PandemicTracker

Test Plan

Sangjin Lee

**Objective**

The PandemicTracker will aggregate information from trusted sources (WHO, CDC, Johns Hopkins etc.) about the current COVID-19. The data will not be limited to just the number of cases but will include other information, such as travel bans and country preparedness. The system will present all this information in an accessible manner to end users, allowing them to browse global as well as national data. Users will be able to rate the information that is presented to them in terms of usefulness.

**Scope of Testing**

|  |  |
| --- | --- |
| Functional  requirements | 1. Software should display comparison for countries 2. Software should display information 3. Software should display pandemic statistics 4. Software should be able to retrieve data |
| Non-functional  requirements | 1. Reliability of the rendered information 2. Capacity of the database to store information 3. The interface should be responsive for the users |

Test cases will be carried out so that some of the specific requirements are met in a reliable measure. In order to validate the requirements above, we will be carrying out **Alpha/Beta testing**. The purpose of performing Alpha testing is to simulate a real user experience through a preliminary testing internally by the team members. This stage of testing ensures that the customer’s requirements are appropriately met with regards to functionality, performance, and durability. The purpose of Beta testing is to gather inputs from a limited number of real users about the requirements. This ensures that the software is reliable because it allows for final changes by the developers.

**Use cases**

The following use cases were selected for testing because they are the key use cases to PandemicTracker.

1. Display comparison for countries
   1. The use case will display the comparison for countries in criteria regarding number of cases, deaths, recoveries, doctors and hospital beds per 1000 beds using graphs and tables for the two chosen countries.
2. Display information
   1. The use case will display the aggregate information that delivers country preparedness and travel bans.
3. Display pandemic statistics
   1. The use case will display scraped data and travel bans for every country in the country list.
4. Retrieve data
   1. The use case will allow the software to retrieve data regularly using the internal timer through web scraping and HTTP requests.

|  |  |
| --- | --- |
| Test Case 1 | |
| Module | Display comparison for countries |
| Title | Compare countries template test |
| Purpose | This test will be a black box test carried out by a member of the public acting as a customer, and a white box text carried out by a member of the developer team. |
| Pass Condition | Pass if the page loads valid and correct data, fail if the page fails to meet quality standards. |

|  |  |
| --- | --- |
| Test Case 2 | |
| Module | Display statistics |
| Title | Country list template test |
| Purpose | This test will be carried out black-box by a customer (member of the public) and white-box by a developer. The test checks that the country list webpage loads correctly and shows the right information. |
| Pass Condition | Pass if the page loads properly without errors, all countries show in the list, the list sorts itself by column, each eligible country has an associated flag. |

|  |  |
| --- | --- |
| Test Case 3 | |
| Module | Display information |
| Title | Homepage template test |
| Purpose | This test will be carried out black-box by a customer (member of the public) and white-box by a developer. The test checks that the homepage of the Pandemic Tracker website loads correctly with the data and the flag links |
| Pass Condition | Pass if the page loads properly without errors, all the country flags of the world are shown, clicking on a flag brings you to that country’s data page. |

|  |  |
| --- | --- |
| Test Case 4 | |
| Module | Update database |
| Title | Query correctness |
| Purpose | This tests for the correctness of input for the Database queries |
| Pass Condition | Pass if queries are written with correct data types and if the updates are not done unnecessarily. |

|  |  |
| --- | --- |
| Test Case 5 | |
| Module | Update database |
| Title | Query correctness |
| Purpose | This tests for the correctness of input for the Database queries |
| Pass Condition | Pass if country names from scrapers match and there are no duplicate entries. |

|  |  |
| --- | --- |
| Test Case 6 | |
| Module | Retrieve data |
| Title | General Webscraping Validation Test – Expected Flow |
| Purpose | This test case will test for requirement compliance and proper operation for the four Scrapers in the program. |
| Pass Condition | Pass if the database can call all of the we scrapers, which then execute without error and return their respective dataframes for database insertion in the proper format. |

|  |  |
| --- | --- |
| Test Case 7 | |
| Module | Retrieve data |
| Title | General Webscraping Validation Test – Expected Flow |
| Purpose | This test case will test for requirement compliance and proper operation for the four Scrapers in the program. |
| Pass Condition | Pass if the scrapers handle execution errors (related to loss of connectivity or wrong data formatting) properly. |

|  |  |
| --- | --- |
| Test Case 8 | |
| Module | Retrieve data |
| Title | StatsScraper Validation Test – Expected Flow |
| Purpose | This test case will test for requirement compliance and proper operation for the StatsScraper which gets country-level case data from Johns Hopkins. |
| Pass Condition | Pass if the scraper does its function correctly by connecting to the most recent Johns Hopkins dataset, scrapes confirmed cases, deaths, and recoveries for every country and stores them in a dataframe, and aggregates these three statistics to get the global numbers, and returns all data. |

|  |  |
| --- | --- |
| Test Case 9 | |
| Module | Retrieve data |
| Title | StatsScraper Validation Test – Expected Flow |
| Purpose | This test case will test for requirement compliance and proper operation for the StatsScraper which gets country-level case data from Johns Hopkins. |
| Pass Condition | The objective of this test is to ensure that the scraper handles various errors well. These errors include trying to scrape updated data without Johns Hopkins updating their dataset first and the parameters of the dataset changing. |

|  |  |
| --- | --- |
| Test Case 10 | |
| Module | Retrieve data |
| Title | TravelScraper Validation Test – Expected Flow |
| Purpose | This test case will test for requirement compliance and proper operation for the TravelScraper which gets country-level travel restrictions from Wikipedia. |
| Pass Condition | Pass if the scraper does its function by connecting to the Wikipedia collection of restrictions, scrapes each country and its restriction, removes links, references, and other non-text data from each entry, and returns a dataframe with all the data |

|  |  |
| --- | --- |
| Test Case 11 | |
| Module | Retrieve data |
| Title | TravelScraper Validation Test – Expected Flow |
| Purpose | This test case will test for requirement compliance and proper operation in various edge cases for the TravelScraper which gets country-level travel restrictions from Wikipedia. |
| Pass Condition | Pass if the scraper can smartly figure out the parameters of the scraping and update them accordingly when encountering changed data. |

|  |  |
| --- | --- |
| Test Case 12 | |
| Module | Retrieve data |
| Title | HealthCareScraper Validation Test – Expected Flow |
| Purpose | This test case will test for requirement compliance and proper operation for the HealthcareScraper which gets the number of beds and doctors per 1000 people for each country from data aggregator IndexMundi. |
| Pass Condition | Pass if the scraper connects to two separate IndexMundi databases and scrapes the number of beds and doctors correctly. |

|  |  |
| --- | --- |
| Test Case 13 | |
| Module | Retrieve data |
| Title | HealthCareScraper Validation Test – Expected Flow |
| Purpose | This test case will test for requirement compliance and proper operation for the HealthcareScraper which gets the number of beds and doctors per 1000 people for each country from data aggregator IndexMundi. |
| Pass Condition | Pass if the scraper can smartly figure out the parameters of the scraping and update them accordingly when encountering changed data. |

**Test Schedule & Resources**

The key use cases outlined above are most crucial and mandatory for PandemicTracker to run efficiently. Therefore these use cases will be developed and tested. Once the development of the project is finalized, I will move on to the validation process that includes the Alpha/Beta testing stage where the user experience will be simulated to test whether it is reliable and does not produce any bugs.

The testing stage will approximately require 2~3 days. More time will be allocated to the Beta testing than the Alpha testing because Alpha testing will first be done with the team members (4 people), whereas for the Beta testing, a number of real users (10 users) will be asked to test out the functionality and performance. The resources will include a series of web pages to scrape from, a database set up using MAMP and phpMyAdmin to be able to store data, CDC and WHO resources for healthcare guidelines. The web pages will be selected according to the availability, maintainability, security, and reliability of the information they provide. The database will be set up using an SQL query and will be filled in with scraped data using the scrapers. The health guidelines will be posted on the guideline page using Twitter API.

**Participants**

In conducting both Alpha and Beta testing for the purpose of validating customer requirements, the project would first require us as a team to conduct Alpha testing on the software. We will carry out a number of tests each to raise the accuracy in troubleshooting and identifying errors. For Beta testing, we would require a few real users to give feedback on the finalized product. For this product, more than 10 internet users will be asked to experience the software themselves. The Beta testing will not only render realistic feedback but also tailor the project in a more reliable and an intuitive fashion.