstructor()

```
String edu.nu.sgm.models.Course.getInstructor () [inline]
```

Retrieves the name of the instructor teaching the course.

Returns

The name of the instructor.

4.3.3.5 getTitle()

```
String edu.nu.sgm.models.Course.getTitle () [inline]
```

Retrieves the name of the course.

Returns

The name of the course.

4.3.3.6 getTotalStudents()

```
int edu.nu.sgm.models.Course.getTotalStudents () [inline]
```

Retrieves the total number of students enrolled in the course.

Returns

The total number of students enrolled in the course.

4.3.3.7 setCourseCode()

```
boolean edu.nu.sgm.models.Course.setCourseCode (  
    String course_code) [inline]
```

Sets the code of the course.

Parameters

<i>course_code</i>	The code of the course.
--------------------	-------------------------

Returns

true if the course code was successfully set, false otherwise.

4.3.3.8 setCreditHours()

```
boolean edu.nu.sgm.models.Course.setCreditHours (  
    int credit_hours) [inline]
```

Sets the number of credit hours assigned to the course.

Parameters

<i>credit_hours</i>	The number of credit hours.
---------------------	-----------------------------

Returns

`true` if the credit hours were successfully set,
`false` otherwise.

4.3.3.9 setId()

```
void edu.nu.sgm.models.Course.setId (  
    int id) [inline]
```

Sets the unique identifier for the course.

Parameters

<i>id</i>	The unique identifier for the course.
-----------	---------------------------------------

Returns

`true` if the ID was successfully set, `false` otherwise.

4.3.3.10 setInstructor()

```
boolean edu.nu.sgm.models.Course.setInstructor (  
    String instructor) [inline]
```

Sets the name of the instructor teaching the course.

Parameters

<i>instructor</i>	The name of the instructor.
-------------------	-----------------------------

Returns

`true` if the instructor name was successfully set, `false` otherwise.

4.3.3.11 setTitle()

```
boolean edu.nu.sgm.models.Course.setTitle (  
    String title) [inline]
```

Sets the name of the course.

Parameters

<i>name</i>	The name of the course.
-------------	-------------------------

Returns

`true` if the name was successfully set, `false` otherwise.

4.3.3.12 setTotalStudents()

```
void edu.nu.sgm.models.Course.setTotalStudents (
    int total_students) [inline]
```

Sets the total number of students enrolled in the course.

Parameters

<i>total_students</i>	The total number of students.
-----------------------	-------------------------------

Returns

`true` if the total number of students was successfully set, `false` otherwise.

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

- [src/main/java/edu/nu/sgm/models/Course.java](#)

4.4 edu.nu.sgm.CourseCreationTest Class Reference**Public Member Functions**

- void **testAddCourse_Success** ()
- void **testRemoveCourse_Success** ()
- void **testDisplayDetails_Success** ()
- void **testImportCourses_Success** ()
- void **testExportCourses_Success** ()

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

- [src/test/java/edu/nu/sgm/CourseCreationTest.java](#)

4.5 edu.nu.sgm.services.CourseService Class Reference

Provides methods to add, remove, retrieve, and manage courses.

Public Member Functions

- boolean `courseExists` (`Course` course)
Checks if a course exists in the database.
- boolean `addCourse` (`Course` course)
Adds a course to the database.
- boolean `removeCourse` (`Course` course)
Removes a course from the database.
- List< `Student` > `getStudents` (`Course` course)
Retrieves a list of students enrolled in a course.
- int `getTotalStudents` (`Course` course)
Retrieves total number of students enrolled in a course.
- List< `Course` > `getAllCourses` ()
Retrieves all courses from the database.
- String `displayDetails` (`Course` course)
Displays the details of a course.
- `Course` `getCourseById` (int id)
Retrieves a course by its ID.
- List< `Course` > `importCourses` (File file)
Imports courses from a file.
- boolean `exportCourses` (List< `Course` > courses)
Exports courses to a file.

4.5.1 Detailed Description

Provides methods to add, remove, retrieve, and manage courses.

4.5.2 Member Function Documentation

4.5.2.1 `addCourse()`

```
boolean edu.nu.sgm.services.CourseService.addCourse (
    Course course) [inline]
```

Adds a course to the database.

Parameters

<code>course</code>	The course to be added.
---------------------	-------------------------

Returns

true if the course was added successfully, false otherwise.

4.5.2.2 `courseExists()`

```
boolean edu.nu.sgm.services.CourseService.courseExists (
    Course course) [inline]
```

Checks if a course exists in the database.

Parameters

<i>course</i>	The course to check for existence.
---------------	------------------------------------

Returns

true if the course exists, false otherwise.

4.5.2.3 displayDetails()

```
String edu.nu.sgm.services.CourseService.displayDetails (  
    Course course) [inline]
```

Displays the details of a course.

Parameters

<i>course</i>	The course for which to display the details.
---------------	----------------------------------------------

Returns

A string containing the course details.

4.5.2.4 exportCourses()

```
boolean edu.nu.sgm.services.CourseService.exportCourses (  
    List< Course > courses) [inline]
```

Exports courses to a file.

Parameters

<i>courses</i>	The courses to export.
----------------	------------------------

Returns

true if the courses were exported successfully, false otherwise.

4.5.2.5 getAllCourses()

```
List< Course > edu.nu.sgm.services.CourseService.getAllCourses () [inline]
```

Retrieves all courses from the database.

Returns

A list of all courses.

4.5.2.6 getCourseById()

```
Course edu.nu.sgm.services.CourseService.getCourseById (  
    int id) [inline]
```

Retrieves a course by its ID.

Parameters

<i>id</i>	The ID of the course to retrieve.
-----------	-----------------------------------

Returns

The course with the specified ID, or null if not found.

4.5.2.7 getStudents()

```
List< Student > edu.nu.sgm.services.CourseService.getStudents (
    Course course) [inline]
```

Retrieves a list of students enrolled in a course.

Parameters

<i>course</i>	The course for which to retrieve the students.
---------------	------------------------------------------------

Returns

A list of students enrolled in the course, or null if an error occurred.

4.5.2.8 getTotalStudents()

```
int edu.nu.sgm.services.CourseService.getTotalStudents (
    Course course) [inline]
```

Retrieves total number of students enrolled in a course.

Parameters

<i>course</i>	The course for which to count students.
---------------	-----------------------------------------

Returns

The total number of students enrolled in the course, or 0 if an error occurred.

4.5.2.9 importCourses()

```
List< Course > edu.nu.sgm.services.CourseService.importCourses (
    File file) [inline]
```

Imports courses from a file.

Parameters

<i>file</i>	The file containing the course data.
-------------	--------------------------------------

Returns

List of imported courses.

4.5.2.10 removeCourse()

```
boolean edu.nu.sgm.services.CourseService.removeCourse (
    Course course) [inline]
```

Removes a course from the database.

Parameters

<i>course</i>	The course to be removed.
---------------	---------------------------

Returns

true if the course was removed successfully, false otherwise.

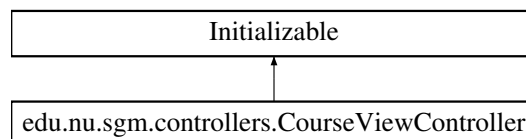
The documentation for this class was generated from the following file:

- src/main/java/edu/nu/sgm/services/[CourseService.java](#)

4.6 edu.nu.sgm.controllers.CourseViewController Class Reference

Handles the course detail view, including enrolled students.

Inheritance diagram for edu.nu.sgm.controllers.CourseViewController:



Public Member Functions

- void **initialize** (URL location, ResourceBundle resources)
Initializes the controller and sets up the table columns.
- void **setCourse** ([Course](#) course)
Sets the course and updates the view with course details.

4.6.1 Detailed Description

Handles the course detail view, including enrolled students.

4.6.2 Member Function Documentation

4.6.2.1 setCourse()

```
void edu.nu.sgm.controllers.CourseViewController.setCourse (
    Course course) [inline]
```

Sets the course and updates the view with course details.

Parameters

<i>course</i>	The course to display.
---------------	------------------------

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

- [src/main/java/edu/nu/sgm/controllers/CourseViewController.java](#)

4.7 edu.nu.sgm.utils.DatabaseManager Class Reference

Classes

- interface [ResultSetMapper](#)

Public Member Functions

- boolean **initializeDatabase** () throws SQLException
- boolean **createCourse** ([Course](#) course) throws SQLException
- boolean **createStudent** ([Student](#) student) throws SQLException
- boolean **createEnrollment** ([Enrollment](#) enrollment) throws SQLException
- boolean **createGrade** ([Enrollment](#) enrollment, [GradeItem](#) grade) throws SQLException
- List< [Course](#) > **fetchCourses** ([Student](#) student) throws SQLException
- List< [Student](#) > **fetchStudents** ([Course](#) course) throws SQLException
- List< [Course](#) > **fetchCourses** () throws SQLException
- [Course](#) **fetchCourse** (int course_id) throws SQLException
- List< [Student](#) > **fetchStudents** () throws SQLException
- List< [GradeItem](#) > **fetchGrades** ([Enrollment](#) enrollment) throws SQLException
- List< [Enrollment](#) > **fetchEnrollment** ([Student](#) student, [Course](#) course) throws SQLException
- List< [Enrollment](#) > **fetchEnrollment** ([Course](#) course) throws SQLException
- List< [Enrollment](#) > **fetchEnrollment** ([Student](#) student) throws SQLException
- int **updateCourse** ([Course](#) course) throws SQLException
- int **updateStudent** ([Student](#) student) throws SQLException
- int **updateGrades** ([GradeItem](#) grade) throws SQLException
- int **deleteCourse** ([Course](#) course) throws SQLException
- int **deleteStudent** ([Student](#) student) throws SQLException
- int **deleteGrade** ([GradeItem](#) grade) throws SQLException
- int **deleteEnrollment** ([Student](#) student, [Course](#) course) throws SQLException
- int **deleteEnrollment** ([Course](#) course) throws SQLException
- int **deleteEnrollment** ([Student](#) student) throws SQLException

4.7.1 Member Function Documentation

4.7.1.1 createCourse()

```
boolean edu.nu.sgm.utils.DatabaseManager.createCourse (  
    Course course) throws SQLException [inline]
```

A method that inserts a course object into the database

Parameters

<i>student</i>	The course to add
----------------	-------------------

Returns

True when the course is added successfully

4.7.1.2 createEnrollment()

```
boolean edu.nu.sgm.utils.DatabaseManager.createEnrollment (  
    Enrollment enrollment) throws SQLException [inline]
```

A method that inserts an [Enrollment](#) object into the database

Parameters

<i>enrollment</i>	The enrollment to add
-------------------	-----------------------

Returns

True when the enrollment is added successfully

4.7.1.3 createGrade()

```
boolean edu.nu.sgm.utils.DatabaseManager.createGrade (  
    Enrollment enrollment,  
    GradeItem grade) throws SQLException [inline]
```

A method that inserts a [GradeItem](#) object into the database

Parameters

<i>enrollment</i>	The course/student pair
<i>grade</i>	The grade to add

Returns

True when the grade is added successfully

4.7.1.4 createStudent()

```
boolean edu.nu.sgm.utils.DatabaseManager.createStudent (  
    Student student) throws SQLException [inline]
```

A method that inserts a [Student](#) object into the database

Parameters

<i>student</i>	The student to add
----------------	--------------------

Returns

True when the student is added successfully

4.7.1.5 deleteCourse()

```
int edu.nu.sgm.utils.DatabaseManager.deleteCourse (  
    Course course) throws SQLException [inline]
```

A method that deletes a [Course](#) object from the database

Parameters

<i>course</i>	The target course
---------------	-------------------

Returns

The number of rows modified

4.7.1.6 deleteEnrollment() [1/3]

```
int edu.nu.sgm.utils.DatabaseManager.deleteEnrollment (  
    Course course) throws SQLException [inline]
```

A method that deletes an [Enrollment](#) object from the database

Parameters

<i>enrollment</i>	The target enrollment
-------------------	-----------------------

Returns

The number of rows modified

4.7.1.7 deleteEnrollment() [2/3]

```
int edu.nu.sgm.utils.DatabaseManager.deleteEnrollment (  
    Student student) throws SQLException [inline]
```

A method that deletes an [Enrollment](#) object from the database

Parameters

<i>enrollment</i>	The target enrollment
-------------------	-----------------------

Returns

The number of rows modified

4.7.1.8 deleteEnrollment() [3/3]

```
int edu.nu.sgm.utils.DatabaseManager.deleteEnrollment (  
    Student student,  
    Course course) throws SQLException [inline]
```

A method that deletes an [Enrollment](#) object from the database

Parameters

<i>enrollment</i>	The target enrollment
-------------------	-----------------------

Returns

The number of rows modified

4.7.1.9 deleteGrade()

```
int edu.nu.sgm.utils.DatabaseManager.deleteGrade (  
    GradeItem grade) throws SQLException [inline]
```

A method that deletes a [GradeItem](#) object from the database

Parameters

<i>grade</i>	The target grade
--------------	------------------

Returns

The number of rows modified

4.7.1.10 deleteStudent()

```
int edu.nu.sgm.utils.DatabaseManager.deleteStudent (  
    Student student) throws SQLException [inline]
```

A method that deletes a [Student](#) object from the database

Parameters

<i>student</i>	The target student
----------------	--------------------

Returns

The number of rows modified

4.7.1.11 fetchCourse()

```
Course edu.nu.sgm.utils.DatabaseManager.fetchCourse (  
    int course_id) throws SQLException [inline]
```

A method that gets all courses in the database

Returns

The list of courses

4.7.1.12 fetchCourses() [1/2]

```
List< Course > edu.nu.sgm.utils.DatabaseManager.fetchCourses () throws SQLException [inline]
```

A method that gets all courses in the database

Returns

The list of courses

4.7.1.13 fetchCourses() [2/2]

```
List< Course > edu.nu.sgm.utils.DatabaseManager.fetchCourses (  
    Student student) throws SQLException [inline]
```

A method that gets the courses a student is enrolled in

Parameters

<i>student</i>	The target student
----------------	--------------------

Returns

The list of courses the target student is enrolled in

4.7.1.14 fetchEnrollment() [1/3]

```
List< Enrollment > edu.nu.sgm.utils.DatabaseManager.fetchEnrollment (  
    Course course) throws SQLException [inline]
```

A method that gets a specific enrollment

Parameters

<i>student</i>	The target student
<i>course</i>	The target course

Returns

The [Enrollment](#) object

4.7.1.15 fetchEnrollment() [2/3]

```
List< Enrollment > edu.nu.sgm.utils.DatabaseManager.fetchEnrollment (
    Student student) throws SQLException [inline]
```

A method that gets a specific enrollment

Parameters

<i>student</i>	The target student
<i>course</i>	The target course

Returns

The [Enrollment](#) object

4.7.1.16 fetchEnrollment() [3/3]

```
List< Enrollment > edu.nu.sgm.utils.DatabaseManager.fetchEnrollment (
    Student student,
    Course course) throws SQLException [inline]
```

A method that gets a specific enrollment

Parameters

<i>student</i>	The target student
<i>course</i>	The target course

Returns

The [Enrollment](#) object

4.7.1.17 fetchGrades()

```
List< GradeItem > edu.nu.sgm.utils.DatabaseManager.fetchGrades (
    Enrollment enrollment) throws SQLException [inline]
```

A method that gets all grades of a student/course pair

Parameters

<i>enrollment</i>	The student/course pair
-------------------	-------------------------

Returns

The list of grades

4.7.1.18 fetchStudents() [1/2]

```
List< Student > edu.nu.sgm.utils.DatabaseManager.fetchStudents () throws SQLException [inline]
```

A method that gets all students in the database

Returns

The list of students

4.7.1.19 fetchStudents() [2/2]

```
List< Student > edu.nu.sgm.utils.DatabaseManager.fetchStudents (  
    Course course) throws SQLException [inline]
```

A method that gets the students enrolled in a course

Parameters

<i>course</i>	The target course
---------------	-------------------

Returns

The list of students enrolled in the target course

4.7.1.20 updateCourse()

```
int edu.nu.sgm.utils.DatabaseManager.updateCourse (  
    Course course) throws SQLException [inline]
```

A method that updates a [Course](#) object in the database

Parameters

<i>course</i>	The target course with updated attributes
---------------	-------------------------------------------

Returns

The number of rows modified

4.7.1.21 updateGrades()

```
int edu.nu.sgm.utils.DatabaseManager.updateGrades (  
    GradeItem grade) throws SQLException [inline]
```

A method that updates a [GradeItem](#) object in the database

Parameters

<i>grade</i>	The target grade with updated attributes
--------------	------------------------------------------

Returns

The number of rows modified

4.7.1.22 updateStudent()

```
int edu.nu.sgm.utils.DatabaseManager.updateStudent (
    Student student) throws SQLException [inline]
```

A method that updates a [Student](#) object in the database

Parameters

<i>student</i>	The target student with updated attributes
----------------	--------------------------------------------

Returns

The number of rows modified

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

- src/main/java/edu/nu/sgm/utils/DatabaseManager.java

4.8 edu.nu.sgm.DatabaseManagerTest Class Reference

Public Member Functions

- void **test_AddStudent** ()
- void **test_AddCourse** ()
- void **test_AddEnrollment** ()
- void **test_RemoveCourse** ()

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

- src/test/java/edu/nu/sgm/DatabaseManagerTest.java

4.9 edu.nu.sgm.controllers.EnrollCourse Class Reference

Public Member Functions

- void **setDialog** (Dialog< ButtonType > dialog)
- void **setStudent** ([Student](#) student)

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

- src/main/java/edu/nu/sgm/controllers/EnrollCourse.java

4.10 edu.nu.sgm.models.Enrollment Class Reference

Represents the enrollment relationship between a student and a course.

Public Member Functions

- [Enrollment](#) (int id, int student_id, int course_id)
The enrollment object constructor.
- int [getId](#) ()
Enrollment ID getter.
- int [getStudent](#) ()
Enrollment student getter.
- int [getCourse](#) ()
Enrollment course getter.
- void [setID](#) (int id)
Enrollment ID setter.

4.10.1 Detailed Description

Represents the enrollment relationship between a student and a course.

4.10.2 Constructor & Destructor Documentation

4.10.2.1 Enrollment()

```
edu.nu.sgm.models.Enrollment.Enrollment (
    int id,
    int student_id,
    int course_id) [inline]
```

The enrollment object constructor.

Parameters

<i>id</i>	The enrollment ID.
<i>student_id</i>	The student to enroll.
<i>course_id</i>	The course to enroll in.

4.10.3 Member Function Documentation

4.10.3.1 getCourse()

```
int edu.nu.sgm.models.Enrollment.getCourse () [inline]
```

[Enrollment](#) course getter.

Returns

[Course](#) enrolled in.

4.10.3.2 getId()

```
int edu.nu.sgm.models.Enrollment.getId () [inline]
```

[Enrollment](#) ID getter.

Returns

[Enrollment](#) ID.

4.10.3.3 getStudent()

```
int edu.nu.sgm.models.Enrollment.getStudent () [inline]
```

[Enrollment](#) student getter.

Returns

[Student](#) enrolled.

4.10.3.4 setID()

```
void edu.nu.sgm.models.Enrollment.setID (  
    int id) [inline]
```

[Enrollment](#) ID setter.

Parameters

Enrollment	ID.
----------------------------	-----

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

- src/main/java/edu/nu/sgm/models/[Enrollment.java](#)

4.11 edu.nu.sgm.services.EnrollmentService Class Reference

Provides methods to enroll students, drop courses, and generate report cards.

Public Member Functions

- boolean [enrollStudent](#) ([Student](#) student, [Course](#) course)
Enrolls a student in a course.
- boolean [dropCourse](#) ([Student](#) student, [Course](#) course)
Drops a course from the student's record.
- String [generateReportCard](#) ([Student](#) student)
Generates a student's report card.

4.11.1 Detailed Description

Provides methods to enroll students, drop courses, and generate report cards.

4.11.2 Member Function Documentation

4.11.2.1 dropCourse()

```
boolean edu.nu.sgm.services.EnrollmentService.dropCourse (  
    Student student,  
    Course course) [inline]
```

Drops a course from the student's record.

Parameters

<i>student</i>	The target student.
<i>course</i>	The target course.

Returns

True when the course is successfully dropped.

4.11.2.2 enrollStudent()

```
boolean edu.nu.sgm.services.EnrollmentService.enrollStudent (  
    Student student,  
    Course course) [inline]
```

Enrolls a student in a course.

Parameters

<i>student</i>	The target student.
<i>course</i>	The target course.

Returns

True when the student is successfully enrolled.

4.11.2.3 generateReportCard()

```
String edu.nu.sgm.services.EnrollmentService.generateReportCard (  
    Student student) [inline]
```

Generates a student's report card.

Parameters

<i>student</i>	The target student.
----------------	---------------------

Returns

The generated report card.

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

- [src/main/java/edu/nu/sgm/services/EnrollmentService.java](#)

4.12 edu.nu.sgm.EnrollmentTest Class Reference

Public Member Functions

- void **generateReport** ()

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

- [src/test/java/edu/nu/sgm/EnrollmentTest.java](#)

4.13 edu.nu.sgm.controllers.EnrollStudents Class Reference

Handles the dialog for enrolling a student in a course.

Public Member Functions

- void [setDialog](#) (Dialog< ButtonType > dialog)
Sets the dialog reference for this controller.
- void [setCourse](#) ([Course](#) course)
Sets the course to enroll the student in.

4.13.1 Detailed Description

Handles the dialog for enrolling a student in a course.

4.13.2 Member Function Documentation

4.13.2.1 setCourse()

```
void edu.nu.sgm.controllers.EnrollStudents.setCourse (  
    Course course) [inline]
```

Sets the course to enroll the student in.

Parameters

<i>course</i>	The course to enroll in.
---------------	--------------------------

4.13.2.2 setDialog()

```
void edu.nu.sgm.controllers.EnrollStudents.setDialog (  
    Dialog< ButtonType > dialog) [inline]
```

Sets the dialog reference for this controller.

Parameters

<i>dialog</i>	The dialog instance.
---------------	----------------------

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

- src/main/java/edu/nu/sgm/controllers/[EnrollStudents.java](#)

4.14 edu.nu.sgm.models.GradeItem Class Reference

Represents a grade item (e.g., assignment, exam) for a student.

Public Member Functions

- [GradeItem](#) (int id, String title, String category, double score, double max_score, String feedback, double weight)
Constructs a [GradeItem](#) with specified parameters.
- boolean [setTitle](#) (String name)
Sets the title of the grade item.
- String [getTitle](#) ()
Gets the title of the grade item.
- boolean [setCategory](#) (String category)
Sets the category of the grade item.
- String [getCategory](#) ()
Gets the category of the grade item.
- boolean [setScore](#) (double score, double maxScore)
Sets the score and maximum score of the grade item.
- double [getScore](#) ()
Gets the score of the grade item.
- boolean [setFeedback](#) (String feedback)
Sets the feedback for the grade item.
- String [getFeedback](#) ()
Gets the feedback for the grade item.
- int [getId](#) ()
Gets the unique identifier of the grade item.
- boolean [setId](#) (int id)

- Sets the unique identifier of the grade item.*
 - double `getMaxScore` ()*Gets the maximum score for the grade item.*
- boolean `setMaxScore` (double maxScore)*Sets the maximum score for the grade item.*
- boolean `setWeight` (double weight)*Sets the weight of the grade item.*
- double `getWeight` ()*Gets the weight of the grade item.*

4.14.1 Detailed Description

Represents a grade item (e.g., assignment, exam) for a student.

4.14.2 Constructor & Destructor Documentation

4.14.2.1 GradeItem()

```
edu.nu.sgm.models.GradeItem.GradeItem (
    int id,
    String title,
    String category,
    double score,
    double max_score,
    String feedback,
    double weight) [inline]
```

Constructs a [GradeItem](#) with specified parameters.

Parameters

<i>id</i>	Unique identifier
<i>title</i>	Title of the item
<i>category</i>	Category of the item
<i>score</i>	Score obtained
<i>max_score</i>	Maximum possible score
<i>feedback</i>	Feedback text
<i>weight</i>	Weight as a percentage of the total grade

4.14.3 Member Function Documentation

4.14.3.1 getCategory()

```
String edu.nu.sgm.models.GradeItem.getCategory () [inline]
```

Gets the category of the grade item.

Returns

The category

4.14.3.2 `getFeedback()`

```
String edu.nu.sgm.models.GradeItem.getFeedback () [inline]
```

Gets the feedback for the grade item.

Returns

The feedback

4.14.3.3 `getId()`

```
int edu.nu.sgm.models.GradeItem.getId () [inline]
```

Gets the unique identifier of the grade item.

Returns

The ID

4.14.3.4 `getMaxScore()`

```
double edu.nu.sgm.models.GradeItem.getMaxScore () [inline]
```

Gets the maximum score for the grade item.

Returns

The maximum score

4.14.3.5 `getScore()`

```
double edu.nu.sgm.models.GradeItem.getScore () [inline]
```

Gets the score of the grade item.

Returns

The score

4.14.3.6 `getTitle()`

```
String edu.nu.sgm.models.GradeItem.getTitle () [inline]
```

Gets the title of the grade item.

Returns

The title

4.14.3.7 `getWeight()`

```
double edu.nu.sgm.models.GradeItem.getWeight () [inline]
```

Gets the weight of the grade item.

Returns

The weight as a percentage

4.14.3.8 `setCategory()`

```
boolean edu.nu.sgm.models.GradeItem.setCategory (  
    String category) [inline]
```

Sets the category of the grade item.

Parameters

<i>category</i>	The new category
-----------------	------------------

Returns

True if the category is valid and set; false otherwise

4.14.3.9 `setFeedback()`

```
boolean edu.nu.sgm.models.GradeItem.setFeedback (  
    String feedback) [inline]
```

Sets the feedback for the grade item.

Parameters

<i>feedback</i>	The feedback text
-----------------	-------------------

Returns

Always true

4.14.3.10 `setId()`

```
boolean edu.nu.sgm.models.GradeItem.setId (  
    int id) [inline]
```

Sets the unique identifier of the grade item.

Parameters

<i>id</i>	The ID to set
-----------	---------------

Returns

True if the ID is valid and set; false otherwise

4.14.3.11 setMaxScore()

```
boolean edu.nu.sgm.models.GradeItem.setMaxScore (  
    double maxScore)    [inline]
```

Sets the maximum score for the grade item.

Parameters

<i>maxScore</i>	The maximum score to set
-----------------	--------------------------

Returns

True if valid; false otherwise

4.14.3.12 setScore()

```
boolean edu.nu.sgm.models.GradeItem.setScore (  
    double score,  
    double maxScore)    [inline]
```

Sets the score and maximum score of the grade item.

Parameters

<i>score</i>	The score obtained
<i>maxScore</i>	The maximum possible score

Returns

True if values are valid and set; false otherwise

4.14.3.13 setTitle()

```
boolean edu.nu.sgm.models.GradeItem.setTitle (  
    String name)    [inline]
```

Sets the title of the grade item.

Parameters

<i>name</i>	The new title
-------------	---------------

Returns

True if the title is valid and set; false otherwise

4.14.3.14 setWeight()

```
boolean edu.nu.sgm.models.GradeItem.setWeight (
    double weight) [inline]
```

Sets the weight of the grade item.

Parameters

<i>weight</i>	The weight as a percentage (0–100)
---------------	------------------------------------

Returns

True if valid; false otherwise

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

- [src/main/java/edu/nu/sgm/models/GradeItem.java](#)

4.15 edu.nu.sgm.services.GradelItemService Class Reference

Provides methods to create, retrieve, delete, and process grade items.

Public Member Functions

- boolean [addGradeItem](#) ([Enrollment](#) enrollment, [GradeItem](#) item)
Adds a new grade item to an enrollment.
- List< [GradeItem](#) > [getGrades](#) ([Enrollment](#) enrollment)
Retrieves all grade items for a given enrollment.
- boolean [removeGradeItem](#) ([GradeItem](#) grade, [Enrollment](#) enrollment)
Removes a specific grade item from an enrollment.
- double [calculateTotalGrade](#) ([Enrollment](#) enrollment)
Calculates the total grade for an enrollment based on weighted scores.
- List< [GradeItem](#) > [impoGradeItems](#) (File file)
Imports grade items from a CSV file.
- boolean [exportGradeItems](#) (List< [Enrollment](#) > enrollments)
Exports grade items of multiple enrollments to a CSV file.

4.15.1 Detailed Description

Provides methods to create, retrieve, delete, and process grade items.

4.15.2 Member Function Documentation

4.15.2.1 addGradeItem()

```
boolean edu.nu.sgm.services.GradeItemService.addGradeItem (  
    Enrollment enrollment,  
    GradeItem item) [inline]
```

Adds a new grade item to an enrollment.

Parameters

<i>enrollment</i>	The enrollment associated with the grade
<i>item</i>	The grade item to add

Returns

True if the grade item is successfully added; false otherwise

4.15.2.2 calculateTotalGrade()

```
double edu.nu.sgm.services.GradeItemService.calculateTotalGrade (  
    Enrollment enrollment) [inline]
```

Calculates the total grade for an enrollment based on weighted scores.

Parameters

<i>enrollment</i>	The enrollment whose total grade is to be calculated
-------------------	------------------------------------------------------

Returns

The total grade as a percentage

4.15.2.3 exportGradeItems()

```
boolean edu.nu.sgm.services.GradeItemService.exportGradeItems (  
    List< Enrollment > enrollments) [inline]
```

Exports grade items of multiple enrollments to a CSV file.

Parameters

<i>enrollments</i>	A list of enrollments whose grades will be exported
--------------------	-----------------------------------------------------

Returns

True if export is successful

4.15.2.4 getGrades()

```
List< GradeItem > edu.nu.sgm.services.GradeItemService.getGrades (  
    Enrollment enrollment) [inline]
```

Retrieves all grade items for a given enrollment.

Parameters

<i>enrollment</i>	The enrollment whose grades are to be retrieved
-------------------	-------------------------------------------------

Returns

A list of [GradeItem](#) objects, or an empty list on error or null input

4.15.2.5 impoGradeItems()

```
List< GradeItem > edu.nu.sgm.services.GradeItemService.impoGradeItems (  
    File file) [inline]
```

Imports grade items from a CSV file.

Parameters

<i>file</i>	The CSV file to import from
-------------	-----------------------------

Returns

A list of [GradeItem](#) objects, or an empty list if import fails

4.15.2.6 removeGradeItem()

```
boolean edu.nu.sgm.services.GradeItemService.removeGradeItem (  
    GradeItem grade,  
    Enrollment enrollment) [inline]
```

Removes a specific grade item from an enrollment.

Parameters

<i>grade</i>	The grade item to remove
<i>enrollment</i>	The associated enrollment

Returns

True if the item is successfully removed; false otherwise

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

- [src/main/java/edu/nu/sgm/services/GradeItemService.java](#)

4.16 edu.nu.sgm.GradeItemServiceTest Class Reference

Public Member Functions

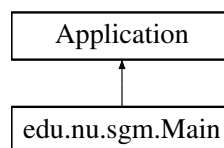
- void **testAddGradeItem_Success** ()
- void **testRemoveGradeItem_Success** ()
- void **testGetGrades_NotEmpty** ()
- void **testCalculateTotalGrade** ()

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

- [src/test/java/edu/nu/sgm/GradeItemServiceTest.java](#)

4.17 edu.nu.sgm.Main Class Reference

Inheritance diagram for edu.nu.sgm.Main:

**Public Member Functions**

- void **start** (Stage stage)

Static Public Member Functions

- static void **main** (String[] args)

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

- [src/main/java/edu/nu/sgm/Main.java](#)

4.18 edu.nu.sgm.controllers.MainViewController Class Reference

Public Member Functions

- void **switchToScene1** (ActionEvent event) throws IOException
- void **SwitchTOAddStudent** (ActionEvent event)
- void **SwitchTOAddCourse** (ActionEvent event)
- void **initialize** ()

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

- src/main/java/edu/nu/sgm/controllers/MainViewController.java

4.19 edu.nu.sgm.utils.Reader Class Reference

Provides static methods for reading and writing CSV files and parsing/exporting model objects.

Static Public Member Functions

- static< T > List< T > **readCSV** (File file, Function< String, T > parser)
Reads a CSV file and parses each line into an object using the provided parser.
- static< T > void **writeCSV** (File file, List< T > list, Function< T, String > parser)
Writes a list of objects to a CSV file using the provided parser.
- static **Course** **parseCourseImport** (String line)
*Parses a CSV line into a **Course** object.*
- static **Student** **parseStudentImport** (String line)
*Parses a CSV line into a **Student** object.*
- static **GradelItem** **parseGradelItemsImport** (String line)
*Parses a CSV line into a **GradelItem** object.*
- static String **parseCourseExport** (**Course** course)
*Converts a **Course** object to a CSV line for export.*
- static String **parseStudentExport** (**Student** student)
*Converts a **Student** object to a CSV line for export.*
- static String **parseGradelItemsExport** (**GradelItem** grade)
*Converts a **GradelItem** object to a CSV line for export.*

4.19.1 Detailed Description

Provides static methods for reading and writing CSV files and parsing/exporting model objects.

4.19.2 Member Function Documentation

4.19.2.1 parseCourseExport()

```
static String edu.nu.sgm.utils.Reader.parseCourseExport (
    Course course) [inline], [static]
```

Converts a **Course** object to a CSV line for export.

Parameters

<i>course</i>	The Course object.
---------------	------------------------------------

Returns

The CSV line.

4.19.2.2 parseCourseImport()

```
static Course edu.nu.sgm.utils.Reader.parseCourseImport (  
    String line) [inline], [static]
```

Parses a CSV line into a [Course](#) object.

Parameters

<i>line</i>	The CSV line.
-------------	---------------

Returns

The parsed [Course](#) object.

4.19.2.3 parseGradeItemsExport()

```
static String edu.nu.sgm.utils.Reader.parseGradeItemsExport (  
    GradeItem grade) [inline], [static]
```

Converts a [GradeItem](#) object to a CSV line for export.

Parameters

<i>grade</i>	The GradeItem object.
--------------	---------------------------------------

Returns

The CSV line.

4.19.2.4 parseGradeItemsImport()

```
static GradeItem edu.nu.sgm.utils.Reader.parseGradeItemsImport (  
    String line) [inline], [static]
```

Parses a CSV line into a [GradeItem](#) object.

Parameters

<i>line</i>	The CSV line.
-------------	---------------

Returns

The parsed [GradleItem](#) object.

4.19.2.5 parseStudentExport()

```
static String edu.nu.sgm.utils.Reader.parseStudentExport (  
    Student student) [inline], [static]
```

Converts a [Student](#) object to a CSV line for export.

Parameters

<i>student</i>	The Student object.
----------------	-------------------------------------

Returns

The CSV line.

4.19.2.6 parseStudentImport()

```
static Student edu.nu.sgm.utils.Reader.parseStudentImport (  
    String line) [inline], [static]
```

Parses a CSV line into a [Student](#) object.

Parameters

<i>line</i>	The CSV line.
-------------	---------------

Returns

The parsed [Student](#) object.

4.19.2.7 readCSV()

```
static< T > List< T > edu.nu.sgm.utils.Reader.readCSV (  
    File file,  
    Function< String, T > parser) [inline], [static]
```

Reads a CSV file and parses each line into an object using the provided parser.

Template Parameters

<i>T</i>	The type of object to parse.
----------	------------------------------

Parameters

<i>file</i>	The CSV file to read.
<i>parser</i>	A function that parses a line into an object of type T.

Returns

A list of parsed objects.

4.19.2.8 writeCSV()

```
static< T > void edu.nu.sgm.utils.Reader.writeCSV (
    File file,
    List< T > list,
    Function< T, String > parser) [inline], [static]
```

Writes a list of objects to a CSV file using the provided parser.

Template Parameters

<i>T</i>	The type of object to write.
----------	------------------------------

Parameters

<i>file</i>	The CSV file to write to.
<i>list</i>	The list of objects to write.
<i>parser</i>	A function that converts an object of type T to a CSV line.

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

- [src/main/java/edu/nu/sgm/utils/Reader.java](#)

4.20 edu.nu.sgm.utils.DatabaseManager.ResultSetMapper< T > Interface Template Reference

Public Member Functions

- **T map** (ResultSet results) throws SQLException

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

- [src/main/java/edu/nu/sgm/utils/DatabaseManager.java](#)

4.21 edu.nu.sgm.models.Student Class Reference

Represents a student with ID, name, and email.

Public Member Functions

- [Student](#) (int id, String firstName, String lastName, String email)
Constructs a [Student](#) object with the specified details.
- int **getId** ()
- void **setId** (int id)
- String **getFirstName** ()
- void **setName** (String firstName)
- String **getLastName** ()
- void **setLastName** (String lastName)
- String **getEmail** ()
- void **setEmail** (String email)
- String [generateEmail](#) ()
Generates an email for the student based on their name and ID.

4.21.1 Detailed Description

Represents a student with ID, name, and email.

4.21.2 Constructor & Destructor Documentation

4.21.2.1 Student()

```
edu.nu.sgm.models.Student.Student (
    int id,
    String firstName,
    String lastName,
    String email) [inline]
```

Constructs a [Student](#) object with the specified details.

Parameters

<i>id</i>	The student's ID.
<i>firstName</i>	The student's first name.
<i>lastName</i>	The student's last name.
<i>email</i>	The student's email address.

4.21.3 Member Function Documentation

4.21.3.1 generateEmail()

```
String edu.nu.sgm.models.Student.generateEmail () [inline]
```

Generates an email for the student based on their name and ID.

Returns

The generated email address.

A method generate email of each student

Returns

email that contain the first letter of the student, the last name and 4 numbers from the id

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

- [src/main/java/edu/nu/sgm/models/Student.java](#)

4.22 edu.nu.sgm.services.StudentService Class Reference

Provides methods to add, remove, update, and retrieve students and their courses.

Public Member Functions

- boolean [addstudent](#) ([Student](#) s)
Adds a student to the database after validation.
- List< [Student](#) > [getStudents](#) ()
Retrieves all students from the database.
- boolean [removeStudent](#) ([Student](#) s)
Removes a student from the list and the database.
- List< [Course](#) > [getCourses](#) ([Student](#) s)
Retrieves the courses a student is enrolled in.
- int [getTotalCourses](#) ([Student](#) s)
Counts the total courses a student is enrolled in.
- String [displayDetails](#) ([Student](#) s)
Displays the details of a student.
- [Student](#) [getStudentById](#) (int id)
Retrieves a student by their ID.
- boolean [updateStudent](#) ([Student](#) student)
Updates a student's information in the database.

4.22.1 Detailed Description

Provides methods to add, remove, update, and retrieve students and their courses.

4.22.2 Member Function Documentation

4.22.2.1 addstudent()

```
boolean edu.nu.sgm.services.StudentService.addstudent (  
    Student s) [inline]
```

Adds a student to the database after validation.

Parameters

<code>s</code>	The student to add.
----------------	---------------------

Returns

true if added successfully, false otherwise.

4.22.2.2 displayDetails()

```
String edu.nu.sgm.services.StudentService.displayDetails (  
    Student s) [inline]
```

Displays the details of a student.

Parameters

<code>s</code>	The student.
----------------	--------------

Returns

String with student details.

4.22.2.3 getCourses()

```
List< Course > edu.nu.sgm.services.StudentService.getCourses (  
    Student s) [inline]
```

Retrieves the courses a student is enrolled in.

Parameters

<code>s</code>	The student.
----------------	--------------

Returns

List of courses.

4.22.2.4 getStudentById()

```
Student edu.nu.sgm.services.StudentService.getStudentById (  
    int id) [inline]
```

Retrieves a student by their ID.

Parameters

<i>id</i>	The student ID.
-----------	-----------------

Returns

The student object or null if not found.

4.22.2.5 getStudents()

```
List< Student > edu.nu.sgm.services.StudentService.getStudents () [inline]
```

Retrieves all students from the database.

Returns

List of students.

4.22.2.6 getTotalCourses()

```
int edu.nu.sgm.services.StudentService.getTotalCourses (  
    Student s) [inline]
```

Counts the total courses a student is enrolled in.

Parameters

<i>s</i>	The student.
----------	--------------

Returns

The number of courses.

4.22.2.7 removeStudent()

```
boolean edu.nu.sgm.services.StudentService.removeStudent (  
    Student s) [inline]
```

Removes a student from the list and the database.

Parameters

<i>s</i>	The student to remove.
----------	------------------------

Returns

true if removed successfully, false otherwise.

4.22.2.8 updateStudent()

```
boolean edu.nu.sgm.services.StudentService.updateStudent (  
    Student student) [inline]
```

Updates a student's information in the database.

Parameters

<i>student</i>	The student to update.
----------------	------------------------

Returns

true if updated successfully, false otherwise.

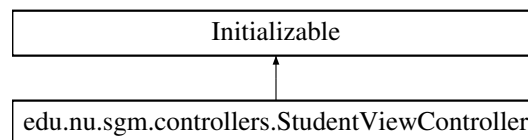
The documentation for this class was generated from the following file:

- src/main/java/edu/nu/sgm/services/[StudentService.java](#)

4.23 edu.nu.sgm.controllers.StudentViewController Class Reference

Handles the student detail view, including enrolled courses.

Inheritance diagram for edu.nu.sgm.controllers.StudentViewController:



Public Member Functions

- void **initialize** (URL location, ResourceBundle resources)
Initializes the controller and sets up the table columns.
- void **setStudent** ([Student](#) student)
Sets the student and updates the view with student details.
- void **refreshView** ()
Refreshes the student view with current details and enrolled courses.

4.23.1 Detailed Description

Handles the student detail view, including enrolled courses.

4.23.2 Member Function Documentation

4.23.2.1 setStudent()

```
void edu.nu.sgm.controllers.StudentViewController.setStudent (
    Student student) [inline]
```

Sets the student and updates the view with student details.

Parameters

<i>student</i>	The student to display.
----------------	-------------------------

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

- [src/main/java/edu/nu/sgm/controllers/StudentViewController.java](#)

Chapter 5

File Documentation

5.1 src/main/java/edu/nu/sgm/controllers/AddCourseController.java File Reference

Controller for adding a new course via the add-course dialog.

```
import edu.nu.sgm.models.Course;
import edu.nu.sgm.services.CourseService;
import javafx.fxml.FXML;
import javafx.scene.control.ButtonType;
import javafx.scene.control.Dialog;
import javafx.scene.control.TextField;
import javafx.scene.control.Alert;
```

Classes

- class [edu.nu.sgm.controllers.AddCourseController](#)
Handles the logic for adding a new course from the dialog.

5.1.1 Detailed Description

Controller for adding a new course via the add-course dialog.

Author

5.2 src/main/java/edu/nu/sgm/controllers/AddStudentController.java File Reference

Controller for adding or editing a student via the add-student dialog.

```
import edu.nu.sgm.models.Student;
import edu.nu.sgm.services.StudentService;
import javafx.fxml.FXML;
import javafx.scene.control.ButtonType;
import javafx.scene.control.Dialog;
import javafx.scene.control.TextField;
import javafx.scene.control.Alert;
```

Classes

- class [edu.nu.sgm.controllers.AddStudentController](#)
Handles logic for adding or editing a student from the dialog.

5.2.1 Detailed Description

Controller for adding or editing a student via the add-student dialog.

5.3 src/main/java/edu/nu/sgm/controllers/CourseViewController.java File Reference

Controller for displaying and managing course details and enrolled students.

```
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.text.Text;
import javafx.stage.Stage;
import edu.nu.sgm.models.Course;
import edu.nu.sgm.models.Student;
import edu.nu.sgm.services.CourseService;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import java.net.URL;
import java.util.ResourceBundle;
import java.util.Collections;
```

Classes

- class [edu.nu.sgm.controllers.CourseViewController](#)
Handles the course detail view, including enrolled students.

5.3.1 Detailed Description

Controller for displaying and managing course details and enrolled students.

5.4 src/main/java/edu/nu/sgm/controllers/EnrollStudents.java File Reference

Controller for enrolling a student in a course from the course view dialog.

```
import edu.nu.sgm.services.EnrollmentService;
import edu.nu.sgm.services.StudentService;
import edu.nu.sgm.models.Course;
import javafx.fxml.FXML;
import javafx.scene.control.Alert;
import javafx.scene.control.ButtonType;
import javafx.scene.control.Dialog;
import javafx.scene.control.TextField;
```

Classes

- class [edu.nu.sgm.controllers.EnrollStudents](#)
Handles the dialog for enrolling a student in a course.

5.4.1 Detailed Description

Controller for enrolling a student in a course from the course view dialog.

5.5 src/main/java/edu/nu/sgm/controllers/StudentViewController.java File Reference

Controller for displaying and managing student details and enrolled courses.

```
import java.net.URL;
import java.util.Collections;
import java.util.List;
import java.util.ResourceBundle;
import edu.nu.sgm.models.Course;
import edu.nu.sgm.models.Enrollment;
import edu.nu.sgm.models.Student;
import edu.nu.sgm.services.EnrollmentService;
import edu.nu.sgm.services.GradeItemService;
import edu.nu.sgm.services.StudentService;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.control.Alert;
import javafx.scene.control.Button;
import javafx.scene.control.ButtonType;
import javafx.scene.control.ComboBox;
import javafx.scene.control.Dialog;
import javafx.scene.control.DialogPane;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
```

Classes

- class [edu.nu.sgm.controllers.StudentViewController](#)
Handles the student detail view, including enrolled courses.

5.5.1 Detailed Description

Controller for displaying and managing student details and enrolled courses.

5.6 [src/main/java/edu/nu/sgm/models/Course.java](#) File Reference

Represents a course in the student grading system.

Classes

- class [edu.nu.sgm.models.Course](#)
Represents a course with its details and enrollment count.

5.6.1 Detailed Description

Represents a course in the student grading system.

This class contains details about a course such as its ID, name, credit hours, and the total number of students enrolled.

5.7 [src/main/java/edu/nu/sgm/models/Enrollment.java](#) File Reference

Represents an enrollment of a student in a course.

Classes

- class [edu.nu.sgm.models.Enrollment](#)
Represents the enrollment relationship between a student and a course.

5.7.1 Detailed Description

Represents an enrollment of a student in a course.

5.8 [src/main/java/edu/nu/sgm/models/GradeItem.java](#) File Reference

Represents a single grade item for a student.

Classes

- class [edu.nu.sgm.models.GradelItem](#)
Represents a grade item (e.g., assignment, exam) for a student.

5.8.1 Detailed Description

Represents a single grade item for a student.

A GradelItem includes information such as title, category, score, maximum possible score, feedback, an identifier, and the item's weight in the total grade calculation.

5.9 src/main/java/edu/nu/sgm/models/Student.java File Reference

Represents a student in the grading system.

Classes

- class [edu.nu.sgm.models.Student](#)
Represents a student with ID, name, and email.

5.9.1 Detailed Description

Represents a student in the grading system.

5.10 src/main/java/edu/nu/sgm/services/CourseService.java File Reference

Service class for managing courses in the grading system.

```
import edu.nu.sgm.models.Course;
import edu.nu.sgm.models.Student;
import edu.nu.sgm.utils.DatabaseManager;
import java.io.File;
import edu.nu.sgm.utils.Reader;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
```

Classes

- class [edu.nu.sgm.services.CourseService](#)
Provides methods to add, remove, retrieve, and manage courses.

5.10.1 Detailed Description

Service class for managing courses in the grading system.

5.11 src/main/java/edu/nu/sgm/services/EnrollmentService.java File Reference

Service class for managing student enrollments in courses.

```
import edu.nu.sgm.models.*;
import edu.nu.sgm.utils.DatabaseManager;
import java.sql.SQLException;
```

Classes

- class [edu.nu.sgm.services.EnrollmentService](#)
Provides methods to enroll students, drop courses, and generate report cards.

5.11.1 Detailed Description

Service class for managing student enrollments in courses.

5.12 src/main/java/edu/nu/sgm/services/GradeItemService.java File Reference

Service class for managing grade items related to student enrollments.

```
import edu.nu.sgm.models.*;
import edu.nu.sgm.utils.*;
import java.io.File;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
```

Classes

- class [edu.nu.sgm.services.GradeItemService](#)
Provides methods to create, retrieve, delete, and process grade items.

5.12.1 Detailed Description

Service class for managing grade items related to student enrollments.

5.13 src/main/java/edu/nu/sgm/services/StudentService.java File Reference

Service class for managing students in the grading system.

```
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashSet;
import java.util.List;
import java.util.Set;
import edu.nu.sgm.models.*;
import edu.nu.sgm.utils.DatabaseManager;
```

Classes

- class [edu.nu.sgm.services.StudentService](#)
Provides methods to add, remove, update, and retrieve students and their courses.

5.13.1 Detailed Description

Service class for managing students in the grading system.

5.14 src/main/java/edu/nu/sgm/utils/Reader.java File Reference

Utility class for reading and writing CSV files for courses, students, and grade items.

```
import java.io.File;
import java.io.IOException;
import java.nio.file.Files;
import java.util.ArrayList;
import java.util.List;
import java.util.function.Function;
import edu.nu.sgm.models.Course;
import edu.nu.sgm.models.GradeItem;
import edu.nu.sgm.models.Student;
```

Classes

- class [edu.nu.sgm.utils.Reader](#)
Provides static methods for reading and writing CSV files and parsing/exporting model objects.

5.14.1 Detailed Description

Utility class for reading and writing CSV files for courses, students, and grade items.

Index

- addCourse
 - edu.nu.sgm.controllers.AddCourseController, 7
 - edu.nu.sgm.services.CourseService, 14
- addGradeItem
 - edu.nu.sgm.services.GradeItemService, 36
- addstudent
 - edu.nu.sgm.services.StudentService, 44
- calculateTotalGrade
 - edu.nu.sgm.services.GradeItemService, 36
- Course
 - edu.nu.sgm.models.Course, 10
- courseExists
 - edu.nu.sgm.services.CourseService, 14
- createCourse
 - edu.nu.sgm.utils.DatabaseManager, 19
- createEnrollment
 - edu.nu.sgm.utils.DatabaseManager, 19
- createGrade
 - edu.nu.sgm.utils.DatabaseManager, 19
- createStudent
 - edu.nu.sgm.utils.DatabaseManager, 19
- deleteCourse
 - edu.nu.sgm.utils.DatabaseManager, 20
- deleteEnrollment
 - edu.nu.sgm.utils.DatabaseManager, 20, 21
- deleteGrade
 - edu.nu.sgm.utils.DatabaseManager, 21
- deleteStudent
 - edu.nu.sgm.utils.DatabaseManager, 21
- displayDetails
 - edu.nu.sgm.services.CourseService, 15
 - edu.nu.sgm.services.StudentService, 45
- dropCourse
 - edu.nu.sgm.services.EnrollmentService, 28
- edu.nu.sgm.controllers.AddCourseController, 7
 - addCourse, 7
 - setDialog, 7
- edu.nu.sgm.controllers.AddStudentController, 8
 - setDialog, 8
 - setStudentData, 8
- edu.nu.sgm.controllers.CourseViewController, 17
 - setCourse, 18
- edu.nu.sgm.controllers.EnrollCourse, 25
- edu.nu.sgm.controllers.EnrollStudents, 29
 - setCourse, 29
 - setDialog, 30
- edu.nu.sgm.controllers.MainViewController, 39
- edu.nu.sgm.controllers.StudentViewController, 47
 - setStudent, 47
- edu.nu.sgm.CourseCreationTest, 13
- edu.nu.sgm.DatabaseManagerTest, 25
- edu.nu.sgm.EnrollmentTest, 29
- edu.nu.sgm.GradeItemServiceTest, 38
- edu.nu.sgm.Main, 38
- edu.nu.sgm.models.Course, 9
 - Course, 10
 - getCourseCode, 10
 - getCreditHours, 10
 - getId, 10
 - getInstructor, 10
 - getTitle, 11
 - getTotalStudents, 11
 - setCourseCode, 11
 - setCreditHours, 11
 - setId, 12
 - setInstructor, 12
 - setTitle, 12
 - setTotalStudents, 13
- edu.nu.sgm.models.Enrollment, 26
 - Enrollment, 26
 - getCourse, 26
 - getId, 26
 - getStudent, 27
 - setId, 27
- edu.nu.sgm.models.GradeItem, 30
 - getCategory, 31
 - getFeedback, 31
 - getId, 32
 - getMaxScore, 32
 - getScore, 32
 - getTitle, 32
 - getWeight, 32
 - GradeItem, 31
 - setCategory, 33
 - setFeedback, 33
 - setId, 33
 - setMaxScore, 34
 - setScore, 34
 - setTitle, 34
 - setWeight, 35
- edu.nu.sgm.models.Student, 43
 - generateEmail, 44
 - Student, 43
- edu.nu.sgm.services.CourseService, 13
 - addCourse, 14
 - courseExists, 14

- displayDetails, 15
- exportCourses, 15
- getAllCourses, 15
- getCourseById, 15
- getStudents, 16
- getTotalStudents, 16
- importCourses, 16
- removeCourse, 17
- edu.nu.sgm.services.EnrollmentService, 27
 - dropCourse, 28
 - enrollStudent, 28
 - generateReportCard, 28
- edu.nu.sgm.services.GradelItemService, 35
 - addGradelItem, 36
 - calculateTotalGrade, 36
 - exportGradelItems, 36
 - getGrades, 37
 - impoGradelItems, 37
 - removeGradelItem, 37
- edu.nu.sgm.services.StudentService, 44
 - addstudent, 44
 - displayDetails, 45
 - getCourses, 45
 - getStudentById, 45
 - getStudents, 46
 - getTotalCourses, 46
 - removeStudent, 46
 - updateStudent, 46
- edu.nu.sgm.utils.DatabaseManager, 18
 - createCourse, 19
 - createEnrollment, 19
 - createGrade, 19
 - createStudent, 19
 - deleteCourse, 20
 - deleteEnrollment, 20, 21
 - deleteGrade, 21
 - deleteStudent, 21
 - fetchCourse, 22
 - fetchCourses, 22
 - fetchEnrollment, 22, 23
 - fetchGrades, 23
 - fetchStudents, 24
 - updateCourse, 24
 - updateGrades, 24
 - updateStudent, 25
- edu.nu.sgm.utils.DatabaseManager.ResultSetMapper< T >, 42
- edu.nu.sgm.utils.Reader, 39
 - parseCourseExport, 39
 - parseCourseImport, 40
 - parseGradelItemsExport, 40
 - parseGradelItemsImport, 40
 - parseStudentExport, 41
 - parseStudentImport, 41
 - readCSV, 41
 - writeCSV, 42
- Enrollment
 - edu.nu.sgm.models.Enrollment, 26
- enrollStudent
 - edu.nu.sgm.services.EnrollmentService, 28
- exportCourses
 - edu.nu.sgm.services.CourseService, 15
- exportGradelItems
 - edu.nu.sgm.services.GradelItemService, 36
- fetchCourse
 - edu.nu.sgm.utils.DatabaseManager, 22
- fetchCourses
 - edu.nu.sgm.utils.DatabaseManager, 22
- fetchEnrollment
 - edu.nu.sgm.utils.DatabaseManager, 22, 23
- fetchGrades
 - edu.nu.sgm.utils.DatabaseManager, 23
- fetchStudents
 - edu.nu.sgm.utils.DatabaseManager, 24
- generateEmail
 - edu.nu.sgm.models.Student, 44
- generateReportCard
 - edu.nu.sgm.services.EnrollmentService, 28
- getAllCourses
 - edu.nu.sgm.services.CourseService, 15
- getCategory
 - edu.nu.sgm.models.GradelItem, 31
- getCourse
 - edu.nu.sgm.models.Enrollment, 26
- getCourseById
 - edu.nu.sgm.services.CourseService, 15
- getCourseCode
 - edu.nu.sgm.models.Course, 10
- getCourses
 - edu.nu.sgm.services.StudentService, 45
- getCreditHours
 - edu.nu.sgm.models.Course, 10
- getFeedback
 - edu.nu.sgm.models.GradelItem, 31
- getGrades
 - edu.nu.sgm.services.GradelItemService, 37
- getId
 - edu.nu.sgm.models.Course, 10
 - edu.nu.sgm.models.Enrollment, 26
 - edu.nu.sgm.models.GradelItem, 32
- getInstructor
 - edu.nu.sgm.models.Course, 10
- getMaxScore
 - edu.nu.sgm.models.GradelItem, 32
- getScore
 - edu.nu.sgm.models.GradelItem, 32
- getStudent
 - edu.nu.sgm.models.Enrollment, 27
- getStudentById
 - edu.nu.sgm.services.StudentService, 45
- getStudents
 - edu.nu.sgm.services.CourseService, 16
 - edu.nu.sgm.services.StudentService, 46
- getTitle
 - edu.nu.sgm.models.Course, 11

- edu.nu.sgm.models.GradelItem, 32
- getTotalCourses
 - edu.nu.sgm.services.StudentService, 46
- getTotalStudents
 - edu.nu.sgm.models.Course, 11
 - edu.nu.sgm.services.CourseService, 16
- getWeight
 - edu.nu.sgm.models.GradelItem, 32
- GradelItem
 - edu.nu.sgm.models.GradelItem, 31
- impoGradelItems
 - edu.nu.sgm.services.GradelItemService, 37
- importCourses
 - edu.nu.sgm.services.CourseService, 16
- parseCourseExport
 - edu.nu.sgm.utils.Reader, 39
- parseCourseImport
 - edu.nu.sgm.utils.Reader, 40
- parseGradelItemsExport
 - edu.nu.sgm.utils.Reader, 40
- parseGradelItemsImport
 - edu.nu.sgm.utils.Reader, 40
- parseStudentExport
 - edu.nu.sgm.utils.Reader, 41
- parseStudentImport
 - edu.nu.sgm.utils.Reader, 41
- readCSV
 - edu.nu.sgm.utils.Reader, 41
- removeCourse
 - edu.nu.sgm.services.CourseService, 17
- removeGradelItem
 - edu.nu.sgm.services.GradelItemService, 37
- removeStudent
 - edu.nu.sgm.services.StudentService, 46
- setCategory
 - edu.nu.sgm.models.GradelItem, 33
- setCourse
 - edu.nu.sgm.controllers.CourseViewController, 18
 - edu.nu.sgm.controllers.EnrollStudents, 29
- setCourseCode
 - edu.nu.sgm.models.Course, 11
- setCreditHours
 - edu.nu.sgm.models.Course, 11
- setDialog
 - edu.nu.sgm.controllers.AddCourseController, 7
 - edu.nu.sgm.controllers.AddStudentController, 8
 - edu.nu.sgm.controllers.EnrollStudents, 30
- setFeedback
 - edu.nu.sgm.models.GradelItem, 33
- setID
 - edu.nu.sgm.models.Enrollment, 27
- setId
 - edu.nu.sgm.models.Course, 12
 - edu.nu.sgm.models.GradelItem, 33
- setInstructor
 - edu.nu.sgm.models.Course, 12
- setMaxScore
 - edu.nu.sgm.models.GradelItem, 34
- setScore
 - edu.nu.sgm.models.GradelItem, 34
- setStudent
 - edu.nu.sgm.controllers.StudentViewController, 47
- setStudentData
 - edu.nu.sgm.controllers.AddStudentController, 8
- setTitle
 - edu.nu.sgm.models.Course, 12
 - edu.nu.sgm.models.GradelItem, 34
- setTotalStudents
 - edu.nu.sgm.models.Course, 13
- setWeight
 - edu.nu.sgm.models.GradelItem, 35
- src/main/java/edu/nu/sgm/controllers/AddCourseController.java, 49
- src/main/java/edu/nu/sgm/controllers/AddStudentController.java, 49
- src/main/java/edu/nu/sgm/controllers/CourseViewController.java, 50
- src/main/java/edu/nu/sgm/controllers/EnrollStudents.java, 51
- src/main/java/edu/nu/sgm/controllers/StudentViewController.java, 51
- src/main/java/edu/nu/sgm/models/Course.java, 52
- src/main/java/edu/nu/sgm/models/Enrollment.java, 52
- src/main/java/edu/nu/sgm/models/GradelItem.java, 52
- src/main/java/edu/nu/sgm/models/Student.java, 53
- src/main/java/edu/nu/sgm/services/CourseService.java, 53
- src/main/java/edu/nu/sgm/services/EnrollmentService.java, 54
- src/main/java/edu/nu/sgm/services/GradelItemService.java, 54
- src/main/java/edu/nu/sgm/services/StudentService.java, 55
- src/main/java/edu/nu/sgm/utils/Reader.java, 55
- Student
 - edu.nu.sgm.models.Student, 43
- updateCourse
 - edu.nu.sgm.utils.DatabaseManager, 24
- updateGrades
 - edu.nu.sgm.utils.DatabaseManager, 24
- updateStudent
 - edu.nu.sgm.services.StudentService, 46
 - edu.nu.sgm.utils.DatabaseManager, 25
- writeCSV
 - edu.nu.sgm.utils.Reader, 42