Task Breakdown for Stories in this Sprint:

User Story 3: As Apple, an instructor, I would like to create problem sets using the questions that I have created previously and set the number for attempts students are allowed for each problem set.

- **Task 24:** Use Problem/Problem Set object models for database API (High Priority)
- **Task 44:** Implement view problem set screen (3)
    - Create interface that shows list of all problem sets
    - User should be able to click on problem sets to show list of questions, reusing the view questions table from previous screens

User Story 4: As Apple, an instructor, I would like to be able to create new accounts for students that stores their name, student# and email.

- **Task 43:** Connect login command to database API (1)
    - Only allow valid student objects in the database to login
- **Task 46:** Link create new student account Screen to back end(2)
    - Implement the buttons in create new student account screen to use the appropriate methods to create a student account.
- **Task 47:** Link login in screen to back end(2)
    - Implement the buttons in login screen to check for user name and display the appropriate screen.

User Story 6: As Jennifer, a student, I want to complete problem sets that is released.

- **Task 42:** Create ProblemSetAttempt model (1)
    - Relationship between ProblemSet, Student, and a collection of answers from the student
    - Should contain method to get the assignment score
- **Task 45:** Implement answering questions screen (5, dependent on 42)
    - Use attempt model to keep track of student answers
    - User attempts one question at a time, navigating using next and previous buttons
- **Task 52:** Create table and functions to store and get the answers submitted by a student composed of four columns: student number, problem set key, the question number, and the answer. (2)
    - The result set returned for the select function should include two columns, the first being the problemID and the second being the student's answer.

Technical Debt:

- **Task 37:** Create API to store professor information into database (1)
- **Task 38:** Create API to extract Instructor information from database (1)
- **Task 39:** Update APIs to use the problem to problem set relation (2)
- **Task 41**: Create object model for Instructor user (1)
- **Task 48:** Update interpreter to use class name as command instead of number(2)
    - Update the hashtable in interpreter to use the command classes' name as the key instead of using number.

- **Task 49:** Complete JUnit tests for interpreter(4)
    - Create JUnit test cases for the interpreter.
- **Task 50:** Create JUnit Tests to test action, commands, databaseAPI, driver and the io using Mock Objects. (8?)
- **Task 51:** Complete JUnit tests for database classes. (4)