Test Strategy Document

Akshat Goyal, 2018101075
Demand forecasting of energy consumption of air conditioning in commercial complex
Team number - 12

Scope

->This document will be reviewed by Mr.Malliswar who is our client form the company INDRIYN DATA ANALYTICS PRIVATE LIMITED.
->He will take into account the basic functionality of the webapp and see if all the predictions regarding energy consumptions made are close to actual values.

Test Approach

Process of testing:

- Setting up the test for a particular piece of an application (called the system under test)
- Performing the actual testing (interacting with the system under test)
- Observing the resulting behavior and checking whether expectations were met or not.

Testing Levels:

- So far we have done unit testing and integration testing for backend and frontend parts.
- Data is cleaned manually.
- We have checked all APIs using tools like mocha, postman.

Test Environment

- We have done unit testing and integration testing for backend and frontend parts using jupyter notebook and UI was tested on local host having node, react and express installed.
- The data for the training model was given in csv file data was loaded to train the model using python script.

- A running internet service is also required to fetch weather and temperature data.
- Code is uploaded on GitLab for efficient version control.
- We have created a copy of the initial raw data provided by the client.

Testing Tools

- Automation and Test management tools needed for test execution
- Figure out the number of open-source as well as commercial tools required, and determine how many users are supported on it and plan accordingly testing Tool
- Mocha is used to automatically test rest APIs in node.js
- Postman is used for testing APIs
- Energy Prediction values are tested through test set accuracy on our model using model.score() function..

Use Cases

- Write the use cases derived from the feature analysis
- Essentially there should be an understandable link between the use cases and the features which are related to it.
- Login and Register
- Select date and time range for energy consumption.
- Prediction of energy consumption.
- Graphical analysis of energy consumption.
- Download graphs.
- View Weather Prediction.
- Budget Prediction.
- Anomaly Detection.
- Notification for anomaly.
- View current Energy
- Consumption.

Test Cases

- Write the test cases derived from the use cases.
- The written test cases should be arranged by the use cases.

- Testing that no two user with same username can register and user can access any webpage only after logging in and is able to access only authorized page.
- Testing the data type and format of date and time.
- Testing the accuracy of model's prediction.
- Testing if the current budget is getting updated.
- Testing the anomalies using graphical analysis.
- Testing graph: able to download correct graph and is clearly visible.
- Testing the open weather api using postman.
- Testing if correct anomalies are covered using graphical analysis.
- Testing that user is getting notifications by generating anomaly manually.
- Testing the APIs used for fetching current energy consumption using postman, mocha.