# Project Report

## Introduction:

The generator log and reporting system uses a GUI to allow for entry of generator run data, viewing of entered data, editing of incorrect data (with proper user rights), and generation of the final annual EPA report. The current system does not provide for data control and relies on manual calculation and entry of data for the annual report. This introduces errors and takes excessive man hours. There is no effective way to monitor system run times against permitted run hours which puts the company at risk of violating EPA regulations incurring fines and possible revocation of operating permits.

Users request the system functionality to control access and rights for data. The system must allow the user to view data based on data parameters and automatically generate the annual report. The system should also allow users to enter generator run data and users with specific access should be able to edit incorrect data. The system should validate data entry to ensure dates and times are the correct format and run times are only positive. The system should also generate alerts as run times approach annual limits.

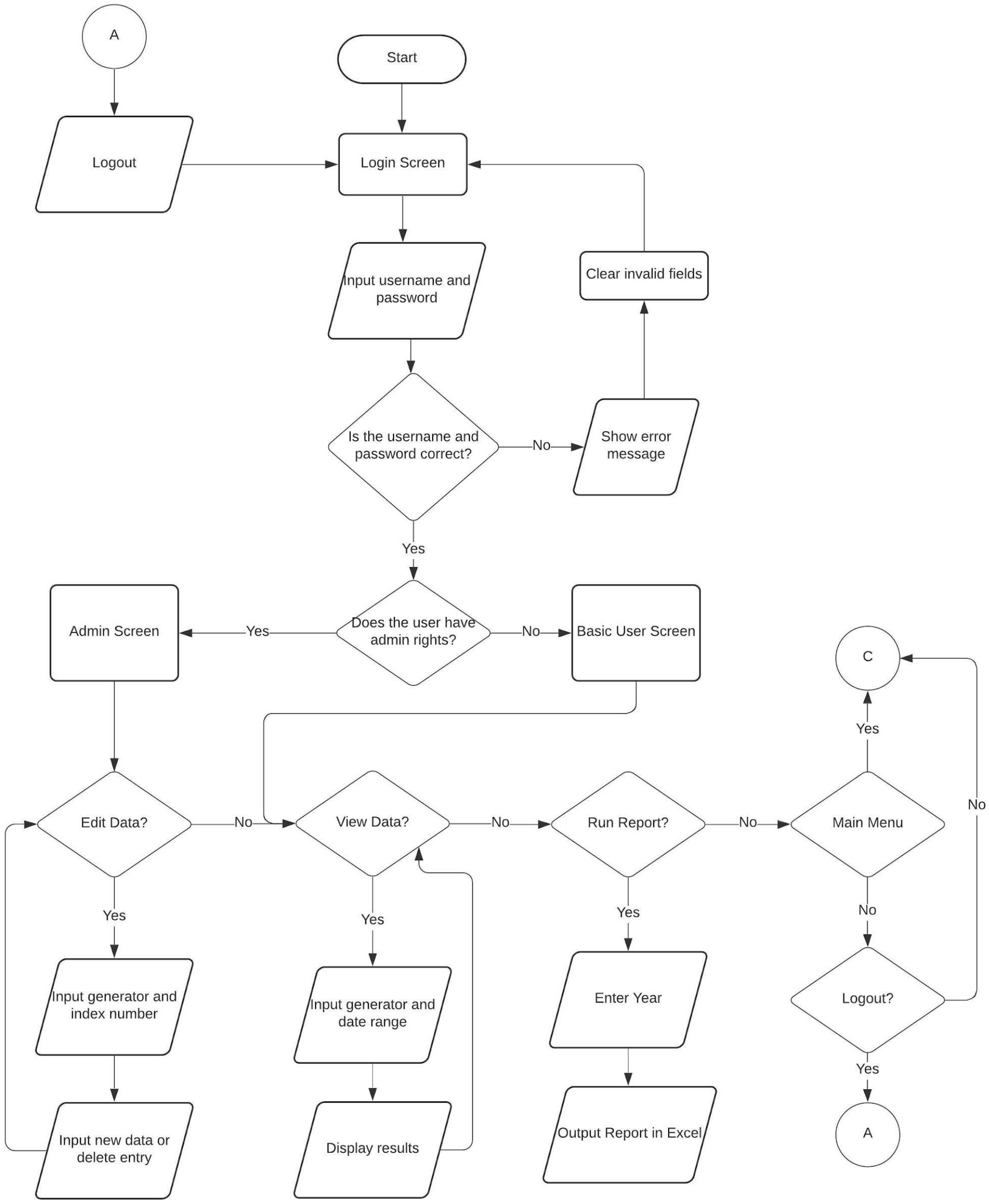
The system will streamline the process of data entry and report generation. It will also improve data quality, provide data validation upon entry, and alert users if operating activities are putting them at risk of permit violations.

## Application Flow:

User logs into the system using assigned credentials. There will be either two or three workflows to choose from, depending on user rights. The system provides error messages if the credentials are entered incorrectly. The two user levels that are defined are admin and non-admin. The admin user will be able to see three workflows, whereas the non-admin user will see two. The admin will see ‘Enter Generator Data’, ‘View/Edit Generator Data’, and ‘Run Annual Report’. The non-admin user will see ‘View Generator Data’ and ‘Run Annual Report’. By restricting the user permissions, we are able to provide additional data protection by limiting the number of users with access to make data modifications. All screens have a return to main menu and logout function to allow users to back out of their current location in the program hierarchy.

List of functions:

- login
- view_data
- enter_data
- edit_data
- delete_data
- create_report
- mkexcel
- cumrunhour
- calcFue



Interface:

(See User Manual)

Data File Design:

- File data structure currently consists of excel files.

Credentials Data File (credentials.xlsx)

- Single sheet in workbook with sheet name of ‘creds’
- This sheet is used by the p[program to validate login data

| Field Name | Data Type | Data Length | Description |
|---|---|---|---|
| username | String | 64 | Name of user |
| password | String | 64 | User password |
| authlvl | String | 5 | Access rights granted to the user |

Generator Data File (Gen_Data.xlsx)

- Each generator has its own sheet in the workbook defined by the sheet name ‘GEN ‘ + Generator Name as a string. (i.e. GEN 1A)
- Workbook created and used by program for data entry and recall for various functions.

| Field Name | Data Type | Data Length | Description |
|---|---|---|---|
| StartDate | Date | 10 | Day that generator run started |
| StartTime | Time | 5 | Time that generator run started |
| StopDate | Data | 10 | Day that generator run ended |
| StopTime | Time | 5 | Time that generator run ended |
| RunHourStart | Float | 6 | Unit run hour counter at start |
| RunHourStop | Float | 6 | Unit run hour counter at start |
| Load | Int | 4 | Average load level during run |
| ReasonForRun | String | 128 | User entered reason for run |
| RunTime | Time | 5 | Calculated run time |
| CalcFuel | Float | 8 | Calculated fuel usage for run |

Report Summary File (Summary__YYYY.xlsx)

- Workbook consists of 2 sheets. Sheet names are ‘SummaryYYYY’ and ‘Annual_Limit’
- This sheet is an output only. It is created by the program but is not utilized by the program in any other way.

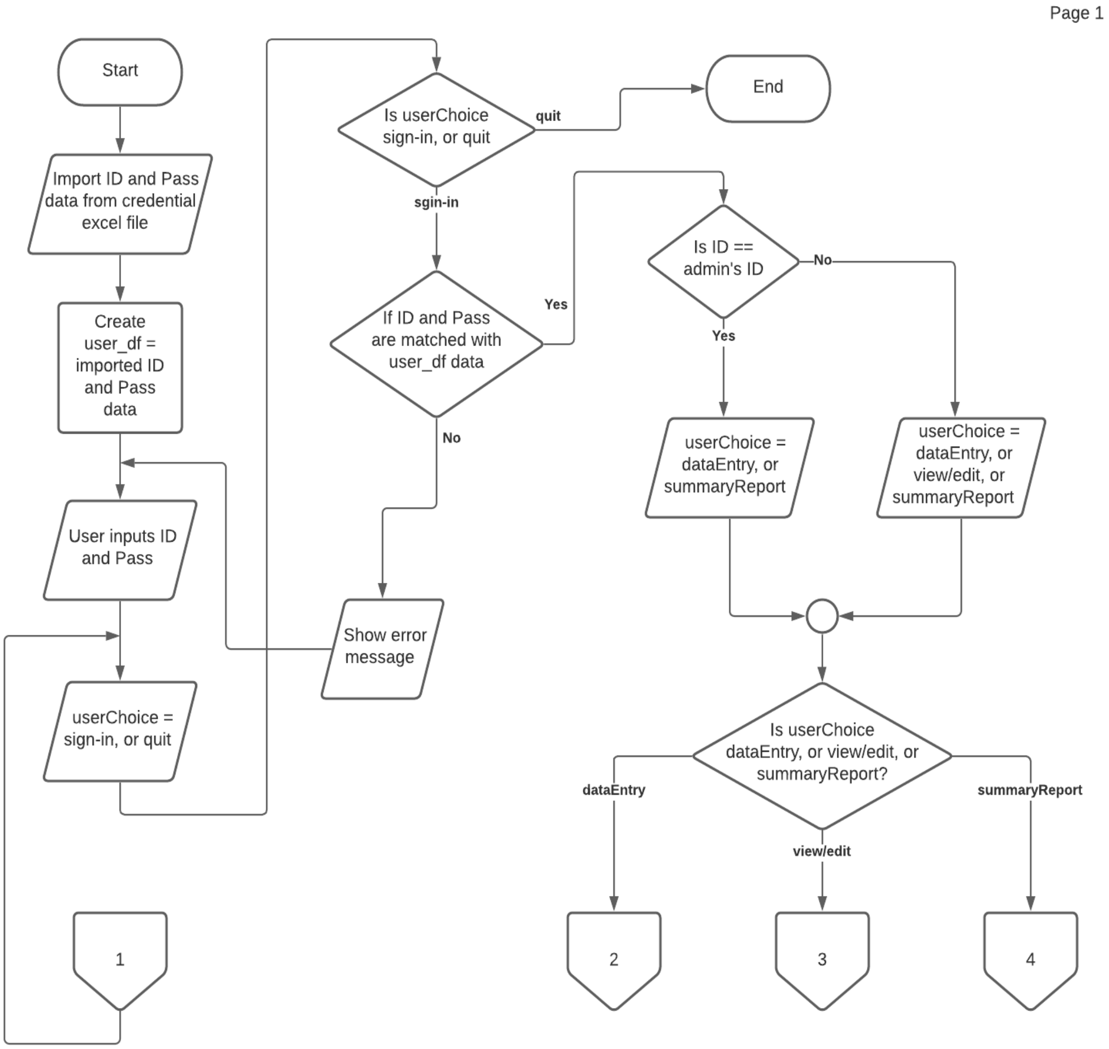
SummaryYYYY Sheet

| Field Name | Data Type | Data Length | Description |
|---|---|---|---|
| Months<br>(Jan-Dec) | Time | 6 | Calculated run time for each generator for the given month. |
| Total Run Hours | Time | 5 | Calculated annual run time for each generator |
| Total Fuel Usage | Int | 5 | Calculated annual fuel usage for each generator |
| Total NOx | Float | 6 | Calculated NOx Emissions for each generator |
| Total SO2 | Float | 6 | Calculated SO2 Emissions for each generator |
| Total CO | Float | 6 | Calculated CO Emissions for each generator |
| Total PM 10 | Float | 6 | Calculated Particulate Emissions for each generator |
| Total VOC | Float | 6 | Calculated Visual Emissions for each generator |

Annual_Limit

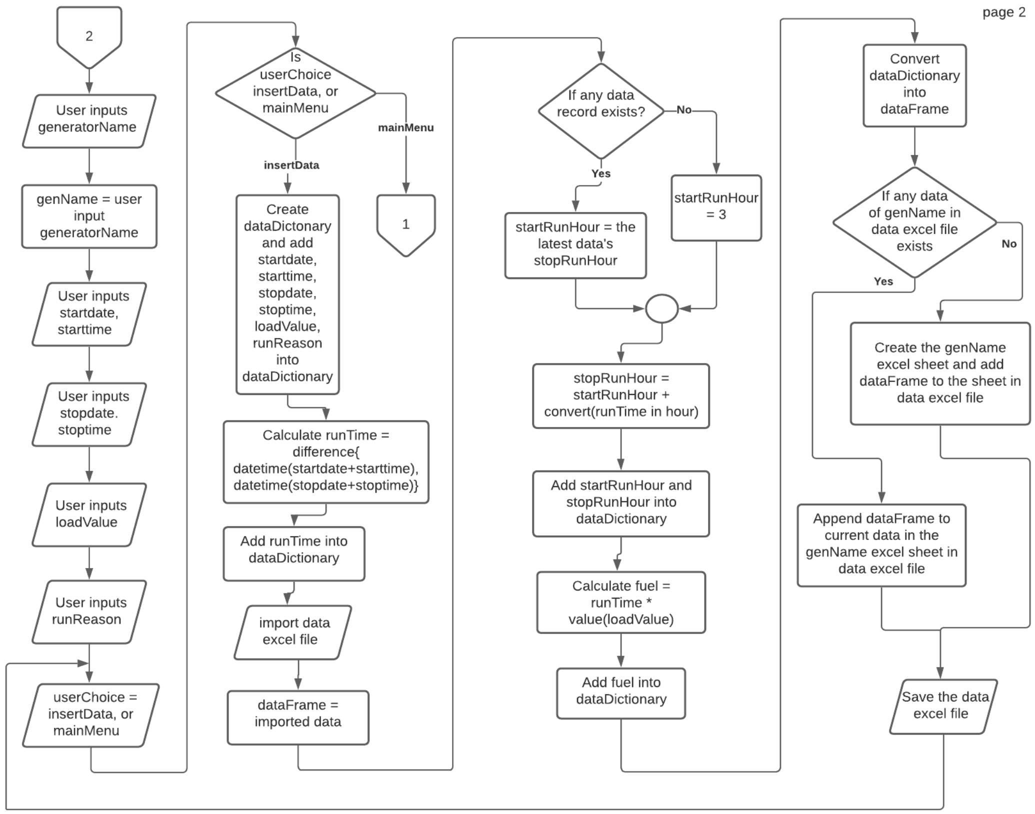
| Field Name | Data Type | Data Length | Description |
|---|---|---|---|
| Emissions limit by AIRS point | Float | 6 | Permit limits for emissions. |
| Total Run Hours For each AIRS point | Time | 6 | Run time associated with the generator with the highest total run time for the year to be compared to AIRS limit |
| Emissions Test | String | 64 | Displays if emissions are near limit |
| Run Time Test | String | 64 | Displays if run time of any generator is near limit |

Detailed Design:


Page 1
Start
Import ID and Pass data from credential excel file
Create user_df = imported ID and Pass data
User inputs ID and Pass
userChoice = sign-in, or quit
1
Is userChoice sign-in, or quit
quit
End
sgin-in
If ID and Pass are matched with user_df data
Yes
No
Show error message
Is ID == admin's ID
No
Yes
userChoice = dataEntry, or summaryReport
userChoice = dataEntry, or view/edit, or summaryReport
Is userChoice dataEntry, or view/edit, or summaryReport?
dataEntry
view/edit
summaryReport
2
3
4

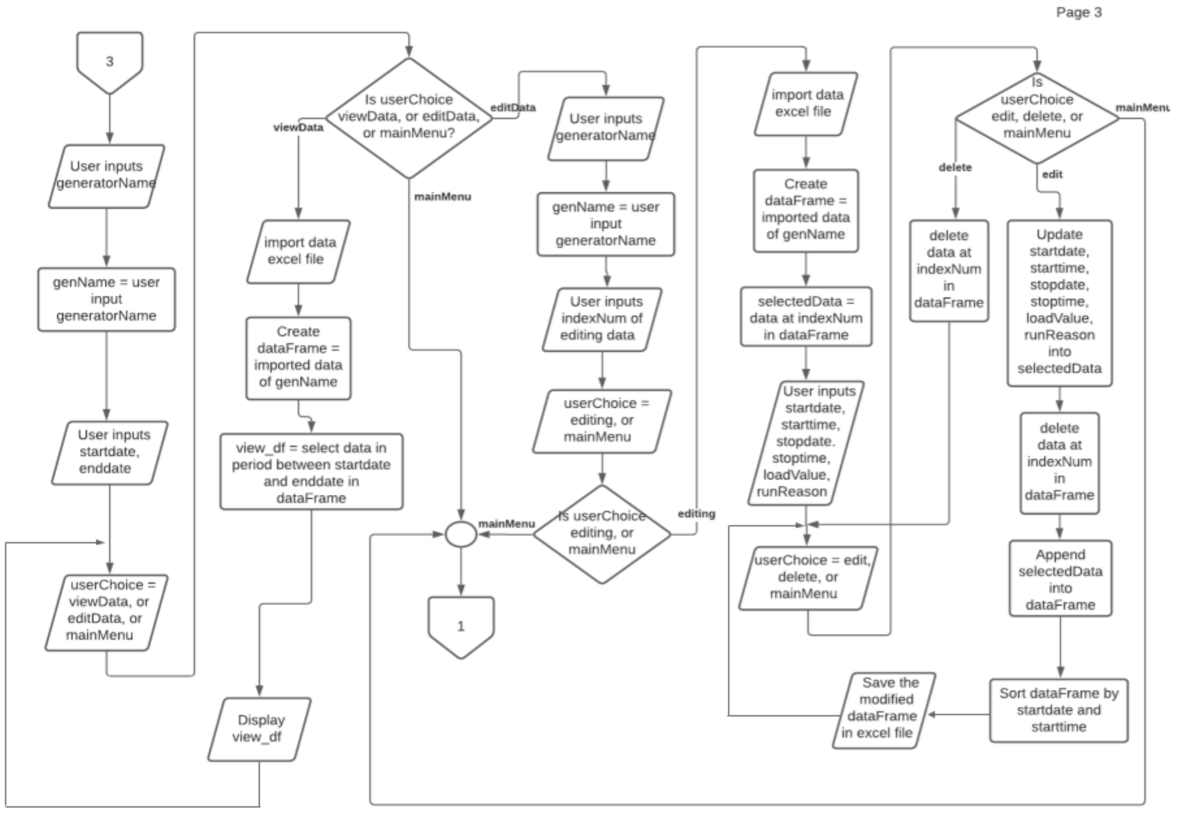
- Import data from credential excel file which includes user ID and Password into python, and create a data frame using Pandas library.
- Get user inputs for user ID and Password.
- Get user input for the next action which is sign-in or quit.
- If the user input for the next action is quit, the program is ended.
- If the user input for the next action is sign-in, the program checks if the user's pair of ID and Password is in the data frame of ID and Password.
- If it is not in the data frame, the program displays an error message and goes back to the section for entering ID and Password.
- If it is in the data frame, the program checks if the user's ID is the same as the admin ID in the data frame.

- If yes, get user input for the next action which is dataEntry, view/edit, or summaryReport. If not, get user input for the next action which is dataEntry, or summaryReport.


page 2
2
User inputs generatorName
genName = user input generatorName
User inputs startdate, starttime
User inputs stopdate. stoptime
User inputs loadValue
User inputs runReason
userChoice = insertData, or mainMenu
Is userChoice insertData, or mainMenu
mainMenu
insertData
1
Create dataDictonary and add startdate, starttime, stopdate, stoptime, loadValue, runReason into dataDictionary
Calculate runTime = difference{ datetime(startdate+starttime), datetime(stopdate+stoptime)}
Add runTime into dataDictionary
import data excel file
dataFrame = imported data
If any data record exists?
No
Yes
startRunHour = the latest data's stopRunHour
startRunHour = 3
stopRunHour = startRunHour + convert(runTime in hour)
Add startRunHour and stopRunHour into dataDictionary
Calculate fuel = runTime * value(loadValue)
Add fuel into dataDictionary
Convert dataDictionary into dataFrame
If any data of genName in data excel file exists
Yes
No
Create the genName excel sheet and add dataFrame to the sheet in data excel file
Append dataFrame to current data in the genName excel sheet in data excel file
Save the data excel file

- If the user chooses the dataEntry option, the program comes to page 2 of the flowchart.
- Get user input for the generator name, and set the name as a variable genName.
- Get user inputs for the information to create new data. The information is such as the generator's start date, start time, stop time, load value, and the reason for running the generator.
- Get user input for the next action which is insertData or mainMenu.
- If the user input for the next action is mainMenu, the program goes back to page 1 of the flowchart, and gets the user input for the next action which is sign-in or quit.
- If the user input for the next action is insertData, the program creates a dictionary which has the keys, startdate, starttime, stopdate, stoptime, loadValue, runReason.
- Calculate the generators runTime by the difference between the stop date time and the start date time. Then add the runTime value in the key name runTime in the dictionary.
- Import data from the excel file regarding generator operating data into python, and create a data frame using Pandas library.
- The program checks if there is any data written or not in the data frame.
- If yes, set the startRunHour value of the latest data. If not, set the startRunHour value of three. Then add the startRunHour value in the key name startRunHour in the dictionary.

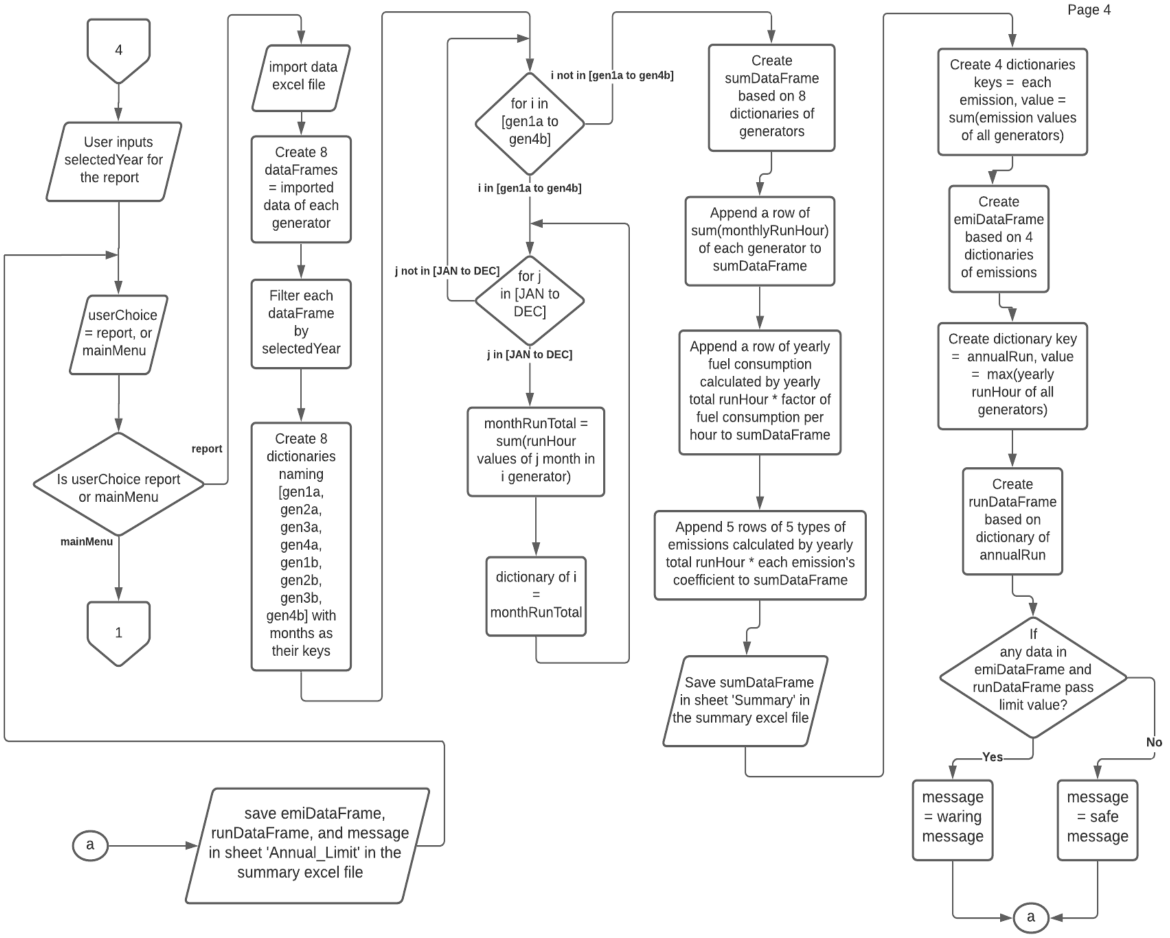
- Calculate the stopRunHour by startRunHour plus runTime. Then add the stopRunHour value in the key name stopRunHour in the dictionary.
- Calculate the fuel value by runTime multiply by the factor value depending on loadValue. Then add the fuel value in the key name stopRunHour in the dictionary.
- Convert the dictionary into a data frame.
- Check the database excel file if there is any sheet about genName or not.
- If yes, append the created data frame to the current data frame in the genName sheet. If not, create a new sheet naming genName and write the created data frame on it.
- Save the updated database excel file in the storage.


Page 3
3
User inputs generatorName
genName = user input generatorName
User inputs startdate, enddate
userChoice = viewData, or editData, or mainMenu
Is userChoice viewData, or editData, or mainMenu?
viewData
editData
mainMenu
import data excel file
Create dataFrame = imported data of genName
view_df = select data in period between startdate and enddate in dataFrame
Display view_df
User inputs generatorName
genName = user input generatorName
User inputs indexNum of editing data
userChoice = editing, or mainMenu
Is userChoice editing, or mainMenu
editing
mainMenu
1
import data excel file
Create dataFrame = imported data of genName
selectedData = data at indexNum in dataFrame
User inputs startdate, starttime, stopdate. stoptime, loadValue, runReason
userChoice = edit, delete, or mainMenu
Is userChoice edit, delete, or mainMenu
mainMenu
delete
edit
delete data at indexNum in dataFrame
Update startdate, starttime, stopdate, stoptime, loadValue, runReason into selectedData
delete data at indexNum in dataFrame
Append selectedData into dataFrame
Sort dataFrame by startdate and starttime
Save the modified dataFrame in excel file

- If the user chooses the view/edit option, the program comes to page 3 of the flowchart.
- Get user input for the generator name, and set the name as a variable genName.
- Get user input for the period of data, startdate and enddate.
- Get user input for the next action which is viewData, editData, or mainMenu.
- If the user input for the next action is mainMenu, the program goes back to page 1 of the flowchart, and gets the user input for the next action which is sign-in or quit.
- If the user input for the next action is viewData, import data from the excel file regarding generator operating data into python, and create a data frame about the genName generator using Pandas library. Then create a view data frame which is filtered by the

period between startdate and enddate in the imported data frame. Finally, the program displays the view data frame, and the program goes back to the section to get user input for the next action which is viewData, editData, or mainMenu.

- If the user input for the next action is editData, the program gets user input for the generator name, and sets the name as a genName variable.
- Get user input for the target data’s index number.
- Get user input for the next action which is editing, or mainMenu.
- If the user input for the next action is mainMenu, the program goes back to page 1 of the flowchart, and gets the user input for the next action which is sign-in or quit.
- If the user input for the next action is editing, import data from the excel file regarding generator operating data into python, and create a data frame about the genName generator using Pandas library.
- Get user inputs for the information to create new data. The information is such as the generator's start date, start time, stop time, load value, and the reason for running the generator.
- Get user input for the next action which is edit, delete, or mainMenu.
- If the user input for the next action is mainMenu, the program goes back to page 1 of the flowchart, and gets the user input for the next action which is sign-in or quit.
- If the user input for the next action is delete, the program deletes the data at the target data’s index. And then go back to the section to get user input for the next action which is edit, delete, or mainMenu.
- If the user input for the next action is edit, the program stores the generator's start date, start time, stop time, load value, and the reason for running the generator into selectedData dictionary.
- Delete the data at the target data’s index.
- Append selectedData to the imported data frame, and then sort the data frame by start date and start time.
- Save the modified data frame into the database excel file. Then then go back to the section to get user input for the next action which is edit, delete, or mainMenu.


Page 4
4
User inputs selectedYear for the report
userChoice = report, or mainMenu
Is userChoice report or mainMenu
report
mainMenu
1
import data excel file
Create 8 dataFrames = imported data of each generator
Filter each dataFrame by selectedYear
Create 8 dictionaries naming [gen1a, gen2a, gen3a, gen4a, gen1b, gen2b, gen3b, gen4b] with months as their keys
for i in [gen1a to gen4b]
i not in [gen1a to gen4b]
i in [gen1a to gen4b]
j not in [JAN to DEC]
for j in [JAN to DEC]
j in [JAN to DEC]
monthRunTotal = sum(runHour values of j month in i generator)
dictionary of i = monthRunTotal
Create sumDataFrame based on 8 dictionaries of generators
Append a row of sum(monthlyRunHour) of each generator to sumDataFrame
Append a row of yearly fuel consumption calculated by yearly total runHour * factor of fuel consumption per hour to sumDataFrame
Append 5 rows of 5 types of emissions calculated by yearly total runHour * each emission's coefficient to sumDataFrame
Save sumDataFrame in sheet 'Summary' in the summary excel file
Create 4 dictionaries keys = each emission, value = sum(emission values of all generators)
Create emiDataFrame based on 4 dictionaries of emissions
Create dictionary key = annualRun, value = max(yearly runHour of all generators)
Create runDataFrame based on dictionary of annualRun
If any data in emiDataFrame and runDataFrame pass limit value?
Yes
No
message = waring message
message = safe message
a
a
save emiDataFrame, runDataFrame, and message in sheet 'Annual_Limit' in the summary excel file

- If the user chooses the summaryReport option, the program comes to page 4 of the flowchart.
- Get user input for selectedYear for the report.
- Get user input for the next action which is report, or mainMenu.
- If the user input for the next action is mainMenu, the program goes back to page 1 of the flowchart, and gets the user input for the next action which is sign-in or quit.
- If the user input for the next action is report, import data from the excel file regarding generator operating data into python, and create eight(8) of data frames of each generator using Pandas library.
- Filter each data frame by selectedYear, so each data frame has only the data in the year of selectedYear.
- Create eight(8) of dictionaries of each generator which have months as keys.
- Using nested 'for loop', calculate the sum of the monthly run-hour of each generator, and then store it into the corresponding generator dictionary.

- Using all the generator dictionaries, make a data frame whose columns and indexes are the name of generators and month names respectively.
- Add a row to the data frame that is the sum of run-hour of each generator for a year.
- Add a row to the data frame that is the total fuel consumption based on the sum of yearly run-hour of each generator.
- Add 5 rows to the data frame that are the total emissions such as NOx, SO2, CO, PM10, VOC which are calculated by the total run-hour multiplied by each emission's coefficient
- Save the data frame in the sheet 'Summary' in the summary excel file.
- Create four(4) of dictionaries. Each of the dictionaries has 4 types of emission as keys and sum of all generators' emission values as a value. Then create an emission data frame based on the four(4) of dictionaries.
- Create a dictionary that has annualRun as a key and the highest value among the yearly run-hour values of all generators as a value. Then create a run-hour data frame based on the run-hour dictionaries.
- Check if any value in the two data frames, emission and annual run-hour, passes the annual limit or not. If any value passes the limit, create a string variable of the warning message. If not, create a string variable of the safe message.
- Save the emission data frame, annual run-hour data frame, and the string variable in the sheet 'Annual_Limit' in the summary excel file. Then go back to the section to get user input for the next action which is report, or mainMenu.

Project Fulfillment Estimation:

The final system will meet the overall requirements of the project in the following manner:
Variables

- System will use variables to calculate the overall emissions data, to validate user access rights, and to hold data prior to entry into the database

Flow control (if ... else, and loops)

- Loops utilized to validate user entries
- Loop utilized to collect all data based on selected date range

Functions

- Function for adding new data to the database
- Function for data validation loop
- Function for formatting of output data

Lists and Dictionaries

- Dictionary used to hold data prior to entry into database
- Tupple used for generator specification data such as emission rates, fuel usage, and capacity
- List used for data collection prior to export into EPA report view

Multiple modules

- Datetime module to log run time data
- Math module to calculate run time, fuel usage, and emissions

File or database CRUD (Create, Read, Update, Delete) operations.

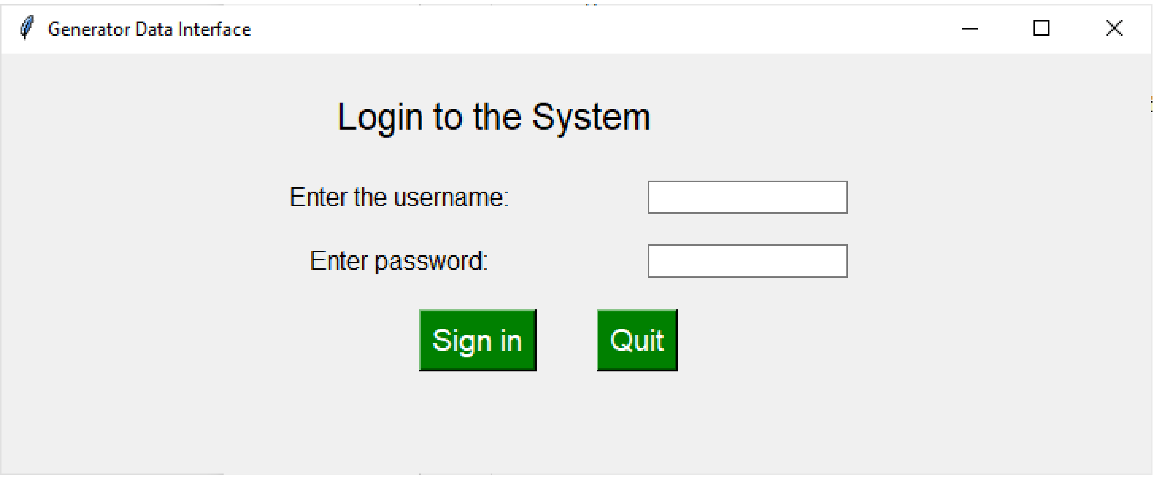
- Currently using a csv file to hold data. The program will have the functionality to allow admins to view and edit data through the GUI.

Data computation and visualization

- System will computed fuel usage and emissions data
- That data will be exported to the EPA report format
- System will provide a view function for raw run data.

# USER MANUAL

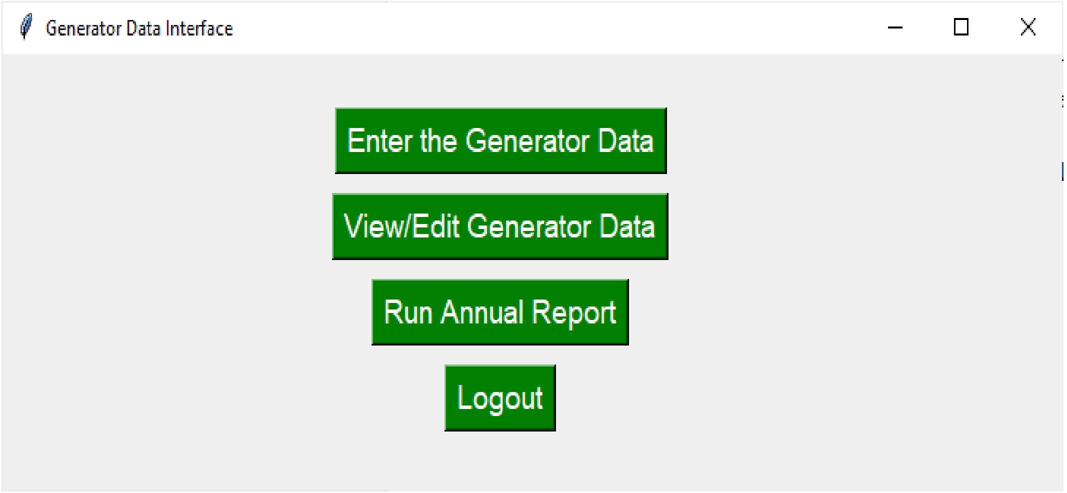
## Login



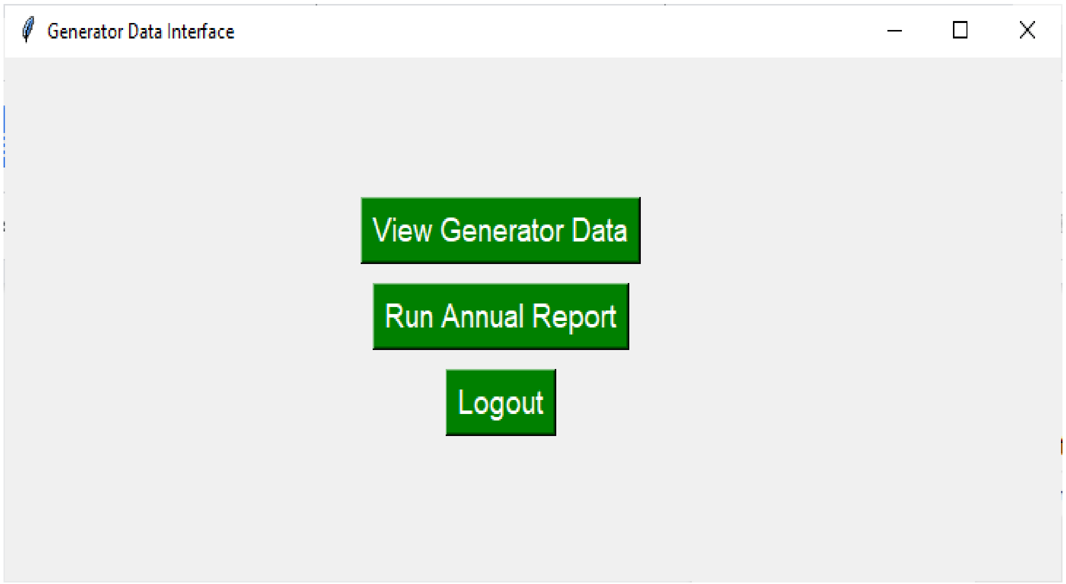
- User enters username and password in associated fields.
- Upon sign in the system validates username and password combination and assigns rights based on the username.
- Username and password are case specific. Incorrect entries will result in an error.
- Contact a system administrator for login credentials.
- Default credentials:
    - Administrator
        - Username: admin Password: admin
    - Basic User
        - Username: user Password: pass
- (It is recommended that system administrator changes these default credentials after install)
- Clicking Logout button on any screen will log the user out of the system and return you to the login screen.
- Clicking the “Quit” button will exit the program.

## Main Menus

- Administrator main menu

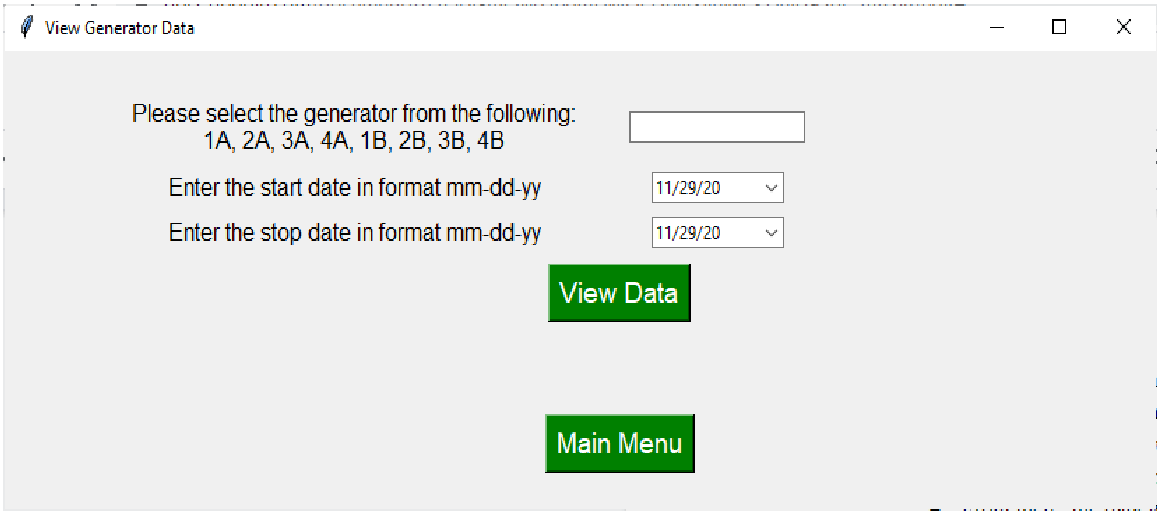


- Basic User main menu



- Main menus are used to navigate to the various functions available for the various user types.
- The Logout button will log out the current user and return them to the Login screen

## Viewing Data


View Generator Data

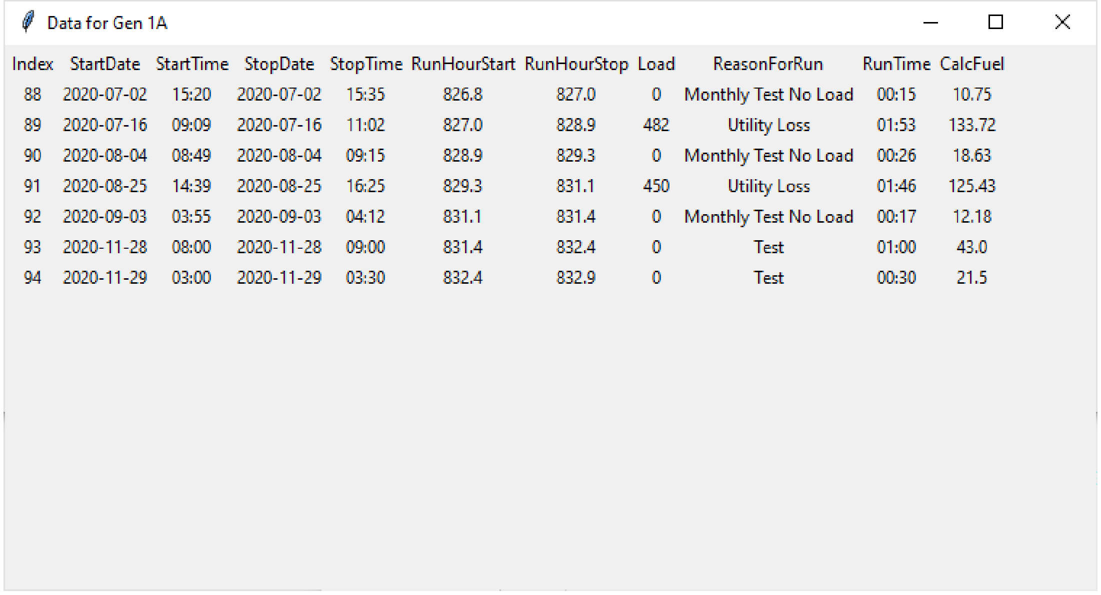
Please select the generator from the following:
1A, 2A, 3A, 4A, 1B, 2B, 3B, 4B

Enter the start date in format mm-dd-yy 11/29/20

Enter the stop date in format mm-dd-yy 11/29/20

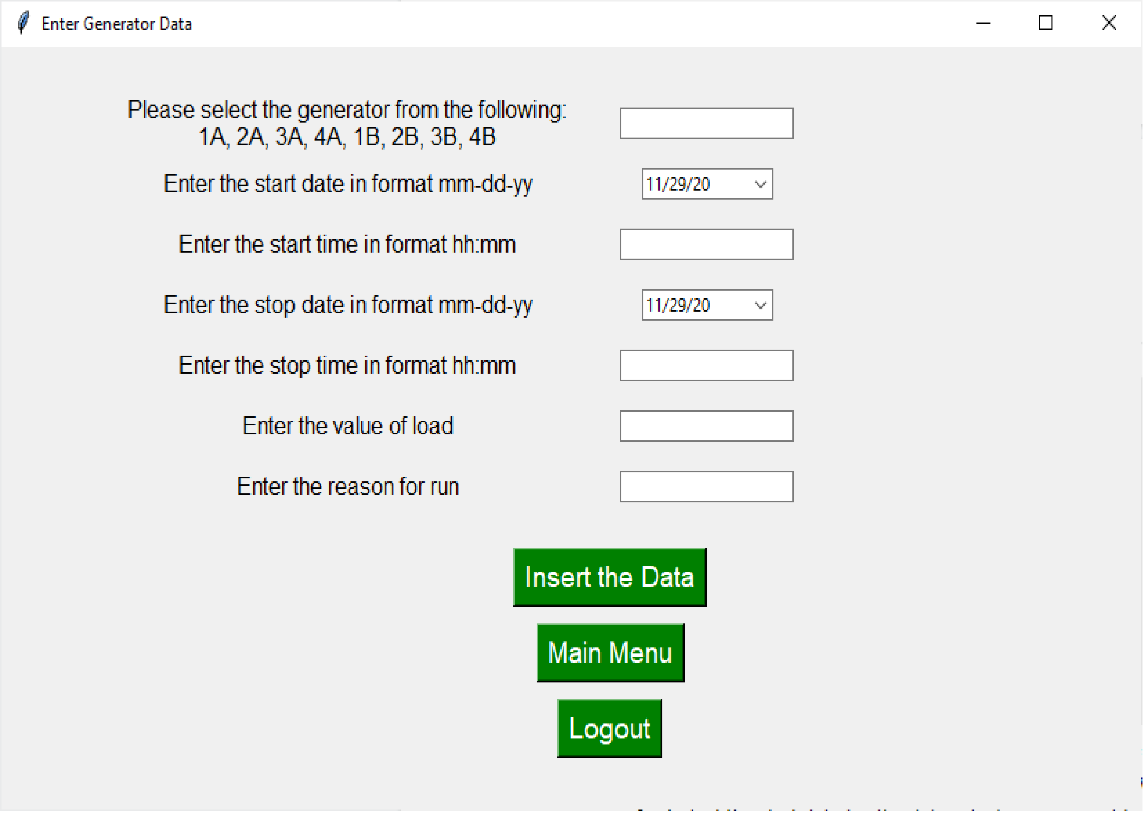
- Enter the generator identifier in the freeform field (i.e. 1A, 2A, etc)
  - (If the generator name does not exist an error will be displayed)
- Select the start date for the data selection you want to view and the end date for the data you want to view. This may be done by using the pro down calendar or by entering the date in the associated field in the format MM/DD/YY.
- Click the View Data button and a separate window will display the results.
- Select Main Menu at any time to return to the previous screen

View window example:


Data for Gen 1A

| Index | StartDate | StartTime | StopDate | StopTime | RunHourStart | RunHourStop | Load | ReasonForRun | RunTime | CalcFuel |
|---|---|---|---|---|---|---|---|---|---|---|
| 88 | 2020-07-02 | 15:20 | 2020-07-02 | 15:35 | 826.8 | 827.0 | 0 | Monthly Test No Load | 00:15 | 10.75 |
| 89 | 2020-07-16 | 09:09 | 2020-07-16 | 11:02 | 827.0 | 828.9 | 482 | Utility Loss | 01:53 | 133.72 |
| 90 | 2020-08-04 | 08:49 | 2020-08-04 | 09:15 | 828.9 | 829.3 | 0 | Monthly Test No Load | 00:26 | 18.63 |
| 91 | 2020-08-25 | 14:39 | 2020-08-25 | 16:25 | 829.3 | 831.1 | 450 | Utility Loss | 01:46 | 125.43 |
| 92 | 2020-09-03 | 03:55 | 2020-09-03 | 04:12 | 831.1 | 831.4 | 0 | Monthly Test No Load | 00:17 | 12.18 |
| 93 | 2020-11-28 | 08:00 | 2020-11-28 | 09:00 | 831.4 | 832.4 | 0 | Test | 01:00 | 43.0 |
| 94 | 2020-11-29 | 03:00 | 2020-11-29 | 03:30 | 832.4 | 832.9 | 0 | Test | 00:30 | 21.5 |

Entering Data (administrator only)



Enter Generator Data

Please select the generator from the following:
1A, 2A, 3A, 4A, 1B, 2B, 3B, 4B

Enter the start date in format mm-dd-yy 11/29/20

Enter the start time in format hh:mm

Enter the stop date in format mm-dd-yy 11/29/20

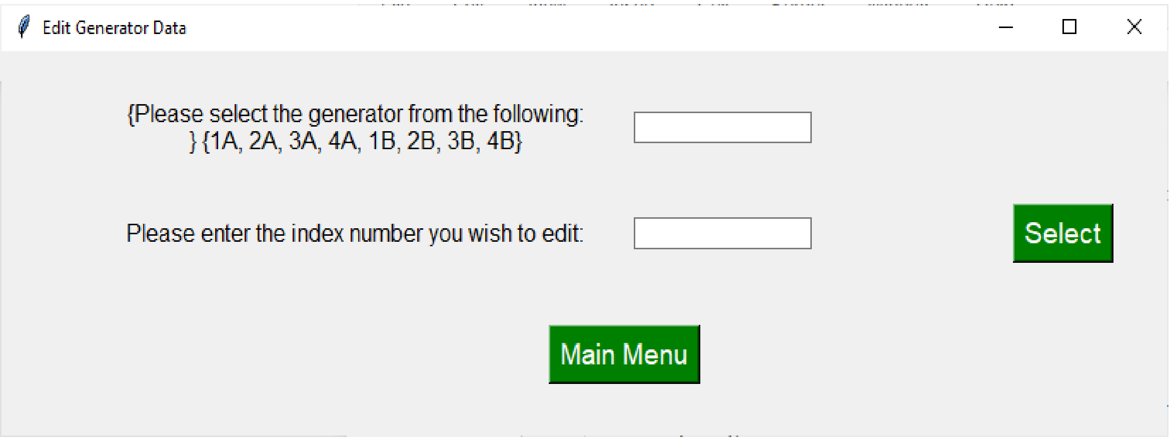
Enter the stop time in format hh:mm

Enter the value of load

Enter the reason for run

- This function is only available to users with administrator credentials
- User will enter the generator name for which data needs to be logged.
- Select the start date and the stop date for the data you are entering. This may be done by using the pro down calendar or by entering the date in the associated field in the format MM/DD/YY.
- Time entries should be made using a 24 hour clock with a colon separating hours and minutes (i.e. 09:00, 13:00, etc) and must be in the format of HH:MM.
- Click the "Insert the Data" button to add the entry to the data file
- Click the "Main Menu" button to return to the previous Menu
- Click the logout button the log the user out of the system and return to the Login window

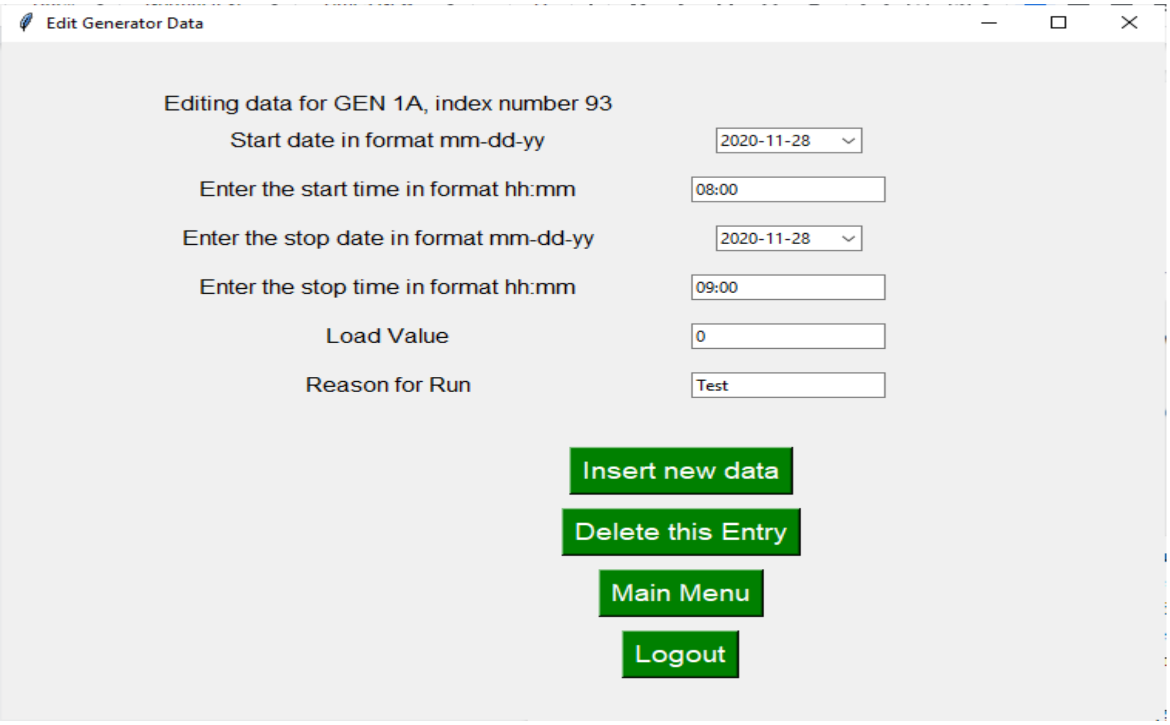
## Editing Data (administrator only)


Edit Generator Data

{Please select the generator from the following: } {1A, 2A, 3A, 4A, 1B, 2B, 3B, 4B}

Please enter the index number you wish to edit:

- This function is only available to users with administrator credentials
- User will enter the generator name for which data needs to be edited and the index number for the data entry that needs to be modified.
  - Note: Index number can be found using the view data function.
- The system will preload the existing data for the index number selected.


Edit Generator Data

Editing data for GEN 1A, index number 93

Start date in format mm-dd-yy: 2020-11-28

Enter the start time in format hh:mm: 08:00

Enter the stop date in format mm-dd-yy: 2020-11-28

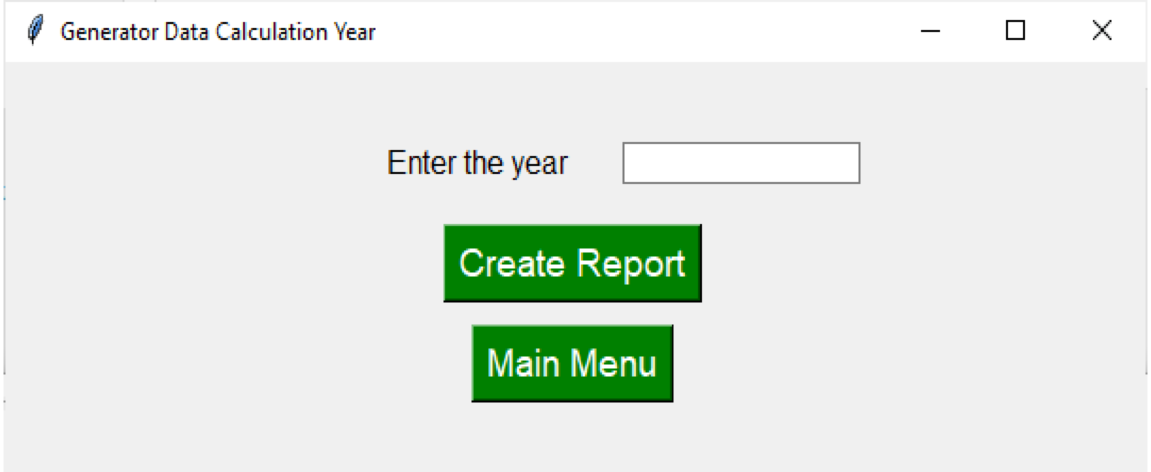
Enter the stop time in format hh:mm: 09:00

Load Value: 0

Reason for Run: Test

- Make necessary changes to the fields for the entry and click “Insert New Data”.
- Clicking “Delete this Entry” will delete the entire entry from the data set.
- Click the “Main Menu” button to return to the previous Menu
- Click the logout button the log the user out of the system and return to Login.

## Running Report



- Users enter the year for which they want to generate the report. Year should be entered in the format YYYY.
- Click the create report button and a summary report file will be created with the run time and emission data for the year for each generator. The summary will also show applicable limits and notify the user if they are getting close to any of the applicable EPA limits.
- Clicking Main Menu will return user to the previous Menu

## Additional Information

- The following files must be located in the same directory as the program:
  - Gen_Data.xlsx
  - credentials.xlsx

## Peer comments and team response to suggestions from design presentation:

Peers brought up several presentation points including introducing ourselves, making the text more legible on the powerpoint, and simplifying the explanation of the idea. These inputs were integrated into our final presentation.

Several peers brought up the idea of automated data entry  for generator runs (i.e. push start button and start date, time, load, are pulled from a management system directly). This functionality was thought of by the team for a future iteration but was beyond the scope of the initial deployment. The lack of a live data source to draw from would have made coding difficult and designing a live data emulator was far beyond the scope of work.

Instructor recommended varying levels of permissions for users to prevent data corruption. An administrator flag was built in with separate available functionality if the user logging in is designated as an admin.

Peers and instructor brought up the need for an edit function and data validation. Both of which were part of the initial design request. Functionality was coded in to allow administrative users to edit or delete data from the data file using the interface. Data validation was also coded in to verify dates, times, generator names, fields completely filled out, and to make sure start time and date came before stop time and date.

Peers asked if it was possible to build in functionality to add generators or change EPA data. At this time the data and generators are hard coded into the system. New generator registration and changing of EPA limits was beyond the scope as these activities happen very infrequently and were not key points of functionality for users. However, we did modify the code so rather than using hard coded data spread throughout the program, generator names were integrated into global variables that could easily be replaced by a new generator registration function in the future.