

Design Document for Homework Programming Assignment

Part 1:

There are 3 schools in the dropdown menu which the user can select from: Fast elementary, Fast middle school and Fast high school. Initially all the students are displayed with each student having a class id which is equal to the student's school id.

```
class=${student.classEnrollments[0].school.schoolId}>${student.lastName},  
${student.firstName}
```

The list of students is changed dynamically without reloading or navigating from the page with the help of JQuery. As the user selects one of the schools, the onchange method is called that passes the school id to a function which uses JQuery to hide or show the list of students studying in that school.

```
if (select.value == '1') {  
    $('#1').show();  
    $('#2').hide();  
    $('#3').hide();  
}
```

Part 2:

Each student has the information about which classes he is enrolled in in front of his/her name. The student class has a number of attributes in which one of the attributes is a list classes in which the student is enrolled. The course and grade level are accessed to display the class information.

```
${student.classEnrollments[0].course} - ${student.classEnrollments[0].gradeLevel}
```

Design Directions:

One of the design direction that could've been used was AJAX. Each time the user chose a school, the database could've been queried for the students studying in that particular school and a part of page could've been dynamically modified. This design was rejected as it was too complex for this simple task.

Assumptions:

We are assuming that the number of schools is small so that we can separately hide and show students of each school. We are also assuming that the structure of the student class object doesn't change because we are accessing an attribute of an array which itself is the attribute of an object.

Testing:

We can test this part by selecting all the schools one at a time. We can also try many different permutations of choosing a school and we can also choose back the option of all the schools.

Scalability:

For scaling over 200 student records, we can include a loop to show and hide the students of different schools so that we don't have to write the code for each and every school.

For scaling over 20,000 student records, it will be better to use AJAX to access only that data which is required and then dynamically update the page instead of accessing all the data at once.