Assignment 1023, 1025

With this assignment, we can wrap up our time with Headmaster’s web service layer.

Outcomes

Assignment 1023 affects your proficiency measures for outcomes 1a–1c, 3b–3d, and 4a–4f. So, yes, this should help make up for any proficiency weaknesses from the previous assignment.

Assignment 1025 is a little more focused, and thus only affects outcomes 1a, 3d, and 4a–4f.

Background Reading

The programming readings have been mentioned before. Although not directly related, the current readings involve the relational algebra and calculus, and that would be in Chapter 6 of Silberschatz/Korth/Sudarshan.

For Submission, 1023

Unit Test Coverage and Other Loose Ends

Add some spit and polish (and fixes!) to your adopted Headmaster issue from the previous assignment. Most of you need better unit test coverage, so that is a minimum. Some of you have additional loose ends (as stated in your last round of feedback), so cover those here also.

For Submission, 1025

Split resource Into service and resource

As has been pointed out in class, the current Headmaster service code base combines pure service (business logic) functionality with the HTTP interface (REST) code. As a final exercise for Headmaster, break this monolithic layers into two:

• A new service package should be created
• Resource interface/implementations become straight-up classes; these classes take services as instance variables
• Services should be broken into interfaces and implementations; these take DAOs or other services as instance variables
• Create/retain superclasses (abstract or not) as you see fit (this may take some coordination)
• Make sure to update the Spring context file to reflect the new object structure

Note that this refactor will go much more smoothly once there is a strong bank of unit tests that can catch potential problem, which is why the work has been partitioned strictly.

For both tasks, commit and push to your forks, then send me a pull request when ready.