

5 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.

5.1 UNIT TESTING

What units are being tested? How? Tools?

- Frontend
 - o Components
 - o Forms
 - o Buttons
 - o Functions
 - o Methods
- Backend
 - o API endpoints
 - Job Queue endpoints
 - User function endpoints
 - o Functions
 - Algorithm functions
 - o Methods

For the frontend we plan to use jest and React's built in testing library to test all of our essential react components, primarily to ensure that form and field validation is working correctly. For the backend user and visualization APIs, we will need to use a third-party library called Pytest to unit test our backend.

5.2 INTERFACE TESTING

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

The main interface in our design is our React UI which consists of multiple components rendered together. React fortunately has built in test utilities that work great for interface testing. Our backend also has the API interface that allows the frontend to communicate with the backend. Again, we should be able to use Pytest and data dependency injection to ensure that data is being processed and returned in the correct format.

5.3 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?

There are many integration paths that will need to be tested in our application such as user authentication, data retrieval and data transformation, and state management. All of these examples require “round trip” data paths that flow through the entire system. A great tool to test these paths is Cypress as we can ensure that all components in the system are interacting correctly with one another.

5.4 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?

System testing will be carried out by modeling pre-calculated models to check that the system works. The system testing will use unit tests from the algorithm and visualization API and ensure that the transfer from the backend to frontend integration is working properly. It will model basic user interactions using UI tests. The entire process will be executed to ensure that the software meets the basic functional requirements provided by the Client. This will all be executed by the CI/CD pipeline process on GitLab and will report data on failures and successes.

5.5 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?

Regression testing will be performed throughout the CI/CD pipeline where builds will be compiled, tested, and eventually published to ensure that all parts are working together. We will be able to see points of failure in the system when all the parts are integrated together as a part of the CI/CD pipeline. More specifically, we will be using the GitLab CI/CD pipeline to carry this out.

5.6 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?

Acceptable testing will be performed by both unit testing and client confirmation. The unit tests will test the algorithms and parameters to ensure that it produces the expected output. The clients will analyze and review the output of the program to ensure compliance with their functional requirements and accuracy.

5.7 SECURITY TESTING (IF APPLICABLE)

In our design we are going to be prioritizing testing the functionality of the system as we have a limited time to implement and test. As such, if time permits, we will ensure that no major security concerns exist in the system. We could also use source code analysis tools to scan for any known vulnerabilities in our system. Mainly we want to ensure that our login system does not let unauthorized users into the system. One in the system we also want to ensure that the current user can only access data that belongs to them/their organization.

5.8 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.

As with any software product testing can help with various aspects of concern such as:

- Identifying and fixing bugs
- Verifying functionality
- Validating requirements
- Ensuring components can communicate with each other

Testing is a vital aspect to our project plan which is why we have budgeted time in our project plan to implement the described tests. Actual results will be provided once the system is implemented, and tests are written.