Assignment 11

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 create an empty gradebook and initialize MyGradeBook with file, string, then give each student their assignment grade. By using Grade Book System, they can track grades within their courses.

2. Grade book stores basic information about each student.

3. Grade book calculates and stores each student's assignments (e.g. homework assignment, lab, quiz, test, participation) grade and grade general information (median,min,max).

4. Grade book can calculate current grade of each student.

Requirements

After initializing, course instructor is able to look over basic information about both students and assignments.

The system shall allow the instructor to input new, such as new student, new assignment (homework assignment, lab, quiz, test, participation), new grade for specific student's assignment.

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 username, student's firstname, lastname and advisor, expected graduation year.

Grade class contains Assignment Name, Total points, Percent of semester grade.

The system shall calculate each student's current grade and average, min median, max grade for a particular assignment.
Methods for calculation can be updated by system manager.

Domains:

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

Manager interface: Input module for providing the basic data about all students in this section.

Algorithm: Initialize from file, process file, output gradebook, compute the average, min, max, median 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, current grades.

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.

If the students didn't submit the assignment, he will get zero.

Module dependency diagram:

```
+----------------+          +----------------------+
| User Interface|  <----->|  Manager Interface  |
+----------------+          +----------------------+
                                       ^
                                       |
                                       |
                                       v
                            +--------------+
                            | Algorithms |
                            +--------------+


----------------------------------------------------------
```
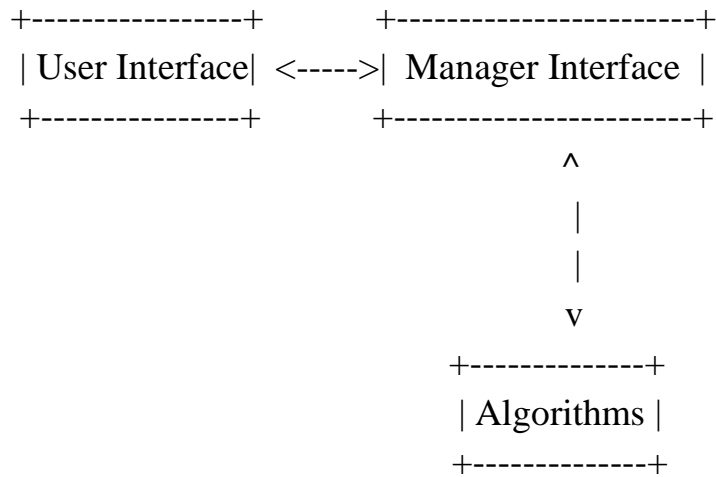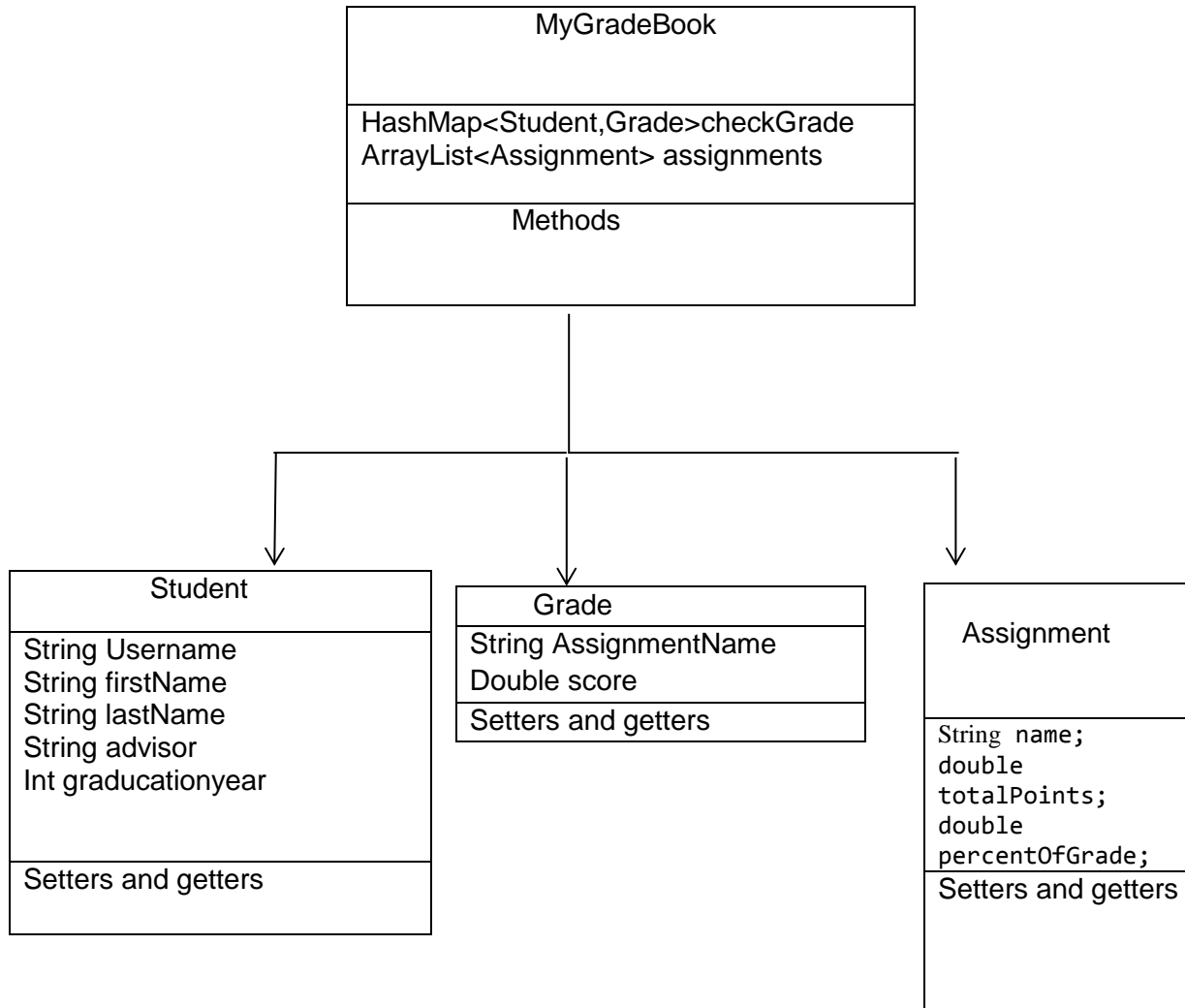
Data model:

HashMap: collections of Student and Grade

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

username, firstName, lastName, advisor, graduation year

Grade: list of grade of each student. The grade include assignmentName, score.

Assignment: list of assignment including Assignment, Total points, Percent of semester grade

----

UML diagram:

MyGradeBook
HashMap<Student,Grade>checkGrade ArrayList<Assignment> assignments
Methods

Student
String Username String firstName String lastName String advisor Int graducationyear
Setters and getters

Grade
String AssignmentName Double score
Setters and getters

Assignment
String name; double totalPoints; double percentOfGrade;
Setters and getters

Team member contributed to the assignment:

To complete this assignment, we need to write code of each method in myGradeBook, and connect it to GUI.

Yuecan Fan and Wei Zhen designed the GUI part.

By the same time, Chao Fang and Haonan Zhao wrote the method codes.

Finally, we put them together to connect the methods with GUI interface.