4 Testing

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, power system, or software.

The testing plan should connect the requirements and the design to the adopting test strategy and instruments. In this overarching introduction, given an overview of the testing strategy. Emphasize any unique challenges to testing for your system/design.

4.1 Unit Testing

What units are being tested? How? Tools?

Testing units will include each component in our app. These will include things like each pumpkin component on the volunteer side and each stencil entry on the admin side. Each page will also be tested as they are also components to our large application. Testing the image classification will also be important when it comes to classification accuracy. This part will be tested within TensorFlow with their testing libraries. The JS part of our application will be tested with Jest, ensuring that components are behaving as intended, and displaying the correct information.

4.2 Interface Testing

What are the interfaces in your design? Discuss how the composition of two or more units (interfaces) are being tested. Tools?

Jest will also be used in interface testing. We will be testing each API endpoint for their existence, as well as intended responses to input. Each API endpoint is hosted in Next.JS and will be simple to test each endpoint as they are created. These tests will be run at the start of each deployment in development and production to ensure that our app works as intended.

4.3 Integration Testing

What are the critical integration paths in your design? Justification for criticality may come from your requirements. How will they be tested? Tools?

In our design, there are two critical integration paths that require thorough testing:

- Firstly, it is essential to ensure that the data between the main components, such as the volunteer and admin sides, are consistent. This is critical because any inconsistency in data could lead to errors or incorrect information being displayed to the users, which could result in a negative user experience. Therefore, we will test the data consistency between components using integration testing tools such as Selenium, which can test the functionality and flow of the entire application.
- Secondly, the AI stencil recognition is another critical integration path that must be tested. We need to ensure that the stencil recognition works effectively with our current data. This is crucial as any inaccuracies in the stencil recognition could lead to incorrect pumpkin information, resulting in a negative user experience. We will use TensorFlow's testing libraries to test the accuracy of the AI stencil recognition.

The tool that will be used for integration testing is Selenium. Selenium is a popular open-source tool widely used for integration testing of web applications. It supports various programming languages, including Java, and can be used for testing applications developed in various web development frameworks, including Next.js. Additionally, Selenium supports cross-browser testing, making it a suitable choice for testing web applications across different browsers.

4.4 System Testing

Describe system level testing strategy. What set of unit tests, interface tests, and integration tests suffice for system level testing? This should be closely tied to the requirements. Tools?

The system level testing verifies the overall behavior and performance of a software application as a whole, and typically involves a mix of unit, interface, and integration tests before the additional tests that verify the functionality of the individual components together. Our system level testing strategy will be closely tied to the requirements and focus on validating the system's behavior against the specified mainly functional but also non-functional requirements. These will include the following requirements that will demand most if not all of the system to be operational:

1. Updating and maintenance of the stencils by the admins.
2. Preparation for the upcoming event which involves tracking of selecting, printing and cutting the stencils in the current year
3. Searching for and changing the pumpkin status from the admin's and the volunteer's side
4. Viewing the update submission logs and approving the statuses from the admin's side

The tools that will be used for system testing include React Testing Library, Jest, and Selenium. In order for these tests to be deemed successful, they will have to end with the correct result without taking too much time.

4.5 Regression Testing

How are you ensuring that any new additions do not break the old functionality? What implemented critical features do you need to ensure do not break? Is it driven by requirements? Tools?

Regardless of the changes made to the application in the current development cycle, it is crucial that the website stays accessible for the visitors and the admins both during the event and throughout the year. So, in order to ensure that the application meets the required specifications, is stable and reliable after each cycle, we will re-execute the test cases that have been executed in the previous cycles and add any additional cases that cover the new functionalities. The execution of the regression tests will be automated through CI/CD pipeline on Gitlab. A runner image will be generated to process the tests on our team's virtual machine on Iowa State Network.

As part of the Agile development process, these tests will mainly prioritize the satisfaction of the functional requirements that are most important for the application to run. These functionalities include the ability for volunteers to update pumpkin/stencil information, the admin's control over selecting, printing and cutting capabilities of the stencils, as well as approving or denying the volunteer's status submission.

4.6 Acceptance Testing

How will you demonstrate that the design requirements, both functional and non-functional are being met? How would you involve your client in the acceptance testing?

Our acceptance testing is being done on a bi-weekly basis with our client with additional tests if needed. Client acceptance is very important in this deliverable and thus a large decision making tool in our process. Each week leading up to the first week of October 2023 will consist of the tests with the client. Beginning this week, we plan to be putting our app into production and deep testing the process and workflow start to finish. A trial run of the event will occur. Our team will be available on site to solve any problems that arise. Based on feedback, and issues that occur, our team will work around the clock to ensure that our app can perform as expected in the event for years to come.

4.7 Security Testing (if applicable)

N/A

4.8 Results

What are the results of your testing? How do they ensure compliance with the requirements? Include figures and tables to explain your testing process better. A summary narrative concluding that your design is as intended is useful.

At this point, we do not have any test results to report.