Software Engineering and Intro to Java
Final Project
Grade Calculation Tool

Jack Titzman

Outline

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

Domain Model

Operation Contracts

Sequence Diagrams

Design Model

Justification of GRASPs

Use of Design Patterns