1. Develop a test plan for the new Annalise Web Demo. Be sure to include:

a. The test methods and tools to be used
b. The specific test steps to be executed and acceptance criteria to be evaluated

TEST PLAN

The Annalise Web Demo should be tested using functional and nonfunctional methods. We can perform both black-box and white-box testing techniques to ensure that the site has the highest quality.

Functional testing

Unit testing : Each component should have its own set of unit tests to ensure that it works as expected.

Integration testing : Integration tests should be added whenever possible to ensure that components work together as expected.

More tests should be covered in the above two layers according to the shift-left strategy and test pyramid.

API Testing: The API testing can be performed manually first. To accomplish this, we can use tools such as Postman. The main GET and POST request scenarios should be automated using Cypress or similar tools.

Manual testing:

High level test scenarios

- Upload images

  Positive scenarios

  - Check and ensure that the image should be of JPG, PNG or DICOM format with a minimum resolution of 1024 x 1024.
  - Check and ensure that accepted DICOM transfer syntax and modalities are supported.
  - Check and ensure that the upload process is working fine.
  - Verify the data in the back end database to make sure that the uploaded data is inserted properly.
  - Verify using both the drag and drop and the Select image method.
  - Ensure that the upload is successful if only one required image is given.
  - Ensure that the upload is successful if only one required image and one optional image are given.
  - Ensure that the upload is successful if one required image and two optional images are given.
  - Check whether the 'Terms and Conditions' checkbox is clickable.
  - Ideally, the 'Upload study' button should be enabled when the 'Terms and Conditions' checkbox is clicked.
  - The images should be uploaded and processed within 30 seconds.
  - The 10-study limit per week should be verified.
  - Upload multiple studies and ensure that the studies left field is showing the correct value.
  - The three pre-loaded cases can be selected, and they function as expected.
  - The cookies opt in/opt out functionality is working as expected.
  - Check whether the keyboard shortcuts and tabs work as expected on the page.
  - Ensure that upload file size limits are properly verified.

Negative scenarios

- Check whether the appropriate error message appears when the required image is not selected and the upload button is pressed.
- Clear the browser cookies and check if we can upload more than 10 studies.
- Check if we can upload an image without clicking on the 'Terms and Conditions' checkbox.
- Check if a very large file will crash the site.
- Check whether the site will show appropriate error messages when an unsupported image type is uploaded.
- Check if the system gracefully handles the situation where the front image is not uploaded.

- View Finding form page

Positive scenarios

- On the View Clinical Findings page, check whether the details can be filled in properly and the 'View AI Findings" button can be clicked.
- If you choose 'Send info of Annalise product & events' the newsletter is getting sent properly.
- The unsubscribe option is working as expected.
- The fields on the page should have proper validation checks.
- The fields can be filled properly using keyboard shortcuts, and the tabs are in proper order.

Negative scenarios

- Validation and blank values in fields should be checked.
- Check if the page can be broken by inputting long strings, and ensure proper checks are in place.
- Check to see if you subscribed first, then unsubscribed and resubscribed; if so, the newsletter will be received successfully.

- Viewing AI Findings

  Positive scenarios

  - Ensure that AI findings are displayed on the right panel.
  - Verify whether the Chest X-ray is displayed in the main view area.
  - Check whether, when you hover over the finding name in the right panel, the confidence level displays beneath the chest X-ray.
  - Check if there is an eye icon displayed next to the region of interest.
  - Verify that there are multiple eye icons for multiple regions of interest.
  - Verify that the 'priority items' are displayed at the top on the right hand side.
  - Verify if the number displayed next to priority is accurate.
  - Verify that the 'other items' are displayed below the priority items.
  - Verify if the number displayed next to Other is accurate.
  - Verify that the 'Not detected' are displayed below the Other items
  - Verify if the number displayed next to 'Not detected' is accurate.
  - Clicking on the icon next to ‘Priority’ will show child elements.
  - Clicking on the icon next to ‘Other’ will show child elements.
  - Clicking on the icon next to ‘Not detected’ will show child elements.
  - Ensure ‘L’ icon for findings with a Left laterality.
  - Ensure ‘R’ icon for findings with a Right laterality.
  - Ensure ‘L+R’ icon for findings with Bilateral laterality.
  - Ensure that findings with no localisation are indicated with a left arrow icon.
  - Ensure that the confidence score is displayed in the UI according to its value in the database.
  - Ensure that the X-ray zoom functionality is working as expected.
  - Ensure that the X-ray pan functionality is working as expected.
  - Ensure that the X-ray reset functionality is working as expected.
  -

- Others

  - Ensure that HELP functionality works as expected
  - Ensure that SHARE functionality works as expected.

Automation testing:

We can do both API and UI automation testing. Both can be done using Cypress or Selenium based tools. Only the critical path of the UI needs to be automated, as the UI tests can be fragile compared to the API ones and will also take more time to run.

Non Functional testing

Cross browser testing: We can use tools like BrowserStack or our own automation framework to run those tests. These tests should ensure that the site works well with Google Chrome, Microsoft Edge, Mozilla Firefox, and Safari.

Load Testing : We can perform load testing to determine the system's performance at peak and normal times. The results can be used to identify areas of concern, and we can use tools like jmeter or blazemeter to test them.

Performance testing : We can determine how fast the system performs with a particular load. We can use tools like JMeter or BlazeMeter to perform this.

Security testing: We need to perform security testing in order to determine the vulnerability of the system. The site currently uses a query parameter in the URL, which is not secure. We need to check whether SQL injection or any other security lapse is present in the system using Burp Suite or Sync.

A/B testing : We should consider doing A/B testing with a slightly different version of the site to determine the user's preferences.

Usability testing : We should consider doing usability testing to determine the ease of use of the system by real users.

2. Please describe how you could test the web service independently and what tools/code you would use to test this

We can test Annalise web services independently and verify that the GET and POST requests are working fine. We need to have an authorisation header in order to perform this in a secure way. The most reliable tool that can be used is Postman for manual testing, and for automation, we can use Cypress as the front end of the application is using Javascript. For performance testing, we can use Apache JMeter.

The other tools that are worth exploring are GraphQL, Katalon, Apigee, SOAP UI, REST-assured, and Karate.

For the REST web service, it is a best practice to use JSON for sending and receiving data. We should set *Content-type* in the response header to *application/json*. Use the *https* protocol to make it more secure.

The endpoint for the API request can be **https://api.annalise.ai/study**

For the POST request, the following payload can be used:

```
{
 "AccessionNumber": <uuid>,
 "Description": "Chest X-ray of Patient1",
 "PatientAge": 50,
 "patientId": 100,
" studyInstanceUid": <Uid>,
"ImageContent": <encoded base64 image>
}
```

If successful it will return **201 Created**

For the GET request the end point can be **https://api.annalise.ai/study/patient/<id>**

It should return the image when we perform a GET request.

3. Can your propose a sql query to verify the results in the DB based on the table structure below

When we enter the relevant accession number and study instance id, the SQL query below will retrieve the study image from the AWS S3 server. It is retrieved by using join with the primary and foreign key combination. In this way, we can independently verify within the database whether we are getting the right result. It also checks whether the status of the record is in completed stage.

```
SELECT patient.image.s3_url , patient.study_images.study_id from patient.study
INNER JOIN patient.study_images
ON patient.study.id = patient.study_images.study_id
INNER JOIN patient.image
ON  patient.study_images.study_id = patient.image.id
WHERE patient.study.study_status = “Complete”
AND patient.study.accessionNumber = ?
AND patient.study.studyinstanceUid = ?
```