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 adopted test strategy and instruments. In this overarching introduction, give an overview of the testing strategy and your team's overall testing philosophy. Emphasize any unique challenges to testing for your system/design.

The testing plan connects the requirements and the design to the adopting test strategy for the project. The goal is to have a majority of the application covered with a variety of different testing methods to ensure the application is functional and does not break under normal conditions. With the microservice design of the project, this not only allows for scalability but also enables individual components to be tested.

In the sections below, describe specific methods for testing. You may include additional types of testing, if applicable to your design. If a particular type of testing is not applicable to your project, you must justify why you are not including it.

5.1 Unit Testing

What units are being tested? How? Tools?

We will be testing all of the endpoints in all of our components, as well as integration testing all the components with the API to see if the system components communicate well with each other, we will use tools like PYtest and jest.

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?

There are a lot of components in our project that will require a great amount of testing in regards to the interfaces. First, the majority of our project runs through an API that takes data from the power grid and integrates it into our environments. We will have to test that interface to make sure that the data is being ingested correctly. Next, once the data is into our environments, we will need to test the interface between the databases and our Google Cloud Platform environment, which is where most of the computation will happen. GCP controls the hosting of all aspects of our project, so it will be important to make sure that it interfaces correctly with the databases.

Our methods of interface testing will be workflow and performance tests. We will be testing the workflow to make sure that it is able to ingest live data and support our real-time analytics that will be part of the functionality of the site. Second, we will have to test the performance of our infrastructure to make sure that it can efficiently integrate high volumes of data.

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?

In the system, all services communicate through a centralized API. This is very useful as it allows us to manipulate and monitor the flow of data and interactions between services. We will leverage this to perform integration testing on our system. There will be a number of action flows and control paths to test that involve two or more systems.

Examples include user login/authentication, uploading and managing data, and defining filters and alerts among other macro-actions consistent with user behavior. We will use Cypress to perform integration testing.

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?

Using a blend of unit testing, integration testing, and regression testing. We will make sure that each of our system components is functional and communicate with other components successfully. The system will be in a microservice configuration, so we will test that each microservice functions correctly and that they communicate successfully. Because we will be combining unit, integration, and regression testing, we will combine PyTest, Cypress, and CI/CD pipelines to complete system testing.

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?

Having a CI/CD pipeline to make sure system integrity after every push. This will make sure the critical features in all parts of the systems do not break when adding or developing new features. We also will be making sure the database does not break, This will be done by utilizing GitLab CI/CD.

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?

We will make a report of the overall system performance from a both functional and non-functional perspective and check if all the design requirements are met or not.
We will execute several real-life scenarios from the perspective of an end-user. We will evaluate the software's performance and how well they comply with the requirements and also how well with works from the userend.

5.7 Security Testing (if applicable)

Security testing will be very important for our project. We will be handling a lot of sensitive data from the power grid that should not be publicly accessible by anyone outside of local governments and utility companies. To comprehensively assess the security of our platform, we will need to thoroughly test the API to make sure that data access is strictly limited to only known users, and to make sure that there is no way to intercept our data as it’s being transmitted from the power grid, through the API, to our databases. Additionally, we will have to perform vulnerability scans on our user interfaces to make sure that anything accessible on the site is protected and that there are no vulnerabilities on the front-end that can allow people to access databases without proper rights.

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.

Below is a table outlining the type of test, the process used for the test, and the expected outcome. These tests will help ensure compliance with the project requirements as they each test different aspects of these project requirements. Therefore, we will be aiming to pass as many tests as possible and clearly communicate to the client where our project has not passed our desired test

Testing	Process (How to ensure complaince)	results
Unit	jest	Prove that internal functions and systems are running
interface	postman	Test API endpoints
integration	cypress	Ensure that multiple components of the systems work as expected when combined to produce the desired result.
System	Cypress	Have complete end-to-end testing done on the complete software to ensure each system works as expected.
regression	CI/CD	Ensure existing features/functionality remains as new functionalities are added.
Acceptance	Manual QA	Ensure the software meets the requirements of the clients or users. Through constant communication with the client, this is easily accomplished.
Security	Manual QA	Attempt to break a software's security checks to gain access to confidential data. This is crucial for our project because many consumers rely on energy.