

Software Engineering and Intro to Java
Final Project
Grade Calculation Tool

Jack Titzman

Link to Git Repository:
<https://github.com/jrt0799/GradeCalculator>

Outline

1. Title Page and Link to Git Repository
2. Outline
3. Vision Statement
4. Roles
5. Gantt Diagram
6. Requirements
7. Business Rules
8. Use Cases
11. Use Case Diagrams
12. System Sequence Diagrams
13. System Operations
14. Domain Model
15. Operation Contracts
16. Sequence Diagrams
19. Design Model
20. Justification of GRASPs
21. Use of Design Patterns
22. Hours Spent and Logical Lines of Code

Vision Statement

Many professors at Baylor University do not post their students' grades on Canvas. This forces the students to have to calculate their own grades separately. The goal of this project is to allow students to keep track of their grades without the hassle of having to calculate them themselves.

Roles

(Individual Project)

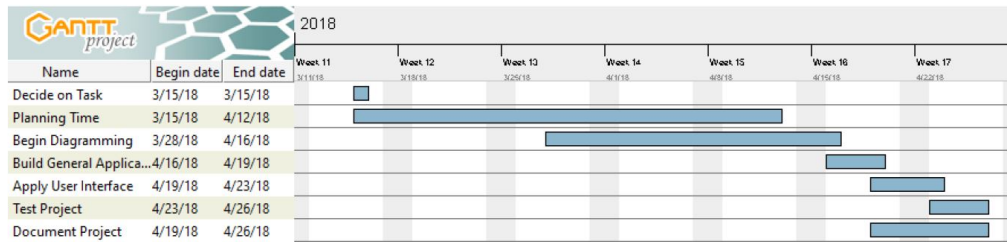
Project Manager - Jack Titzman

Developer - Jack Titzman

Documenter - Jack Titzman

Tester - Jack Titzman

Gantt Diagram



Requirements

Functional Requirements

- User can create a new class
- User can remove a class
- User can create assignments for each class
- User can modify assignments for each class
- User can remove assignments from each class
- System will calculate grade for each class
- System will calculate GPA for each class

Non-Functional Requirements

- User Interface should be easy to navigate
- Grade calculations should be accurate

Business Rules

- A grade of 90.0 or higher is a 4.0 GPA. A grade of 80.0-89.9 is a 3.0 GPA. A grade of 70.0-79.9 is a 2.0 GPA. A grade of 60.0-69.9 is a 1.0 GPA. A grade below 60.0 is a GPA 0.0.
- Assignments that are not included do not count towards the Class grade.
- The percentages for each assignment type in a class should add up to 100 percent.

Use Cases

Use Case: Create a new class

Scope: GradeCalculator application

Actors: User

Precondition: Application is running

Postcondition: New class is added

Main path:

1. User starts application
2. User chooses add new class option
3. User fills out class details
4. User chooses save class
5. System creates class
6. System adds class to application

Alternate paths:

1. Application does not start
3. Class details filled out incorrectly
 - 3.1 Class creation fails
 4. User chooses cancel
 - 4.1 Class creation fails
 5. System fails to create class
 - 5.1 Class creation fails
 6. System fails to add class to application

Use Case: Remove a class

Scope: GradeCalculator application

Actors: User

Precondition: class exists in application

Postcondition: class is removed

Main path:

1. User starts application
2. User selects a class to remove
3. User chooses remove class
4. System removes Class from application

Alternate paths:

1. Application does not start
2. No classes exist in the application
4. System fails to remove class

Use Case: Create a new assignment for a class

Scope: GradeCalculator Application

Actors: User

Precondition: Class for assignment exists

Postcondition: New assignment is added to class

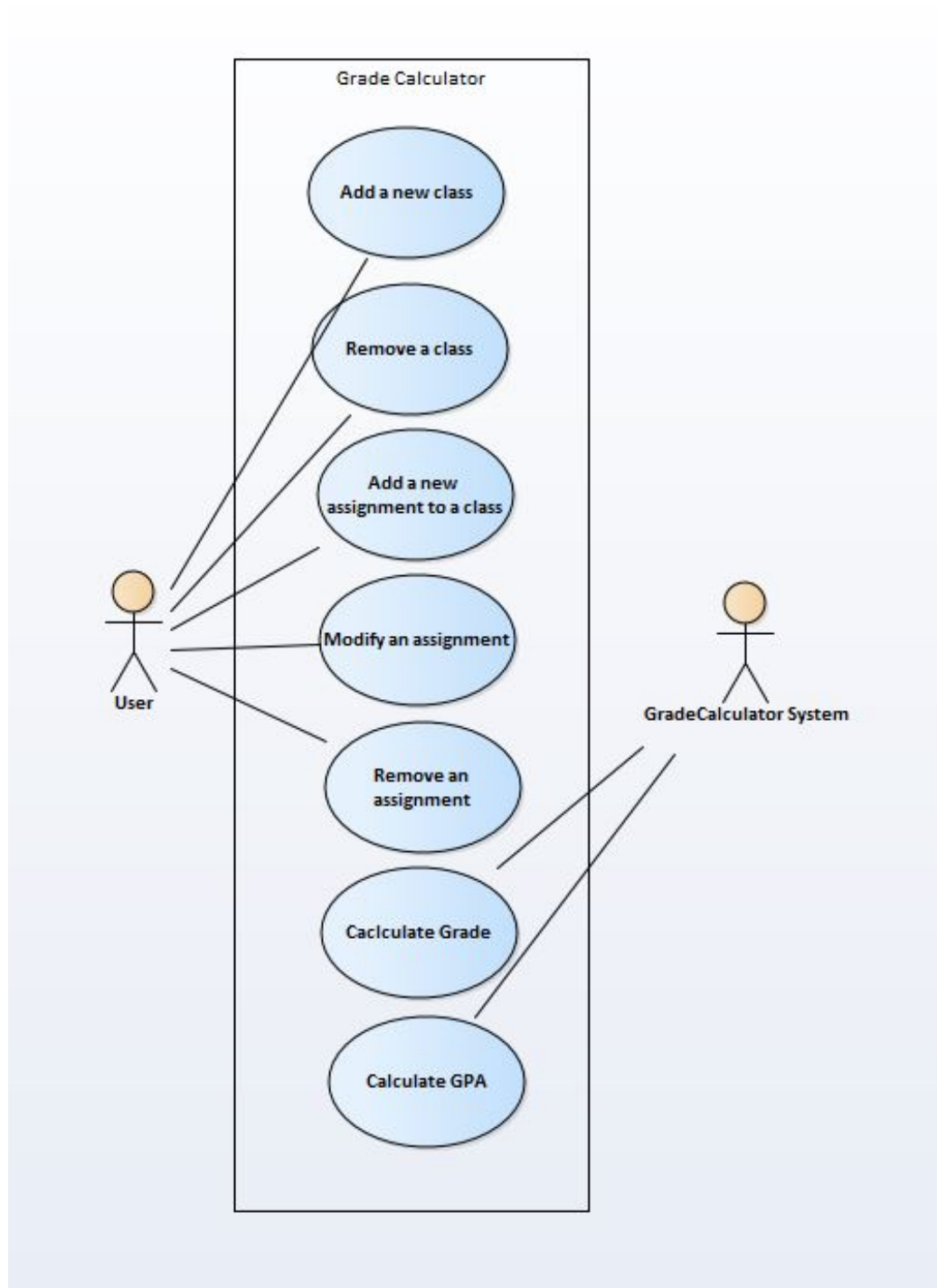
Main path:

1. User starts application
2. User selects a class
3. User chooses add new assignment
4. User enters assignment details
5. User chooses add assignment
6. System creates assignment
7. System adds assignment to application

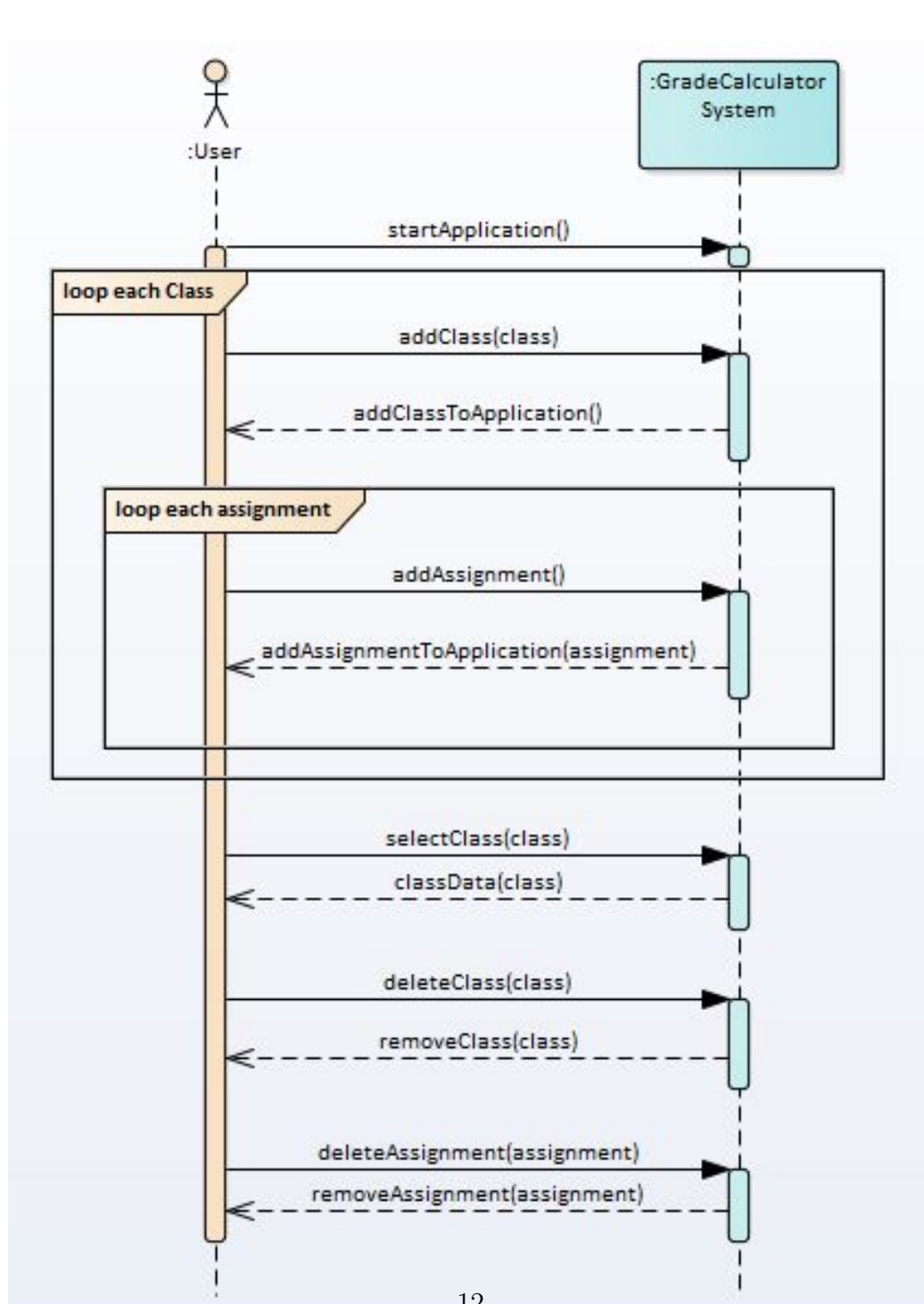
Alternate paths:

1. Application does not start
2. No classes exist in the application
4. Assignment details filled out incorrectly
 - 4.1 Assignment creation fails
 5. User chooses cancel
 - 5.1 Assignment creation fails
 6. System fails to create assignment
 7. System fails to add assignment to application

Use Case Diagram



System Sequence Diagrams



System Operations

`addClassToApplication(class)`

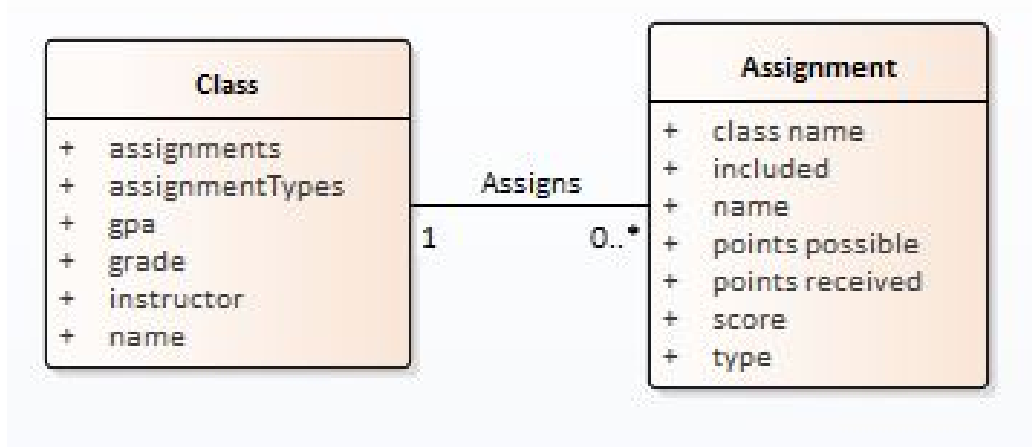
`addAssignmentToApplication(assignment)`

`classData(class)`

`removeClass(class)`

`removeAssignment(assignment)`

Domain Model



Operation Contracts

Operation: `addClassToApplication(class)`

Cross References: Create a new class

Preconditions:

There is a class to add to the application

Postconditions:

The class has been added to the application

Operation: `addAssignmentToApplication(class)`

Cross References: Create a new assignment

Preconditions:

There is an assignment to add to the application

Postconditions:

The assignment has been added to the application

Operation: `classData(class)`

Cross References: Create a new class, Remove a class

Preconditions:

There is a class in the application

Postconditions:

The user will have the selected class' data

Operation: `removeClass(class)`

Cross References: Remove a class

Preconditions:

There is a class to remove from the application

Postconditions:

The class has been removed from the application

Operation: `removeAssignment(assignment)`

Cross References: Remove an assignment

Preconditions:

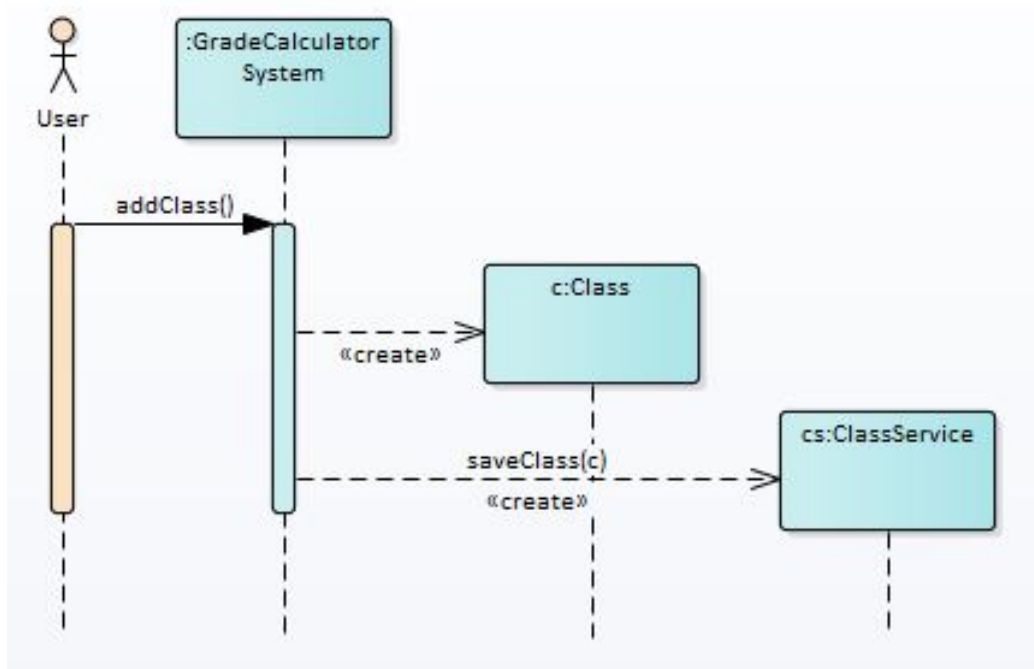
There is an assignment to remove from the application

Postconditions:

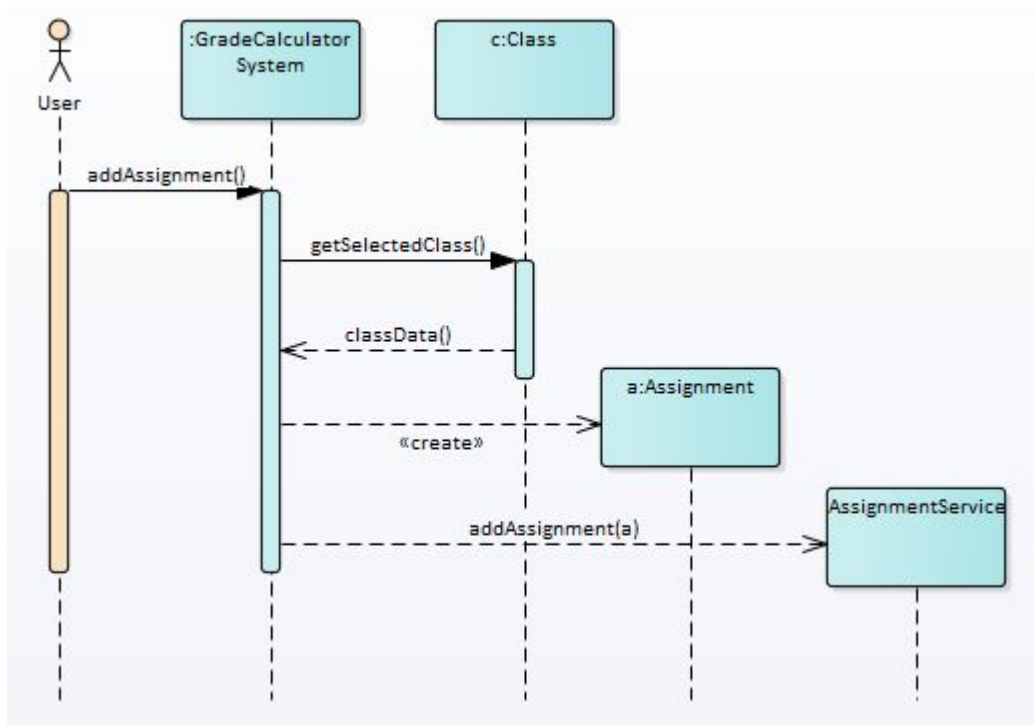
The assignment has been removed from the application

Sequence Diagrams

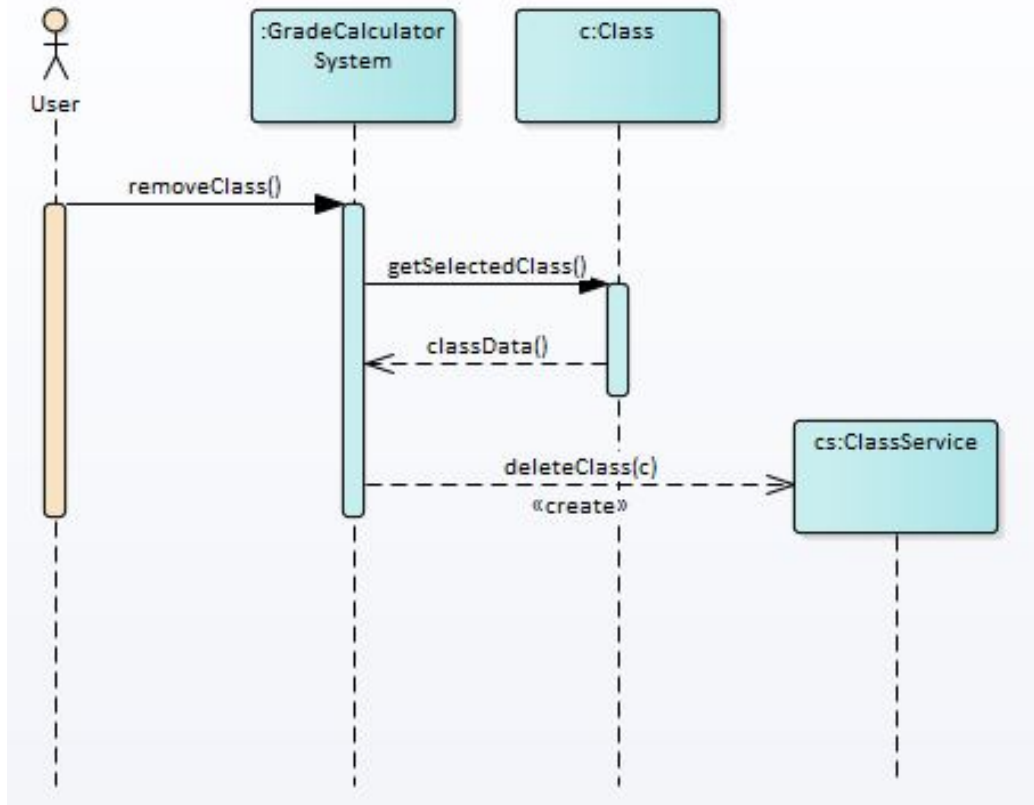
Use Case: Create a new class



Use Case: Create a new assignment



Use Case: Remove a class



Design Model

Justification of GRASPs

Any time a user interacts with a component of the user interface, the `ServiceResponse` is used. This is using the GRASP Controller because it handles the system events.

The classes are designed such that an `AssignmentService` does not know about a `ClassService` and vice versa. This is using the GRASP Don't talk to strangers.

The classes are broken up into several different parts such that one element is not strongly connected to, has knowledge of, or relies upon other elements. This is using the GRASP Low Coupling.

The classes are broken up into several different parts such that each element has strongly related and focused responsibilities. This is using the GRASP High Cohesion.

The previous two uses of GRASPs talked about Low Coupling and High Cohesion. This was achieved by making artificial objects (`ClassDAO`, `AssignmentDAO`). This is using the GRASP Pure fabrication.

Use of Design Patterns

Factory - ConnectionFactory (creates new instances of database connection)

Observer - Various objects: ClassListModel, AssignmentService, ClassService, MainView (notifiesObservers when changes are made)

Command - ServiceResponse from different swing components

Strategy - SQLAssignmentDAO, SQLClassDAO (wraps the save, update, delete, and accessor methods including calculating grade and gpa)

Proxy - Construct UI elements as placeholders and update them later.

Hours Spent

50 hours.

Logical Lines of Code

1057

Java LOC Counter			Choose File
Include in Total	LOC Count	Filename	
<input checked="" type="checkbox"/>	14	D:\Computer Science\CSI 3130 Intro to Java\Final Project\GradeCalculator\src\connection\ConnectionFactory.java	
<input checked="" type="checkbox"/>	37	D:\Computer Science\CSI 3130 Intro to Java\Final Project\GradeCalculator\src\data\Assignment.java	
<input checked="" type="checkbox"/>	8	D:\Computer Science\CSI 3130 Intro to Java\Final Project\GradeCalculator\src\data\AssignmentDAO.java	
<input checked="" type="checkbox"/>	36	D:\Computer Science\CSI 3130 Intro to Java\Final Project\GradeCalculator\src\data\Class.java	
<input checked="" type="checkbox"/>	8	D:\Computer Science\CSI 3130 Intro to Java\Final Project\GradeCalculator\src\data\ClassDAO.java	
<input checked="" type="checkbox"/>	85	D:\Computer Science\CSI 3130 Intro to Java\Final Project\GradeCalculator\src\data\SQLAssignmentDAO.java	
<input checked="" type="checkbox"/>	181	D:\Computer Science\CSI 3130 Intro to Java\Final Project\GradeCalculator\src\data\SQLClassDAO.java	
<input checked="" type="checkbox"/>	24	D:\Computer Science\CSI 3130 Intro to Java\Final Project\GradeCalculator\src\model\ClassListModel.java	
<input checked="" type="checkbox"/>	5	D:\Computer Science\CSI 3130 Intro to Java\Final Project\GradeCalculator\src\model\ObservantTableModel.java	
<input checked="" type="checkbox"/>	46	D:\Computer Science\CSI 3130 Intro to Java\Final Project\GradeCalculator\src\model\SelectedAssignmentsTableModel.java	
<input checked="" type="checkbox"/>	27	D:\Computer Science\CSI 3130 Intro to Java\Final Project\GradeCalculator\src\runner\MainViewRunner.java	
<input checked="" type="checkbox"/>	41	D:\Computer Science\CSI 3130 Intro to Java\Final Project\GradeCalculator\src\service\AssignmentService.java	
<input checked="" type="checkbox"/>	8	D:\Computer Science\CSI 3130 Intro to Java\Final Project\GradeCalculator\src\service\AssignmentServiceInterface.java	
<input checked="" type="checkbox"/>	55	D:\Computer Science\CSI 3130 Intro to Java\Final Project\GradeCalculator\src\service\ClassService.java	
<input checked="" type="checkbox"/>	9	D:\Computer Science\CSI 3130 Intro to Java\Final Project\GradeCalculator\src\service\ClassServiceInterface.java	
<input checked="" type="checkbox"/>	11	D:\Computer Science\CSI 3130 Intro to Java\Final Project\GradeCalculator\src\service\ServiceResponse.java	
<input checked="" type="checkbox"/>	85	D:\Computer Science\CSI 3130 Intro to Java\Final Project\GradeCalculator\src\test\Tests.java	
<input checked="" type="checkbox"/>	69	D:\Computer Science\CSI 3130 Intro to Java\Final Project\GradeCalculator\src\view\AddAssignmentPopup.java	
<input checked="" type="checkbox"/>	101	D:\Computer Science\CSI 3130 Intro to Java\Final Project\GradeCalculator\src\view\AddClassPopUp.java	
<input checked="" type="checkbox"/>	207	D:\Computer Science\CSI 3130 Intro to Java\Final Project\GradeCalculator\src\view\MainView.java	
Total: 1057			Clear