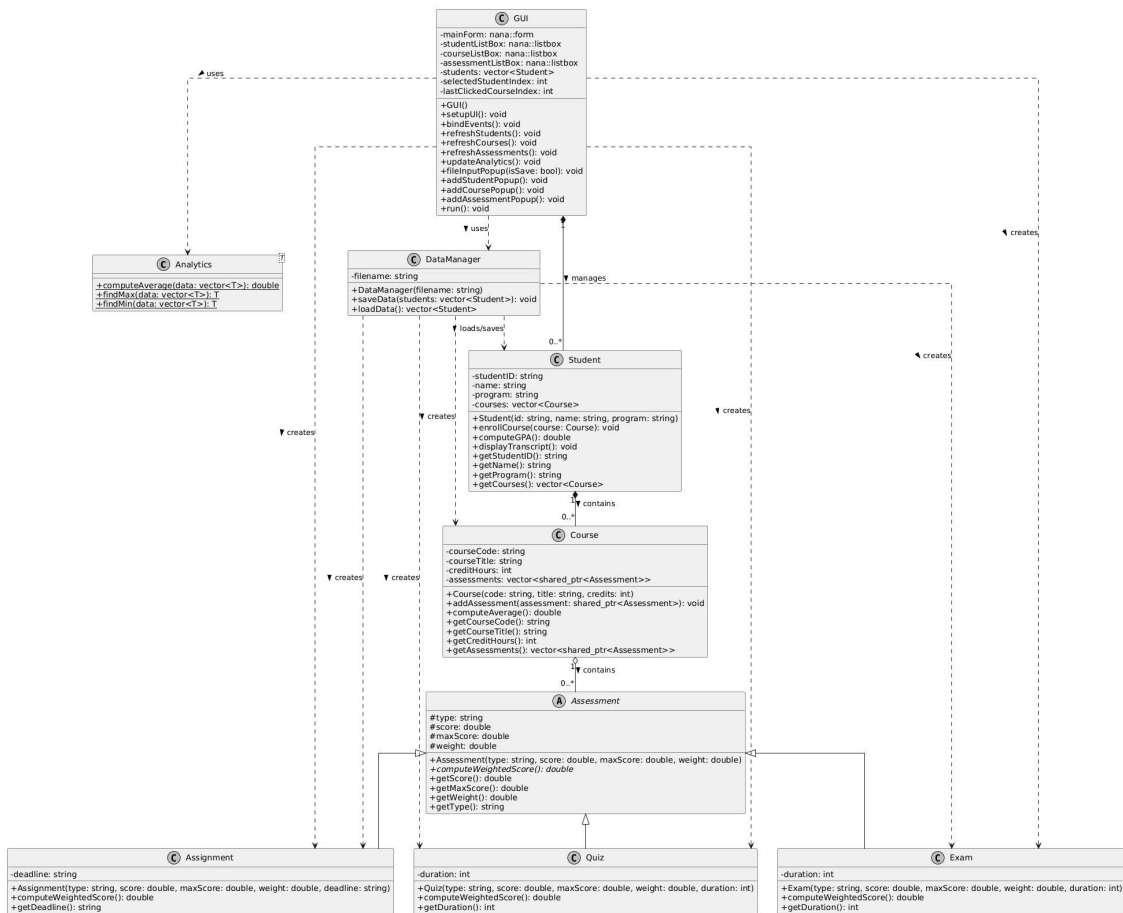# Application Documentation: Student Assessment System

Overview

This document provides technical documentation for the **Student Assessment System** application. It is a program that displays grades and course details of students.

Key features of this application include:

- Student Management: Add Students with unique ids, names, and programs
- Course Management: Enroll students in multiple courses and track credit hours
- Display Analytics in a real-time dashboard including average.
- Save and Load into and from .txt files.

Class Design

UML Class Diagram

GUI
- -mainForm: nana::form
- -studentListBox: nana::listbox
- -courseListBox: nana::listbox
- -assessmentListBox: nana::listbox
- -students: vector<Student>
- -selectedStudentIndex: int
- -lastClickedCourseIndex: int
- +GUI()
- +setupUI(): void
- +bindEvents(): void
- +refreshStudents(): void
- +refreshCourses(): void
- +refreshAssessments(): void
- +updateAnalytics(): void
- +fileInputPopup(isSave: bool): void
- +addStudentPopup(): void
- +addCoursePopup(): void
- +addAssessmentPopup(): void
- +run(): void

Analytics
- +computeAverage(data: vector<T>): double
- +findMax(data: vector<T>): T
- +findMin(data: vector<T>): T

DataManager
- -filename: string
- +DataManager(filename: string)
- +saveData(students: vector<Student>): void
- +loadData(): vector<Student>

Student
- -studentID: string
- -name: string
- -program: string
- -courses: vector<Course>
- +Student(id: string, name: string, program: string)
- +enrollCourse(course: Course): void
- +computeGPA(): double
- +displayTranscript(): void
- +getStudentID(): string
- +getName(): string
- +getProgram(): string
- +getCourses(): vector<Course>

Course
- -courseCode: string
- -courseTitle: string
- -creditHours: int
- -assessments: vector<shared_ptr<Assessment>>
- +Course(code: string, title: string, credits: int)
- +addAssessment(assessment: shared_ptr<Assessment>): void
- +computeAverage(): double
- +getCourseCode(): string
- +getCourseTitle(): string
- +getCreditHours(): int
- +getAssessments(): vector<shared_ptr<Assessment>>

Assessment
- #type: string
- #score: double
- #maxScore: double
- #weight: double
- +Assessment(type: string, score: double, maxScore: double, weight: double)
- +computeWeightedScore(): double
- +getScore(): double
- +getMaxScore(): double
- +getWeight(): double
- +getType(): string

Assignment
- -deadline: string
- +Assignment(type: string, score: double, maxScore: double, weight: double, deadline: string)
- +computeWeightedScore(): double
- +getDeadline(): string

Quiz
- -duration: int
- +Quiz(type: string, score: double, maxScore: double, weight: double, duration: int)
- +computeWeightedScore(): double
- +getDuration(): int

Exam
- -duration: int
- +Exam(type: string, score: double, maxScore: double, weight: double, duration: int)
- +computeWeightedScore(): double
- +getDuration(): int

Implementation Highlights

The following are significant aspects of the application's implementation:
- Polymorphism and Smart Pointers: The system uses std::shared_ptr for the assessment vector allowing it to store quiz,exam and assignments while preventing object slicing.
- Nana GUI Library: The interface is created using Nana using widgets and listboxes for displaying hierarchical data and handling button events.
- Templates: The Analytics class uses generics to be type safe and allows the program to potentially be easily adjustable for different use.

Challenges and Solutions

Grades were calculated incorrectly dividing the already weighted score twice, I removed this mistake and it displayed correctly.

Screenshots

Student Assessment System

Students		
ID	Name	Course
0001	Chloe Capriotti	CETCS
0002	Micheal Smith	CSA
0003	Mimi Sloan	PN

Courses (Multi-select allowed)		
Code	Title	Avg
CST8219	C++ Programming	51.6

Assessments			
Type	Score	Wgt	Details
Quiz	9/10	5	90m
Assignment	17/20	30	2025/11/29
Exam	27/50	40	90m

Student Analytics

Student: Chloe Capriotti (0001)

Selected Courses Analysis

Courses: 1
Average: 51.60%
Highest: 51.60%
Lowest: 51.60%

Refresh Stats

[Add Student] [Add Course] [Add Assess.]   [Save Data] [Load Data]

Student Assessment System

Students		
ID	Name	Course

Courses (Multi-select allowed)		
Code	Title	Avg

Assessments			
Type	Score	Wgt	Details

Student Analytics

Select a student to view performance.

Refresh Stats

[Add Student] [Add Course] [Add Assess.]   [Save Data] [Load Data]