Assignment 10

Member Name: Chao Fang (fang.chao1@husky.neu.edu)
Haonan Zhao (zhao.hao@husky.neu.edu)
Yuecan Fan (fan.yue@husky.neu.edu)
Zhen Wei (weizhen167@gmail.com)
Team number: 6

Contents

Use Case: tracking grades for courses

1. Course instructors can select the section of courses, then give each student their assignment grade. By using Grade Book System, they can track grades within their courses.
2. Grade book calculates and stores general information about each assignment (e.g., homework assignment, lab, quiz, test, participation), including average grade (median, mode if required), feedback, comment.
3. Grade book stores basic information about each student in each specific section.

Requirements

By selecting particular course and section, course instructor is able to look over basic information about both students and assignments in this section.

The system shall allow the instructor to input the new assignment (homework assignment, lab, quiz, test, participation) of the course section.

Various section instructors also can have discussions through this system to decide exactly what is excepted by the course instructors.

System manager should be able to generate initial data information about each course and section.

System manager should also be able to input basic information about each student. Information of students should provide student ID, student's name, and student's Grade.

Student's grade can be either EmptyGrade or AavailableGrade. Instructor is able to add/input availableGrade for each students.

The system shall calculate each student's current grade and average grade for a particular assignment. For each assignments, system shall also provide instructors with feedback based on students' grade (e.g., the number of students who fails test, difficulty of each homework or test).

Methods for calculation and providing feedback can be updated by system manager.

Domains:

User (Instructor) interface: Input module (new assignment, grade for each students) reporting the results (current grade for each student, average grade for recent assignment, discussion for specific expectation)

Manager interface: Input module for providing initial course section for a specific instructor, and also provide the basic data about all students in this section.

Algorithm: Compute the average for a particular assignment, and each student current grade, and reports to the instructor.

Relations:

User (Instructor) interface: interacts with the instructor, let them create new assignments, allows for corrections, interacts with the system manager to get students' information, and give them the lastest grade, recieves from system manager the average grades and compare with the expect value they discuss with others before.

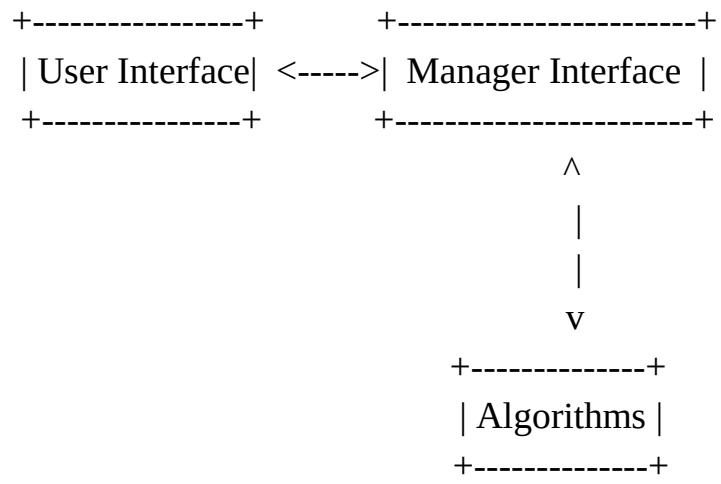
Manager interface: Allow the manager to initialize and to vertify the information about the students and each of their assignment.
Algorithm: Updated by manager. Requests latest grade data, report its results to instructor.

Constraints:

The grade book must include basic information for all students who registered in the instructor's course section.

The grade for each assignment must be greater than 0.

Module dependency diagram:



Data model:

Course: given as a list of course.

Course Id: int, the unquie id for each course

Course Name:String,the name of course

Section: given as a list of section

HashMap: collections of Student and Grade

Student: student that represent the basic data for each of student, which include Id, firstName, lastName

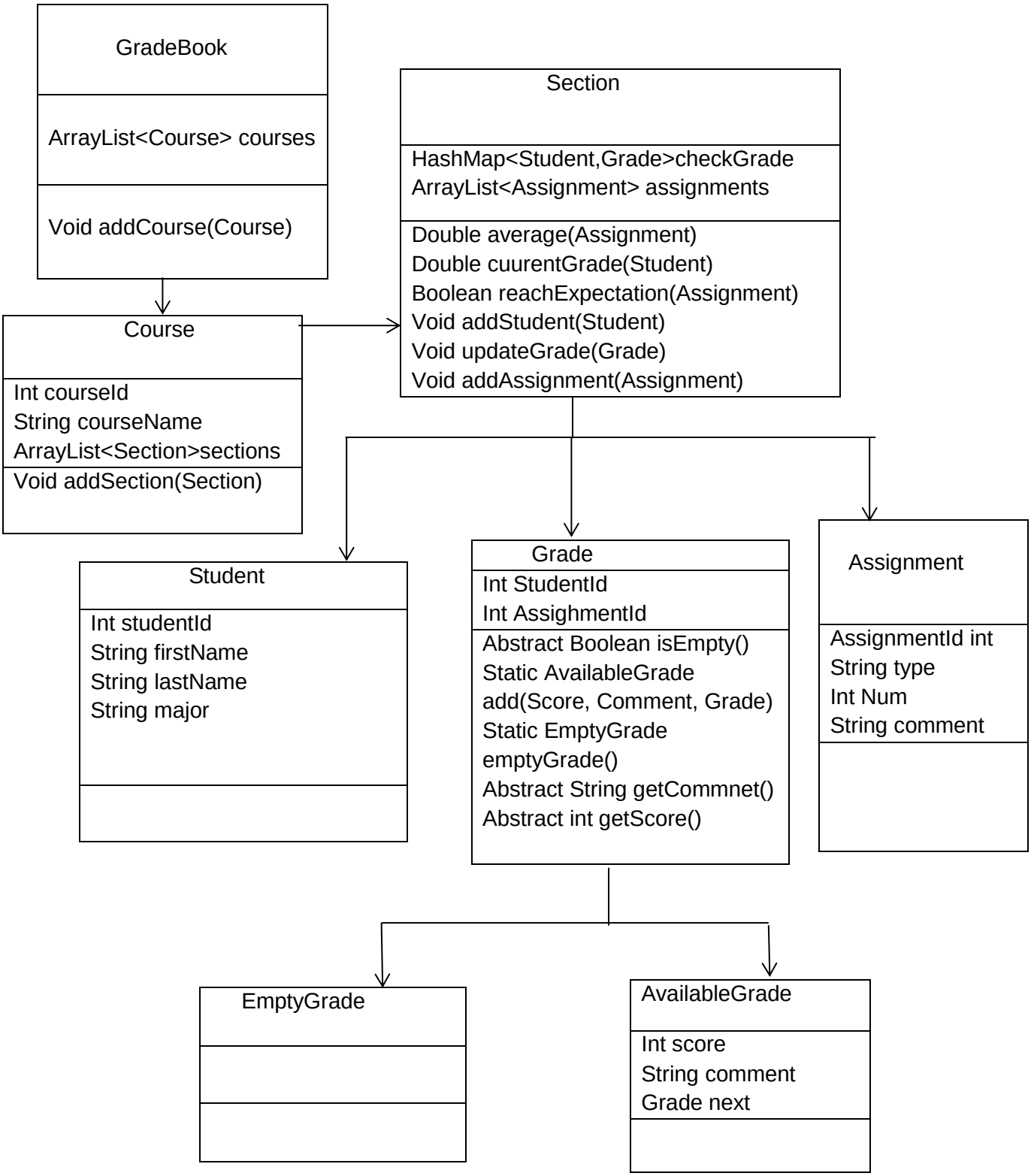
Grade: list of grade of each student. Grade can be either EmptyGrade or AvailableGrade.

AvilableGrade: a combinated list of Scores, comments, and next grade

Empty (Grade): an empty list that represent empty grade

Assignment: list of assignment including tpye, number, comments, assignment Id.

UML diagram:


GradeBook
ArrayList<Course> courses
Void addCourse(Course)
Section
HashMap<Student,Grade>checkGrade
ArrayList<Assignment> assignments
Double average(Assignment)
Double cuurentGrade(Student)
Boolean reachExpectation(Assignment)
Void addStudent(Student)
Void updateGrade(Grade)
Void addAssignment(Assignment)
Course
Int courseId
String courseName
ArrayList<Section>sections
Void addSection(Section)
Student
Int studentId
String firstName
String lastName
String major
Grade
Int StudentId
Int AssighmentId
Abstract Boolean isEmpty()
Static AvailableGrade
add(Score, Comment, Grade)
Static EmptyGrade
emptyGrade()
Abstract String getCommnet()
Abstract int getScore()
Assignment
AssignmentId int
String type
Int Num
String comment
EmptyGrade
AvailableGrade
Int score
String comment
Grade next

Team member contributed to the assignment:

To complete this assignment, we discussed it online and written it part by part by using a google doc.

Chao Fang and Wei Zhen firstly draw the UML diagram for this assignment.

By the same time, Yuecan Fan and Haonan Zhao wrote the documents (Use Case& Requirements, Domains,Relations & Constraints ,Module dependency diagram& Data model). Then, we switched the task, Chao Fang and Wei Zhen looked through the writing documents and added, revised some parts. Finally, Yuecan Fan and Hannan Zhao put the UML diagram and revised part together to hand it. So each of us contributed equally and know very well about this assignment.