**Analyze and Design**

**Assignment J1.L.P0001.-Student-Management-500LOC.pdf**

**1- Analyze:**

From the problem description, the following classes are indentified:

|  |  |
| --- | --- |
| Concept | Class |
| Menu | class Menu extends java.util.ArrayList |
| Student | class Student |
| List of students | class StudentList extends java.util.ArrayList |
| Subject | class Subject |
| List of subjects | class SubjecttList extends java.util.ArrayList |
| Grade | class Grade |
| List of Grades | class GradeList extends java.util.ArrayList |
| Program for managing grades | Class GradeManager |

**2- Class Design:**

**public class Menu extends ArrayList**{

int getUserChoice();

}

**public class Student**{

String stdId, firstName, lastName, gender;

Date DOB;

String email, phone

boolean canDelete=true;

}

**public class StudentList extends ArrayList**{

**private int search (String stdId);** // stdudent ID duplicates are not allowed

**public void addStudent(){**

* Student ID duplicates are not allowed;
* First name, last name. gender are not null;
* Date of birth (DOB) must be valid
* Email must be valid (x@xxx.xxx)
* Phone: 1—12 digits

}

**public void updateStudent(){**

* Input student ID
* int pos = search(stdID);
* if (pos<0) print out message “Student does not exist”
* else{
  + First name, last name. gender, DOB, email, phone may be updated.
  + If user enter a null string, appropriate update is omitted
  + New email must be valid ([x@xxx.xxx](mailto:x@xxx.xxx))
  + New phone number muswt be valid (10-12 digits)

}

}

**public void deleteStudent()**{

Input stdID;

int pos = search(stdID);

if (pos<0) print out message “Student does not exist”;

else if (this.get(pos).canDelete==false) print out message “Student can not be removed”;

else {

remove(pos);

print out message “Student “ + stdID + “ has been removed”;

}

}

}

**public class Subject**{

String subID, subName;

int credit;

boolean canDelete = true;

}

**public class SubjectList extends ArrayList**{

**private int search (String subId);** // subject ID duplicates are not allowed

**public void addSubject(){**

* Subject ID duplicates are not allowed;
* Suject name is not null;
* 0 < Credit <= 30

}

**public void updateSubject(){**

* Input sub D
* int pos = search(subID);
* if (pos<0) print out message “Subject does not exist”
* else{
  + Subject name may be updated if **new name is not null**.
  + 0 < new credit <=30

}

}

**public void deleteSubject()**{

Input subID;

int pos = search(subID);

if (pos<0) print out message “Subject does not exist”;

else if (this.get(pos).canDelete==false) print out message “Subject can not be removed”;

else {

remove(pos);

print out message “Subject “ + subID + “ has been removed”;

}

}

}

**public class Grade**{

Student std;

Subject sub;

double lab, test, FE;

public double average () { return 0.3\*lab + 0.2\*test + 0.5\*FE; }

}

**public class GradeList extends ArrayList**{

StudentList stdList;

StudentList subList;

**public int search** (String stdID, String subID) ;

**public int searchStudent** (String stdID) ;

**public int searchSubject** (String subID) ;

**public void addNewGrade()**{

String stdID, subID;

int stdPos= -1, subPos=-1, gradePos=-1;

do {

do { input stdID;

stdPos = stdList.search(stdID);

} while (stdPos<0);

do { input subID;

subPos = subList.search(subID);

} while (subPos<0);

gradePos = this.serach(stdID, subID);

if (gradePos>=0) print out “This grade existed!”

} while (gradePos>=0);

Input lab, test, FE ( 0<=score<=10);

Create new grade (stdList.get(stdPos), subList.get(sobPos), lab, test, FE);

Add new grade to the grade list;

stdList.get(stdPos).canDelete = false;

subList.get(subPos).canDelete = false;

print out “A new grade have been added!”

}

**public void printStudentReport()**{

String stdID;

Input stdID;

Int gradePos = searchStudent(stdID);

If (gradePos<0) print out “No report can be supported ”

Else{

Print out “Student ID: “ + stdID;

Student st = this.get(gradPos).std;

Print out “Stdent name: “ + st.lastName + “ “ + std.firstName;

Print out table template “No Subject Average Status”

int count =1;

For (Grade g: this) {

If (stdID.equals(g.std.stdID) {

Print out count;

Print out g.sub.subName;

double avg = g.average();

print out avg;

Print out avg>=5? “Pass\n” : “ Fail\n”;

count++;

}

}

}

**public void printSubjectReport()**{

String subID;

Input subID;

Int gradePos = searchSubject(subID);

If (gradePos<0) print out “No report can be supported ”

Else{

Print out “Subject ID: “ + subID;

Subject sub = this.get(gradPos).sub;

Print out “Subject name: “ + sub.subName;

Print out table template “No Student ID Student name Average Status”

int count =1;

For (Grade g: this) {

If (subID.equals(g.sub.subID) {

Print out count;

Print out g.std.stdID;

Print out g.std.lastName;

Print out g.std.firstName;

double avg = g.average();

print out avg;

Print out avg>=5? “Pass\n” : “ Fail\n”;

count++’

}

}

}

}

**public class GradeManager{**

**Data : Menu mnu;**

**StudentList stdList;**

**SubjectList subList;**

**GradeList gradeList (stdList, subList);**

**Method: public static void main(String args[])**

**}**