CS3716 Assignment 2

Diagrams & Skeleton Code

Monday, October 6, 2014

Kayla Blain - 201105103

Christopher Doyle - 200833176

Instructor: Adrien Fiech

1. **DOMAIN MODEL:**

1

0..\*

1

1

*classList*

classList : map<string, ArrayList<list>>

university : string

assignment : int

student

FullName : string

studentNumber : int

grades : list

skills : list

weeklyAvailability : map<string, string>

preferPartners : list[maxPreferPartners]

avoidPartners : list[maxAvoidPartners]

1

1

groupList

groupList : map <String, list>

1

0..\*

1

groupSpecs

deadline : string

assignmentInitialized : Boolean

maxPreferPartners : int

maxAvoidPartners : int

maxGroupSize : int

balanceFactors : list

classList = list

scoreGenerator

score : double

1

1. **SEQUENCE DIAGRAM:**

9: Create Group List

5: Scoring data

6: Scores

4: Next Student

3: Student Information

1: Group Specifications

8: Close Deadline

7: Map of Students

2: Assignment Initialized

ScoreGenerator

GroupList

ClassList

Student

GroupSpecs

1. **CODE ADAPTABILITY:**

The main decisions that will make our software amendable to future changes is that (a) it will allow the instructor and students to modify the information that they have entered, (b) it allows for multiple balance factors, (c) it allows the instructor to make necessary changes to the groups and (d) it allows the instructor to use the program for grouping multiple classes. Those four programming choices allow the correction of human error and modifications for future assignments.

a)

Allowing the instructor to modify an assignment may include changing the deadline, changing which skills need self-evaluation from the students (based on project requirements) or changing the balance factor by which they’d like to sort the groups. We can allow the instructor to make necessary changes by having all the assignment information stored externally on an XML File and updating the file when changes are made. Similar implementation occurs with the Student class by saving all the students as array lists and writing them to XML files in the form of a class list. If a student needs to change the times at which they’re available or have made a mistake when typing their student number then rewriting the XML file will allow them to change just the information they’re looking for and not require them recreate a profile.

b)

The choice of one or more balance factors allows the Instructor to have multiple criteria on which the groups may be sorted. Information from assignment to assignment will differ so having multiple balance factors will allow the instructor to not always sort by schedule, but maybe sort by schedule and skill, or skills and grades, etc.

c)

After the GroupList class generates an optimal group listing, the instructor may want to change which students are paired together, he will be prompted to enter the names of two students, the GroupList class will then modify the XML file to hold those two students in a group together then take their two left over partners and pair them together. The system will notify the instructor if the remaining two students have any serious conflicts.

d)

Finally, allowing the instructor to input multiple classes (ie. CS2711, CS3715 and CS3716) may be done by creating a separate class which takes file paths as arguments (instructor input files) and separates the classes they have but still allows that information to be stored in XML files in closely located areas for easy access/retrieval.

**SKELETON CODE:**

Skeleton codes attached in submission email as .java files.

Files include: student.java, groupSpecs.java, groupList.java, classList.java, and scoreGenerator.java