5 Testing

5.1 Unit Testing

The project is broken into individual questions as a single unit. These questions are tested within Prairie Learn as they are compiled and previewed. The built-in grading system ensures that the questions are working as intended. Most of the tools used are built into the PL framework as files are implemented and the questions are previewed. The other way we are testing our units is by showing them to our advisor at each weekly meeting to make sure they look and function as he wants. He has also helped with making sure the material is correct as the homeworks we are implementing was created by him. A final testing step will be giving our system to a cybersecurity class to see if they can break into our server or any vulnerability in out project.

5.2 Interface Testing

The interface in our design is within the PrairieLearn environment. Within each question we have a json, html, and python file with work with each other. Then we make an assignment which combines multiple questions made. We test how they each communicate and make sure that the output works as desired. This is done within the PrairieLearn environment, giving us feedback on which issues are occurring within the questions. For the assignment portion, again it is within PrairieLearn, but we have not yet worked with this feature yet.

5.3 Integration Testing

The most important connection paths we have is making sure each individual question works on its own, and can be integrated with no issues into one big assignment, so that the students going through the homework can easily finish it. These will be tested by individually going through the Unit testing for each question, and ensuring proper formatting there. After that, we will go through the interface testing and connect the questions into an assignment. This in turn will be finishing the integration testing. The tools we will be using are what are available within the PrairieLearn environment we are using.

5.4 System Testing

The system can be tested by testing homework assignments as a whole and making sure they all work properly each time a question is attempted. This will ensure that when students go to attempt the homeworks that it all works properly and their learning time will be maximized. A lot of the questions within the homework are randomized so testing the homework assignments multiple times will be beneficial to make sure that random value in a question doesn't mess up the student.

5.5 Regression Testing

For regression testing, we will ensure that any new addition will be contained in its own container for each question, homework, and class. Containerization is a tool that was implemented in PrairieLearn previously and helps ensure that everything interacts with the necessary components. One critical feature we need in order to ensure that PrairieLearn will continue to run is to have

every component of the system scalable. Scalability is required for this project because PrairieLearn is a constantly growing system that needs to be adjusted for size and requests over time because it is a requirement driven by PrairieLearn requirements

5.6 Acceptance Testing

In order to demonstrate the design requirements were met, we will ensure that all pages made on the PrairieLearn website match the initial format given. Part of this is ensuring that all code is in a readable format and that each page has a consistent layout. To test this, we will manually go through each of our courses, assessments, and questions to ensure that they match the existing framework. To demonstrate that our functional requirements have been met, we will ensure that each question is auto-graded to an appropriate level for a subset of randomized outputs. Finally, in order to show that our non-functional requirements have been met, we will take a holistic view of the system and meet with our advisor to ensure that the user experience and system properties have been provided. We will involve our client in the acceptance testing by allowing them to use our site and give feedback on each of the three above kinds of requirements. Specifically, we will meet with the client on a regular basis after deliverables have been completed as well as at the termination of development.

5.7 Security Testing

A suggestion from our advisor was to give the production server to one of the cybersecurity classes for them to attempt to break in. We plan to use this to both give the cybersecurity students an interesting project and also test our security measures.

For integrity, the course information is retrieved using Git, so any changes are visible in the Git history.
To maintain confidentiality, we use encryption and logins are handled using OAuth2. This allows us to use the security measures of Google and Okta in our app to protect user information. We also use HTTPS to encrypt traffic between the client and server, making it very difficult to see what the user is doing.
Finally, for availability, we plan to run multiple servers behind a load balancer that can prevent our servers from being overloaded, and ensure downtime is minimal.

To protect the server, we use SSH with public key authentication and multi factor authentication when using a password, as well as a firewall that only accepts traffic from SSH, HTTP, and HTTPS. We also use NGINX to reverse proxy the PrairieLearn server and enable HTTPS, encrypting traffic between client and server. NGINX is also set up to redirect all HTTP traffic to HTTPS, ensuring encrypted traffic.

Finally, all user submitted code is compiled and run in separate docker containers that are isolated from the rest of the system. This means that any and all user input is sandboxed and will be unable to affect the rest of the system.

5.8 Results

For questions that have been developed so far, the Prairie Learn framework has been able to catch all mistakes in code as they are compiled. Our requirements are that the questions compile, display, and auto grade. All of these are implemented within the PL environment. For

non-software related testing, our advisor/client has helped catch any mistakes in formatting or material content and has guided us on finalizing questions as desired.