# 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?

Create tests using python that will test the different functions within our code. We will create test cases for all of the main major keras functions that will be called. We will also test units within our UI such as the API calls.

## 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?

We will test the interface between the user interface and the AI model to confirm that images are successfully sent and the UI receives the result from the AI model. Given that our project is also using a website we will also test the website to make sure that it is processing and sending the information correctly.

## 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?

Some critical integration paths include the path between the UI interface and our dataset. The UI will need to be able to accurately and efficiently and pull the data from our trained model/dataset. We will test this by testing the UI and making sure that it accurately predicts cancer images. If the interface accurately tests the images it can be concluded that the UI is accurately pulling from the model/cloud. Our model will be stored on the cloud. We have not settled on a tool to use but we are considering using applications such as Selenium and Puppeteer to test the UI.

## 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?

We will use challenges and test images to confirm whether or not the A.I. is behaving accordingly. Plus we will check whether the images from the website are being transmitted properly to the server. Example of system tests include but not limited to:

- Check if A.I. is identifying cancer images 75% of the time.
- Check if A.I. is not identifying cancer images as cancer images.
- Check if A.I. is identifying a group of cancer images and not others.
- Check if A.I. is identifying images with peculiar characteristics instead of cancer images.
- Check if A.I. can identify cancer images outside of the training images.

NOTE: While the tests may sound redundant, the tests are there to make sure that we understand the reasons the AI is giving the results it does. For example, the AI could identify cancer images more than 75% of the time, but these cancer images being detected is because they have the ruler in the image and not because it is cancer by itself. In other words, we want to make sure the AI is detecting cancer because it is cancer and not because of something else in the dataset.

Interface testing can also include but not limited to:

- Check if the interface is in line with the actions it's made to do
- Check if the interface is usable to anyone at first glance
- Check if UI is able to communicate effectively what each thing should be doing

Integration testing could be but not limited to:

- Check if the combination of functions and modules work in conjunction with each other
- Check if the system is reliable, functional, and performs well
- Check if each module of parts of the systems functionality is working correctly

### 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?

In order to ensure that new additions do not break our old functionality, we will make new branches in github for each addition. This will make sure that if we do break old functionality, we still have the previous code that we can look back on. Once we have ensured that the additional code does not break the old functionality, we can merge the branch into the main one. We need to ensure that the overall functionality of the AI does not break when we are adding new implementations.

### 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 demonstrate that the design requirements, both functional and non-functional, are met by thorough testing. We will have videos that walk through our design and also show the functionality and that it is functional. With our client, we will have them test out the functionality of our application to ensure that it is working properly and that it has all of the implementations needed.

### 5.7 Security Testing (if applicable)

We will test that the website is only accessible by an authorized user. We will use Selenium IDE to automate UI tests related to login and registering.

## 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.

- The results of the testing once we get to it is that everything is working as expected.