**Software Requirements Specification (SRS) Document**

|  |  |
| --- | --- |
|  | Demand Forecasting of Air Conditioning for Commercial Complex, Team 12 and Akshat Goyal, Kanish Anand, Nikunj Nawal and Sridhar M |

# **Brief problem statement**

Given the energy consumption data of any building we have to estimate the future energy demand of the complex on an hourly and daily basis. We are supposed to create a web application for the pre-stated problem statement.

# **System requirements** Frontend - HTML, CSS , JavaScript Backend - NodeJS Database - MongoDB Machine Learning - Python3 , Mathematics and ML libraries Cloud Hosting - Azure or AWS API - Open Weather API

# **Users profile**

# [Users will be the community where the system is deployed]

|  |  |  |
| --- | --- | --- |
| S.No. | USER | Description |
| 1 | Management | Monitoring of energy consumption rate and act accordingly |
| 2 | Employee/Resident |  |
| 3 | for keeping track |  |
|  |  |  |

# **Feature requirements (described using use cases)**

***Read the instructions below and fill in the table. Delete all the blue text turning it in.***

(This is a numbered list of use cases that are the features of the system to be implemented. Each use case is an operation that the user can perform on/with the system. For each use case, provide a description (2-3 sentences) so you know what to build and so you can write a test case to demonstrate that your system provides that feature. Typically, your project will have 10-15 use cases, but feel free to add or delete table rows if you decide to use finer-grain or coarse-grain use cases).

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Use Case Name** | **Description** | **Release** |
| 1. | Cleaning of data provided and converting them to predefined format | Here cleaning data majorly includes checking and filling missing data. Taking(or converting) data points in minute wise format. | R1 |
| 2. | Manually correcting anomalies in data set | making an option for user to edit data set in case of vague readings |  |
|  | Real Time Prediction of hourly or daily energy consumption with Notebook without User Interface. | Users can do the prediction of hourly and daily energy consumption by using Jupyter Notebook. Functional Web-app not included | R1 |
|  |  |  |  |
|  | Real Time Prediction of hourly or daily energy consumption with proper User Interface. | Users can use our Web application to predict the amount of energy going to be consumed in a particular hour or a day in the futureD. | R2 |
|  | Users can add the current data of energy consumption. | Users can add data for energy consumption which will be helpful for our model for more accurate future predictions as it will increase the size of our dataset. | R2 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Use case diagram**

***Read the instructions below and fill in the table. Delete all the blue text before adding this to your repository or turning it in to your instructor.***

Draw the UML use case diagram for the system. Make sure the use cases shown in the diagram correspond to the use cases described in the previous section.

**Use case description**

***Delete all the blue text and fill-in the template before adding this to your repository or turning it in to your instructor.***

|  |  |
| --- | --- |
| **Use Case Number:** | UC-XX (Replace XX with a number) |
| **Use Case Name:** | Enter the name of Use Case |
| **Overview:** | Describe the purpose of the Use Case and give a 1-2 line description. This could be the same as the description provided in feature requirements section. |
| **Actors:** | List all actors that participate in this Use Case. |
| **Pre condition:** | Enter the condition that must be true before the main flow is executed. |
| **Flow:** | Main (success) Flow: Steps should be numbered. |
|  | Alternate Flows: Include the post condition for each alternate flow if different from the main flow. |
| **Post Condition:** | Enter the condition that must be true when the main flow is completed. |