Validation Plan:

User Story 1: Adding a question

Test Setup:
User logs in as an Instructor using any non-numerical username and password.
User clicks the “Create new question” button.

Test Scenarios:

Scenario	**Expected Result**
User presses Submit without entering any data.	User is prompted to fill in the empty fields. The question is not added.
User presses Submit after only entering the Question.	User is prompted to fill in the empty fields. The question is not added.
User presses Submit after only entering the Answer.	User is prompted to fill in the empty fields. The question is not added.
User presses Submit after entering both a Question and an Answer.	User is prompted that the question has been added. The form is reset.
User successfully submits a problem, and then returns to the screen to submit a new one.	The second problem is successfully submitted.

User Story 2: Viewing all questions

Test Setup:
User logs in as an instructor using any non-numerical username and password.
User clicks the “View Question” button.

Test Scenarios:

Scenario	**Expected Result**
There are no saved problems.	The table is empty.
There is a single saved problem.	The table shows the id, question, and answer for the single problem.
There are multiple saved problems.	All of the saved problems are shown, with their corresponding ID, question, and answer.

User Story 3: Create problem sets

Test Setup:
User has previously saved several problems.
User logs in as an instructor using any non-numerical username and password.
User clicks the Add Problem Set button.

Test Scenarios:

Scenario	**Expected Result**
User attempts to save a problem set with no fields filled in.	User is prompted to fill in all fields.
User attempts to save a problem set without any problems, but with all other fields filled in.	User is prompted to select at least 1 problem.
User double clicks on problems in the problem table.	The problem ID is moved to the current problem set table.
User double clicks on problems in the problem set table.	The problem is removed from the problem set table.
User attempts to create a non-empty problem set with a non-numerical entry for maximum attempts.	User is prompted to enter a valid numerical entry.
User creates a problem set with multiple questions and a valid positive integer of maximum attempts.	The problem set is saved. The form is reset.

User Story 4: Create student accounts

Test Setup:
User logs in as an instructor using any non-numerical username and password.
User clicks “Create New Student Account”.

Test Scenarios:

Scenario	**Expected Result**
User submits the empty form.	User is prompted to fill in empty fields.
User submits the form with a non-numerical student number.	User is prompted to enter a valid student number.

User submits the form with differing password and confirm password fields.	User is prompted that the passwords do not match.
User submits the form with a student number that already exists.	User is prompted that the student may already exist.
User submits the form with a unique student number.	User is prompted that the student account was successfully created.
Log out. Attempt to log in with the credentials of a valid student that was previously added.	User is logged in as a student.
Log out. Attempt to log in with a student number that doesn't exist.	User cannot log in.
Log out. Attempt to log in with a student number that exists, but with the wrong password.	User cannot log in.

User Story 5: Assigning release dates and deadlines to problem sets

Test Setup:
User logs in as an instructor using any non-numerical username and password.
User clicks the Add Problem Set button.
User adds multiple problems into the problem sets.

Test Scenarios:

Scenario	**Expected Result**
User saves a problem set without entering a release date and due date.	User is prompted to enter release date and due date.
User saves a problem set with a release date that is later than the due date.	User is prompted to enter a valid date range.
User saves a problem set with a release date that is equal to the due date.	User can successfully save the problem set.
User saves a problem set with a release date that is before the due date.	User can successfully save the problem set.

Verification Plan:

Unit Tests:
The unit tests will be focused around classes in Actions, Commands, Database, and Database API.

Integration Tests:
These tests are focused on verifying the pipeline specified in the original design: