ABC

Sprint 1 Retrospective

Team Members

Gaurav Srivastava	— srivast6@purdue.edu
Naveen Ganessin	— nganessi@purdue.edu
Alex Rosenberg	— rosenbea@purdue.edu
Michael Schloss	— mschlos@purdue.edu

What Went Well

We successfully completed most of Sprint 1. A user, no matter their stature, can log into the application. Primarily, this sprint focused on Admins. We successfully implemented logic to retrieve an Admin's data, as well as display to them the number of courses, students, and teachers in the school at that time. We also implemented the management of teachers, students, and courses. We also implemented the display of grades for an Admin. These grades, however, do not yet reflect live data in our servers.

With regards to the database and JSON translation, we were very successful in establishing a live connection between the user module and user information in the database providing a secure log-in for all User Types. Unit testing has been written for very important methods within UserQuery.

The implementation of the object management layer went well. Everything that was intended to be completed ended up being finished by the end of the sprint. The primary focuses of this sprint were to complete the User and Grade classes, including their constructors and public APIs (accessors, mutators, and other methods).

What Did Not Go Well

Communicating with the database and syncing data across the server and client was our issue in this Sprint. Setting up the database, while initially seemed trivial on paper, was much more difficult in practice. We focused on getting users and user types implemented so a user could log into the application. Due to the time resources this took up, we did not have enough time before the end of the sprint to fully implement Grades. The database table was formed, however the necessary underlying code to pull grade data was not implemented. This will be part of our next Sprint. Unit tests were also written to test methods within UserQuery. However, there have been some thread issues hindering us from executing the tests properly. We will fix these in the next Sprint.

Backlog ID: 6 (*continuation of Sprint 1)*
User Story: As an admin, I can get course-wide or school-wide grade stats.

Task Description	Expected Hours	Owner
Create GradeQuery JSON Translation class that allows for addition, modification and removal of grades	**20**	**Naveen Ganessin**
Create database tables that can store different courses, assignments and grades	**2**	**Gaurav Srivastava**
Create API that returns information about for every course	**10**	**Gaurav Srivastava**

ACCEPTANCE CRITERIA

- Given that the database and client information is built, I expect to see current grade stats from the server

How Should We Improve

We can utilize much more working testing tools. We can finalize UML because as of right now it is still fluid and changing. And to ensure consistency, we need a concrete database UML.

We can improve the readability of the code by dividing our code into separate class and methods. This can be achieved by doing more code reviews and locating pieces of code that can be used again for the user stories in sprint 2.