Testing

When developing any form of software, even if everything appears to be working, there will be uncaught errors and issues in the program. It is therefore essential to ensure that testing is undertaken to discover any of these errors.

For this project two forms of testing were used to evaluate the program, and decide whether it meets its goals. To test the system’s functionality a series of tests, which correlated to specific user stories, were run manually. User experience was tested with a short survey, where users were given tasks to follow and provide their answers, as well as providing general feedback about the user experience, successfully testing both the system itself and showcasing the views of potential users.

System Testing

The following table outlines all the primary technical tests run to analyse the product. For each test, two methods were used to test functionality (where applicable).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Description | Stories Tested | Postman | Website |
| 1 | Run forecast to display energy production statistics | 1 | Pass | Fail |
| 2 | Run simulation to see energy production statistics | 2 | Pass | Fail |
| 3 | Input a forecast with hourly set, and compare to one not set hourly | 1,3 | Pass | Fail |
| 4 | Run two simulations with all same settings other than timescale and compare results | 2,4 | Pass | Fail |
| 5 | Run simulations with different solar/offshore/onshore settings | 2,5 | Pass | Fail |
| 6 | Update/delete a previously run forecast and simulation and check stored values | 8 | Pass | Pass |
| 7 | Check lists of forecasts and simulations on webpage against database | 7 | Pass | Pass |
| 8 | Save a forecast or simulation to database | 9 | Pass | Pass |
| 9 | Rerun a stored simulation and check data is valid | 2,7,10 | Pass | Fail |
| 10 | Search for specific known plants from database in online display | 11 | Pass | Pass |
| 11 | Add a new theoretical plant to a specific region | 12 | Pass | Pass |
| 12 | Add a new user to system | 15 | Pass | Fail |
| 13 | Try to login with a created user | 15 | Fail | Fail |
| 14 | Call API independently of a forecast | 14 | Pass | Pass |
| 15 | Check that you can’t add user with existing email or username | 15 | Pass | Fail |

The two methods of testing shown in the table were

* Test the program using the website to check for full functionality and meeting of user story requirements and checking for bugs
* Testing by sending appropriate HTTP requests using Postman to check that if there is an issue whether it is relating to how the front end is receiving and handling the data or the back-end logic. Some of these tests can be seen here: . Link also in appendix.

User Testing

A simple questionnaire was designed to allow users to provide feedback on the product. Several of the questions in the questionnaire pertain to simple tasks set for the users, as well as several questions giving the users to provide general views on the usability, aesthetic and general feel of the program. The general questions were, where possible, kept to neutral language as to prevent any unintentional bias towards a specific view of the program. Where users were allowed to input their own choice of values, they were instructed to mention these choices in their response section. The reason for allowing users to input their own choices rather than giving a set of inputs in the task description was to ensure that varied testing was undertaken to check for any bugs and prevent only testing the system in a way that didn’t only test parts of the system known to be fully functional. In addition to this, allowing the user to input their own values more accurately matches how a non-technical user such as a student may approach using the application. “Stories Tested” is only shown here for clarity, this was not shown to the users.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Task | Response | Stories Tested |
| 1 | Find the forecast page, run and save a new forecast with your choice of settings. Answer with Forecast ID |  | 1,3,5,9 |
| 2 | Try to add a new onshore windfarm in the location of your choice. Or update an existing one. Give plant ID |  | 11,12 |
| 3 | Navigate to the info page, read the documentation and run an API call, did you receive a 24hr forecast? |  | 14 |
| 4 | From the simulation page access the list of previously run simulations, then rerun a chosen simulation.  Write down your chosen simulation and its first value |  | 2,8,10 |
| 5 | Register an account for yourself, were you successful |  | 15 |
| 6 | Rerun a forecast of your choosing and answer as for Q4 |  |  |
| 7 | Run a new simulation using your choice of settings |  | 1,2,4,5,6 |
| 8 | Navigate to all the pages, were you successful? |  |  |
| 9 | Delete a forecast of your choice, were you able to delete it? |  | 8 |
| 10 | Update a forecast, what happened? |  |  |
|  | General Questions | Response |  |
| 11 | How did you find the navigation around the website? |  |  |
| 12 | What do you think of website colour scheme? |  |  |
| 13 | What opinions do you have on the page layout if any? |  |  |
| 14 | Did you understand what the information you were looking at represented? |  |  |
| 15 | What features would you like to see in a future version of the program? |  |  |
| 16 | Were you able to complete most of the tasks from the instructions? |  |  |

\*Full results of user questionnaires can be found in the appendices.

Analysis of Testing

System Testing

The system testing shows that nearly all of the main functionalities work on the backend, however several fail tests when using the web application, this highlights the difficulties that were encountered during the build.

The only system test to fail in every aspect was test 13, this was an expected failure as only the very basic structure of an authentication system was attempted to be implemented in the initial design period.

User Testing