Sprint 4 :

Completion of the following task from Sprint 3:

Task 20: Implement GUI for CreateNewProblem(4)

- Create Controller and Manager to create new problem in instructor screen

Task 24: Use Problem/Problem Set object models for database API implementations(4)

- Overload ActOnDatabse on store api to receive problem and problem set object models
- Parse and pass info to database function that inserts to database
- Overload ActOnDatabse on Extract api to receive problem and problem set object model along with their unique identifiers
- Parse information provided by the database into attribute for the object and return them

GUI implementation:

Task 30: Refactor code in GUI(4)

- Refactor code in GUI to reduce repeated code

Task 31: Implement GUI for creating new student account for instructor(1)

- Create Controller and Manager for creating a new student account

Task 32: Implement GUI for student screen(1)

- Create Controller and Manager for the student scene
- Create Controller and Manager for the home screen for students
- Link login screen to check for student logins

Task 33: Implement login action (1)

- Implement code to compare login credentials with user objects obtained from the API

Task 34: Implement OutputGenerator to return information to the GUI (4)

- Add new implementation for OutputGenerator interface, strictly dealing with GUI
- Add a method to retrieve the latest output result
- Add a method to retrieve the current output generator from the Interpreter

Task 35: Implement GUI for viewing all available problems (3)

- Create Controller and Manager for view problems screen

Task 36: Implement GUI for choosing questions to add to problem set (4)

- Create Controller and Manager for Add Problem Set screen
- Use existing problem table view to select problems to add

User Story 5: As Apple, an instructor, I would like to set release dates and deadlines for the problem sets that I have created.

Task 26: Create table storing professor accounts (2)

- Create a table for professor accounts storing the professor's name, ID, email, and password.
- Create functions to insert and retrieve the professor’s information.

Task 27: Create relational table between professor and problems (1)

- Use foreign keys to create a table to store the relationship between the professors and the problems they create.

- Modify the insertProblem function to require a professor's ID.

Task 28: Create relational table between professor and problem sets (1)

- Use foreign keys to create a table to store the relationship between the professors and the problems sets they create.
- Modify the insertProblemSet function to require a professor's ID.

Task 29: Create relational table between problems and problem sets (2)

- Use foreign keys to create a table to store the relationship between the problem sets and the problems they contain.
- Modify the createProblemSet function to add an entry to this new table for each problem.

Task 37: Create API to store instructor information into database (1)

- Overload actOnDatabse on DatabseStoreAPI to take instructor object
- Parse its attribute and call respective function to store on databse

Task 38: Create API to extract instruct information from database (1)

- Overload actOnDatabse on DatabseExtractAPI to take instructor object and instructor id
- Retrieve result from DatabaseSelector and parse it
- Store the info into the supplied Instructor object and return it

Task 39: Update APIs to use the problem to problem set relation (2)

- Modify actOnDatabase for both APIs to store problem and problem sets into database to send an extra attribute to the database function
- Modify actOnDatabase for both APIs to extract problem and problem sets from database to receive an extra attribute from the database function

Task 40: Create the database entity relationship diagram for jworks.db(2)