Plan

For this iteration we are going to implement some more user stories, but focus on getting the product ready for a final release by doing additional refactoring and testing.

The SQLDatabase class was properly implemented at the start of this iteration. The intention was for it to have been finished earlier, but a small bug in the onUpgrade method and a simple lack of understanding of the version property led to a number of run-time exceptions. Fortunately, with a working database development has sped up considerably as it can be relied upon to store data persistently and in a manner that is relatively simple to access.

Specifically, Ian, Anthony, and Graham are implementing the Grade Calculator, Calendar, and Reminders user stories. David is handling the final testing along with our customer.

The Grade Calculator user story will be implemented so that students using the application can keep track of their performance in a particular course, and potentially their degree program. This will be done by extending the functionality of the Database (Stub and SQL) to find tasks by the associated user and course attributes, modify the individual tasks, and by extending the Task class to store the grade and weight of an assignment or test. The Grader class was also created to handle the calculation of the grade and remaining weight for each Course, it has been tested with the GraderTest class.

If time permits we will also attempt to implement other user stories from our original vision statement. These user stories include:

* Passing Grade: As a student, I’d like to be able to view my grades for each course and calculate my overall GPA for the term and my degree
  + Priority: low
  + Cost: 1 week
* Organize Courses into Terms: As a student, I’d like to be able to organize my courses into terms so that I can easily find course information
  + Priority: low
  + Cost: 1 week
* Edit and View Prof Info: As a student, I’d like to be able to enter and view my prof and TA information so that I can easily find my professor’s office room, hours, etc.
  + Priority: medium
  + Cost: 1 week

The costs for these items have been revised as they would all require significant changes to be made to the application, and would require additional tests to be written. Other user stories we will most likely not be able to implement include: class attendance, change settings, degree requirements, update academic history.

One of, if not the most important things that we have to do with this iteration is test the application. We need to thoroughly test the important units, as well as test how well they work together. Many of these tests are already written; however, they need to be done in more detail. Specifically, we are going to try and focus on the edge cases where bugs can be left unnoticed for long periods of time. Furthermore, scaffolding needs to be added to the UIs themselves, so that user input will be less likely to result in a crash. For example, the weight of the tasks needs to be between 0 and 1, and the UI should not accept any other input for that category.

Additionally, we will do as much acceptance testing as we can. We will make use of the Robotium plugin for Android studio, but since we can only use it for free a small number of times we will need to be careful and not waste it. This means that we will need to push the acceptance testing back as much as possible, or test it ourselves without Robotium.