Testing Documentation

Thicc Peas

z5238611 Hao Cheong

z5258064 Brandon Green

z5191527 Fanrui Li

z5257129 Ryan Vu Anh Nguyen

z5263663 Max Owen

Table of Contents

1.0 Testing Processes Used

1.1 Environment and / or tools used

Our testing framework is written **Node.js**, with the following dependencies / packages:

Dependencies / Package	Justification
Mocha JS	Used for test framework
Chai JS	Used to make assertions
SuperTest	Used for making API calls
loadtest	Used in load and performance testing

1.2 Limitations

As of submission of this document, our API only has GET requests, therefore no authentication. Because of this, we cannot test the security of this API, as there is none to test.

As of this report, we have yet to fully configure CI/CD with GitHub actions, and so the tests must be run manually.

1.3 Testing Process

Our API will be tested with a multitude of Mocha API tests. They will perform queries with different query parameters and ensure that the correct status codes and responses are generated. The tests test:
- Stress and volume test
- Correct responses for valid queries
- Correct responses for invalid queries
- Correct status codes
- Edge cases

The tests are run manually using the `mocha` command inside the 'PHASE_1/TestScripts'.

Our API will be tested with mock JSON responses from the database. This ensures that regardless of the state of the database the API should be tested with the same input.

2.0 Overview of Tests

2.1 Overview of test cases, test data and results

/diseases

Test Case	Results
200 response and has salmonella	Tests that the /diseases endpoint has

/disease/{diseaseid}

Test Case	Results
200 response and gets salmonella from id	Tests that the /disease endpoint has Salmonella, given by disease id '*777d1c109f5eef1d64c418062a918d33*'

2.2 Output of testing and actions taken