Title: Alma LMS
Members: Stanislaw Jaroszynski
Vision Statement: To create a more intuitive and functional Learning Management System.

User Acceptance Test Plans

Feature 1: Adding, Editing and Interacting with Posts
This feature will be tested by providing the user with a student account and a class in which they can interact with the class forum. The class will not have any initial posts; This will cause a "feature discovery" as shown below which prompts the user on how to add a new post and introduces them to the floating action button used universally throughout the application.

The user will then follow the prompts to create a post. Next the user will be instructed to attempt to edit their post. After that, the user will be instructed to comment on their post. The user will then be interviewed on the ease of the process and asked for their comments and concerns.

Feature 2: Account Creation

The user will be routed to the Alma home page and asked to register for their classes. A QA team member will be present to assist the user with any roadblocks they run into. The user will then be interviewed about the process. The goal of this test is to determine the ease with which the user can interact with the application and whether a user with no prior experience with the application can easily register for their classes.

Feature 3: Assignment Creation and Grading

The user will be provided with an existing course and professor account. The user will be asked to create a new assignment based off a provided simple paper copy. Next the user will be directed to another prepared assignment with existing student submissions. The user will be asked to attempt to review and grade their students' submissions. The user will then be interviewed and asked on their comments and concerns on the process.

Automated Unit Tests

Automated Unit Testing will be performed by the integrated testing system provided by Django. There are some automated tests for both the models and views. The model tests create a test Model (Python representation of a database row) and tests that its functions work as expected.

The view tests create a clean testing database and populate it with an example scenario. It then performs requests on a test http server and compares the responses against the expected outcome.

The tests can be run from the root of the git repository with `python3 ./manage.py test forum`, however for the tests to run, you need a python3 environment with Django installed.