Validation Plan:

User Story 1: Adding a question

Test Setup:
User logs in as an Instructor using the credentials "admin"/"admin".
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 the credentials "admin"/"admin".
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 the credentials "admin"/"admin".
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 creates a problem set with multiple questions.	The problem set is saved. The form is reset.

User Story 4: Create student accounts

Test Setup:
User logs in as an Instructor using the credentials "admin"/"admin".
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 the credentials "admin"/"admin".
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.

User Story 6: Completing problem sets

Test Setup:
User logs in as a student.
User clicks the View Problem Sets button.

Test Scenarios:

Scenario	Expected Result

There exists problem sets that are not yet released.	The user cannot see these problem sets until the release date.
User clicks on the Attempt Problems button without a problem set selected.	Nothing happens.
User clicks on the Attempt Problem button with a problem set selected.	User is navigated to a screen where the first question of the problem set is displayed. User can enter non-empty answers to go to the next question. User can submit the attempt after the last question to receive instant feedback. User is returned to the home page.

User Story 7: Search problems and problem sets by tags

Test Setup:
User logs in as an instructor using the credentials "admin" / "admin".
User clicks the View Questions button or the View Problem Sets button (repeat scenarios for both).

Test Scenarios:

Scenario	Expected Result
User types in part of a tag into the search field and presses search.	Only problems / problem sets that contain that partial tag are displayed.
User presses clear when the list of problems / problem sets is filtered.	The search field is cleared. All problems / problem sets are displayed.
User searches for multiple terms, separated by spaces.	Only problems / problem sets that contain all of the terms are displayed.
User searches for a term that matches no results.	No results are displayed in the table.

User Story 8: Bulk import students

Test Setup:

User logs in as an instructor using the credentials "admin" / "admin".
User clicks on the "Create new student account" button.

Test Scenarios:

Scenario	Expected Result
Click the Bulk Import button and press cancel on the dialog.	File choice dialog appears. Pressing cancel dismisses the dialog. No errors are shown.
Click the Bulk Import button and select a invalidly formatted file.	File choice dialog appears. An error is shown when error occur while adding accounts.
Click the Bulk Import button and select a validly formatted file.	File choice dialog appears. User is prompted that the addition of student accounts is successful. User can log in as any of the students added.

User Story 9: Bulk import problems

Test Setup:
User logs in as an instructor using the credentials "admin" / "admin".
User clicks on the "Create new question" button.

Test Scenarios:

Scenario	Expected Result
Click the Bulk Import button and press cancel on the dialog.	File choice dialog appears. Pressing cancel dismisses the dialog. No errors are shown.
Click the Bulk Import button and select a invalidly formatted file.	File choice dialog appears. An error is shown when error occur while adding problems.
Click the Bulk Import button and select a validly formatted file.	File choice dialog appears. User is prompted that the addition of problems is successful. User can go to the View Questions section of the application to view all questions.

User Story 10: View statistics for problem sets

Test Setup:
User logs in as an instructor with the credentials "admin" / "admin".
User selects the View Problem Sets option.

Test Scenarios:

Scenario	Expected Result
Click on the View Statistics button without a problem set selected.	Nothing happens.
Click the View Statistics button with a problem set without any attempts selected.	The statistics screen is shown. No records are shown in the table. The number of participants is shown as 0.
Click the View Statistics button with a problem set with multiple student attempts.	The statistics screen is shown. Each student that has at least 1 attempt is shown as one row. The aggregate statistics are correct.

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:

```
┌─────────────────────┐         ┌─────────────────────┐
│    Interpreter      │         │  OutputGenerator    │
│  Handles user input │         │    Shows output     │
└─────────────────────┘         └─────────────────────┘
         │                                 ▲
         │                                 │
         │ Calls using          ┌──────────┘ Sends output to stream
         │ string array         │
         │               ┌──────┴──────────────┐                      ┌─────────────────────┐
         │               │     Command         │  Passes formatted objects │      Action         │
         └──────────────▶│   Parses string     │─────────────────────▶│     Calls API       │
                         │ arguments to actions│◀─────────────────────│                     │
                         └─────────────────────┘  Returns desired output └─────────────────────┘
                                                                          │         ▲
                                                                          │         │
                                                     Calls appropriate method │    │ Returns object models
                                                                          │         │
                                                                          ▼         │
                                                              ┌─────────────────────┐
                                                              │        API          │
                                                              │  Get object models  │
                                                              │      from DB        │
                                                              └─────────────────────┘
                                                                     │         ▲
                                                                     │         │
                                              Calls DB connection │           │ Returns Results from DBQuer
                                                                     │         │
                                                                     ▼         │
                                                         ┌─────────────────────┐
                                                         │      Database       │
                                                         │  Create and execute │
                                                         │   queries on data   │
                                                         └─────────────────────┘
```