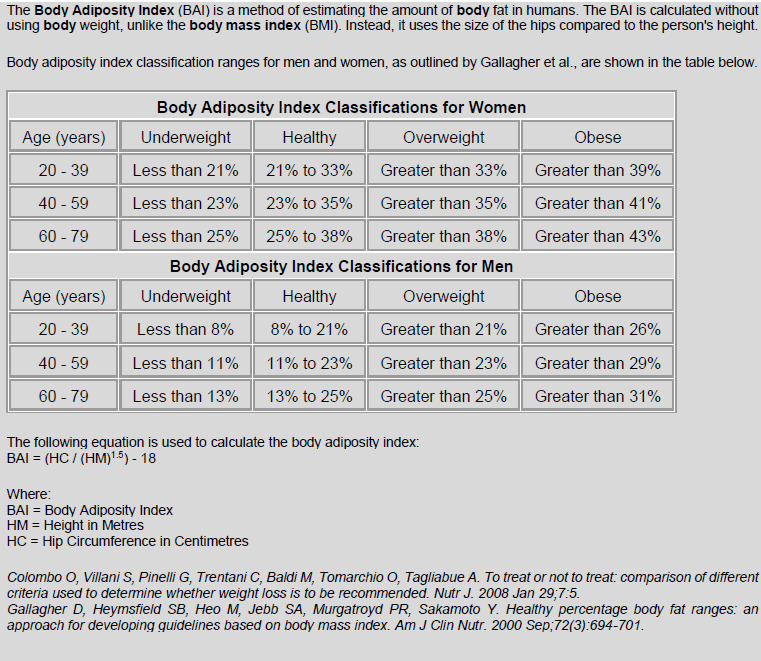
Assignment Solution

# Assignment

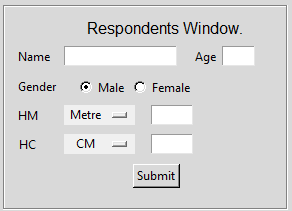


# Decomposition, Planning and Design

Project is decomposed into following sub-parts:

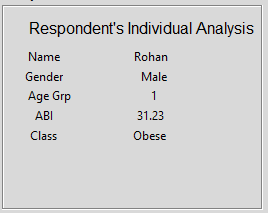
## Respondents Window

This window provides facility to give inputs to the application of a respondent. This window contains entry section for Name, Gender, Age, HM and HC in different linear scales.



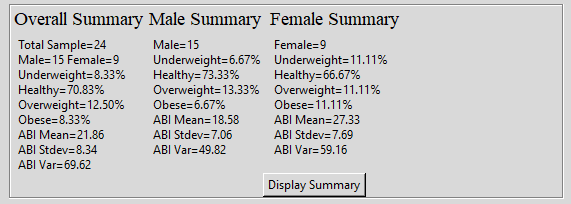
## Individual Respondents Analysis

This window will display processed data of individual at the time of input. There are three age group, 1 for 20-39, 2 for 40-59 and 3 for 60-79. After submitting the data of Rohan , output are shown.



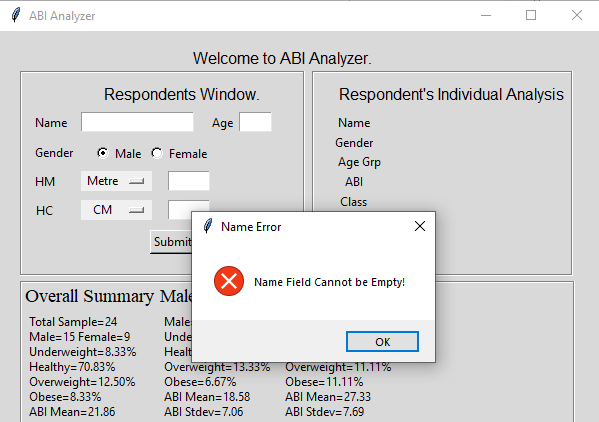
## Over Summary Window

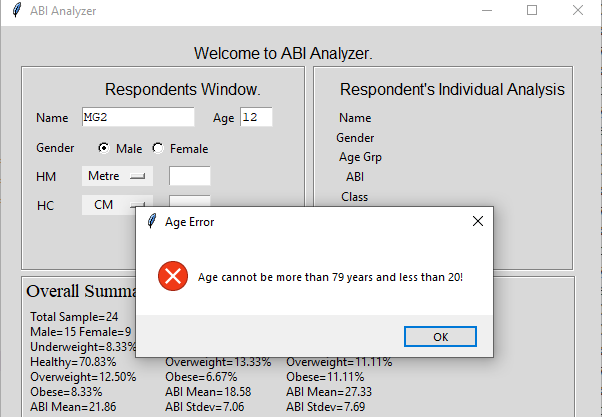
This section contains the analysis of the stored data. Any time “Display Summary” button is clicked, statistics of the latest updated data will be displayed. It contain three part, 1- Overall Summary of data, 2-Male Respondents Summary, and 3-Female Respondents Summary.

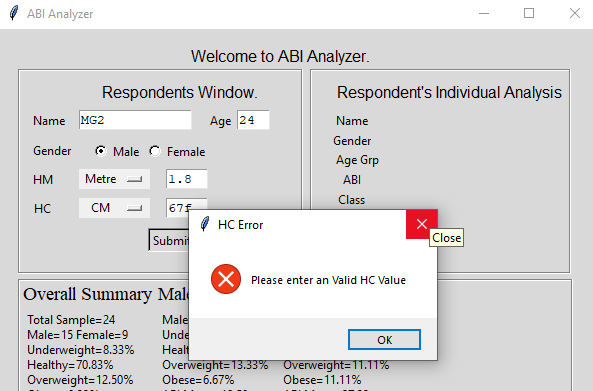


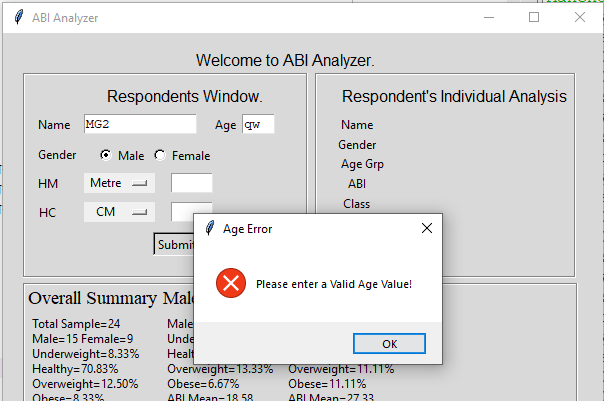
## Input Validation

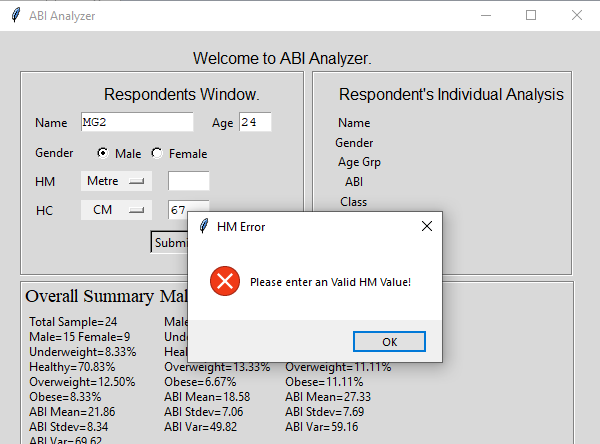
This section validates each entry and data which is to be submitted. Data is submitted only when all data entries are valid. For each invalid entries alert message are shown.







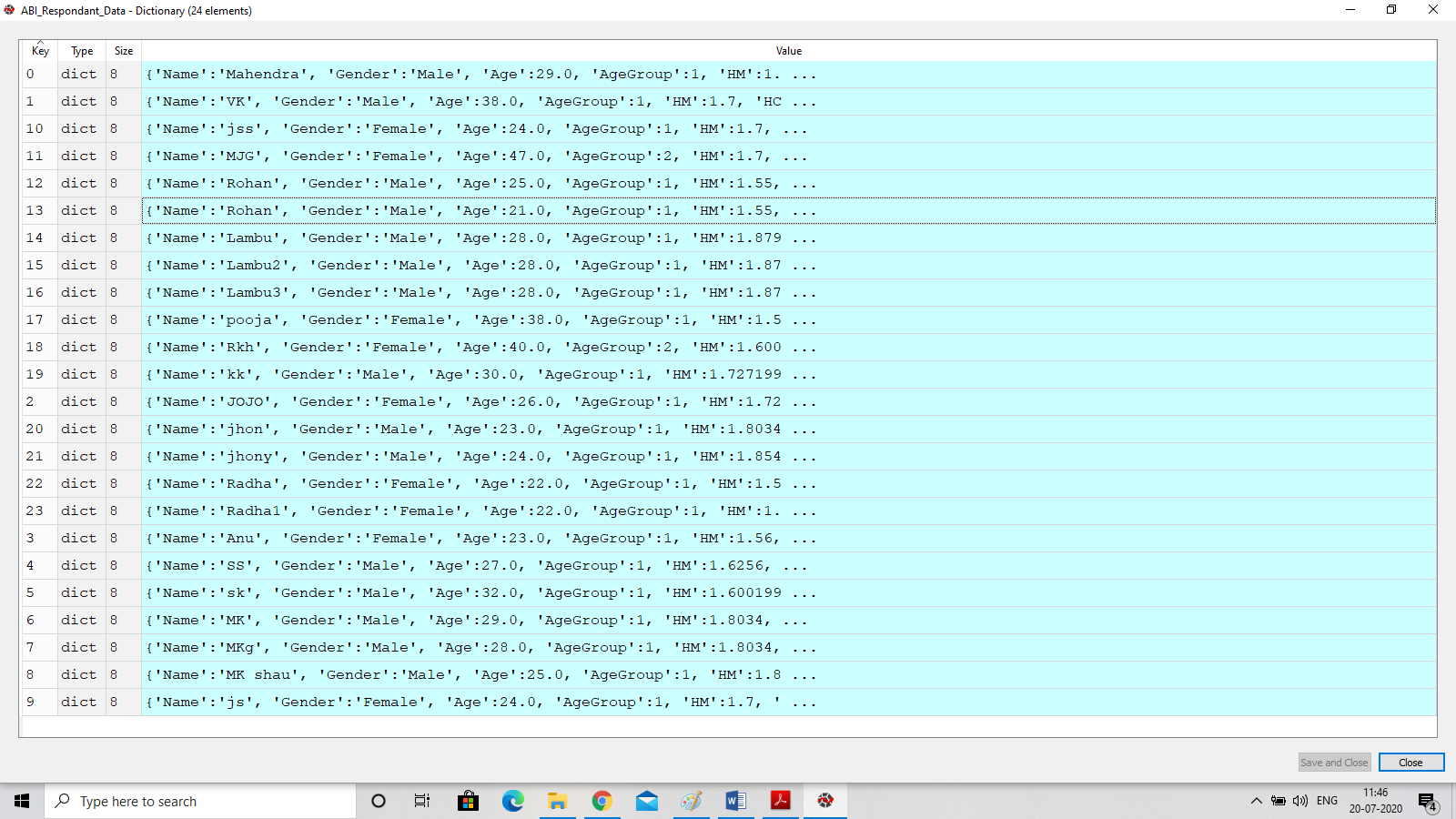




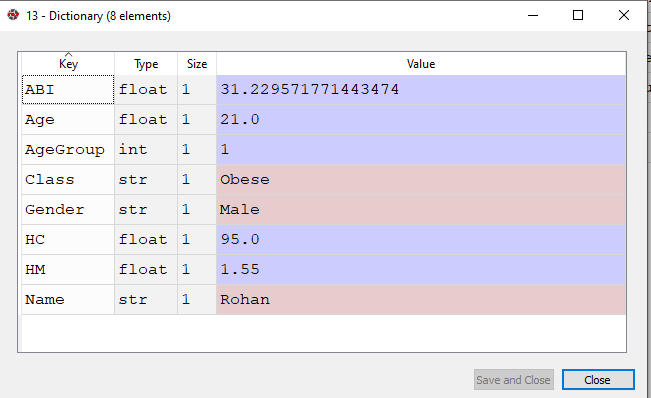
## Data Storage and Updation

Every time respondent submit individual data it is stored in runtime environment and when window is going close, before that all the data gets updated in *“ABI\_Respondant\_Data.json”* file which is stored in same directory.

Complete Data



Individual Data



## Error Checking

Printed different numerical and flags value to track the error in the application.

These printed value can be seen in console and those values are clues of many types of errors

## Hierarchical Structure of Project.

## 

## Flowchart

Fill all entry Field

Check Entry

Submit Button

No

Is Data Valid

Yes

Store & Display

Yes

More Data?

No

Analyze Data

Display All Analyze Data

## Data Dictionary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identifier Name** | **Variable/Constant** | **Data Type** | **Global/Local** | **How Var is Used?** |
| ABI\_Respondant\_Data | Variable | Dictionary | Global | After importing data from .json file, stored in this dictionary in runtime environment. |
| SampleCounter\_n | Variable | integer | Global | It keeps the count of number of data samples. |
| ABI\_Respondant\_Data.json | file | dictionary | Global | All the data are store for future analysis. Whenever application opens data is imported from this file. |
| Data\_summary() | function |  |  | After clicking “Display Summary” this function is called to display summary of data. |
| Female\_Summary(df\_t) |  |  |  | It is called by Data\_summary() function to display Male data summary. |
| Male\_Summary(df\_t) |  |  |  | It is called by Data\_summary() function to display Female data summary. |
| GenderCount | Variable | dictionary | Local | Use in different function to store gender count. |
| ClassCount | Variable | Dictionary | Local | Use in different function to store Class count. |
| MaleTextVar | Variable | String | Global | Keeps Male Analysis as String to display in frames. |
| FemaleTextVar | Variable | String | Global | Keeps Female Analysis as String to display in frames. |
| OverAllTextVar | Variable | String | Global | Keeps Overall Analysis as String to display in frames. |
| Male\_data | Variable | Data Frame | Local | Keeps all data of Male |
| Female\_Data | Variable | Data Frame | Local | Keeps all data of Female. |
| update\_Data\_in\_File() | Function |  |  | Use to update entries into json file. |
| Close\_window() | Function |  |  | To close the Application. |
| Validate\_Data(e) | Function |  |  | Used to validate input data. |
| on\_Click() | Function |  |  | Called when submit button clicked |
| Age | Variable | float | Local | Store Age value of individual |
| HM | Variable | float | float | Keep Height of individual in metre. |
| HC | Variable | float |  | Keep Hip circumference of individual in in CM. |
| Gender | Variable | String | Local | Keep gender of individual. |
| BAI\_Classifier(Gender, AgeGroup, BAI) | Function |  |  | Assigns Label to individuals class. |
| InsertData(RespondantName,Gender, Age, HM, HC) | Function |  |  | Insert data in runtime environment dictionary. |
| IndividualDataDisplay() | Function |  |  | To display Individual Analyzed data. |
| \_bgcolor | constant | String | Global | Bgcolor |
| \_fgcolor | constant | String | Global | fgcolor |
| top | constant | object | Global | Main Window object |
| EntryName | Variable | String | Global | Keep input name |
| EntryAge | Variable | String | Global | Keep input Age |
| EntryHM | Variable | String | Global | Keep input HM |
| EntryHC | Variable | String | Global | Keep input HC |
| scaleTypeHM | Variable | String | Global | Keeps Scale of HM |
| scaleTypeHC | Variable | String | Global | Keeps Scale of HC |
| genderVar | Variable | String | Global | Keeps the selected Gender Value |

.

# Implementation

In the making of this application following libraries and packages have been used.

### Tkinter: This package is used for GUI development

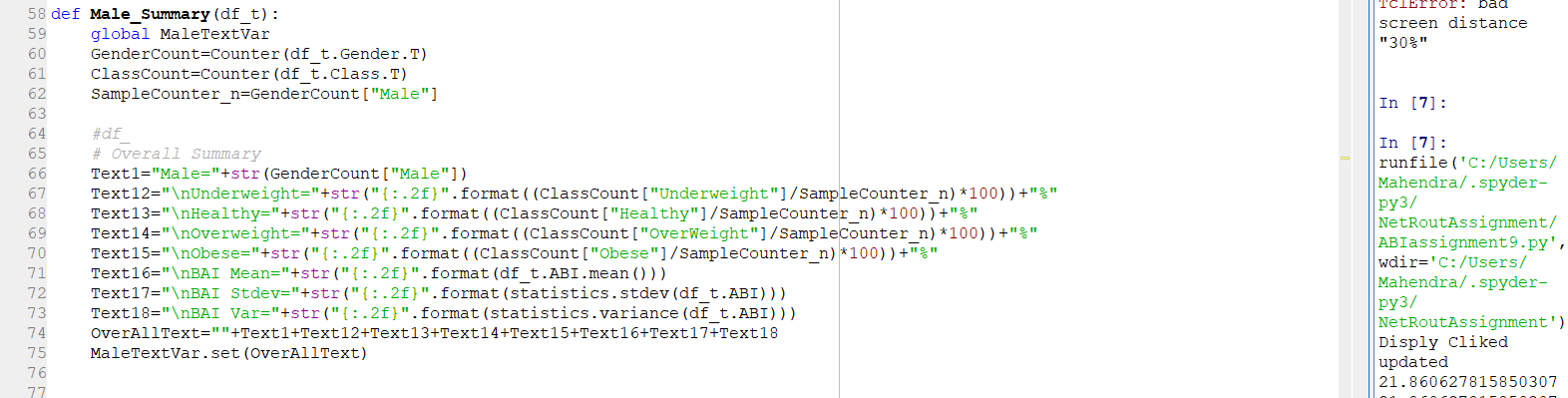
### Json: This package is used to store data.

* **Pandas**: For data selection and Manipulation.
* **Numpy**: To manipulate array data.
* **Collections**: counter is use to count individual attributes
* **Statistics**: From this packages different functions is used to get different stat value.

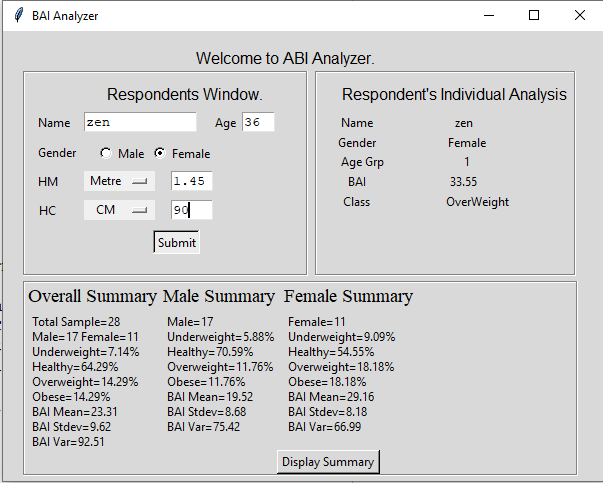
To write the code I use Spyder which is part of Anaconda Plateform.



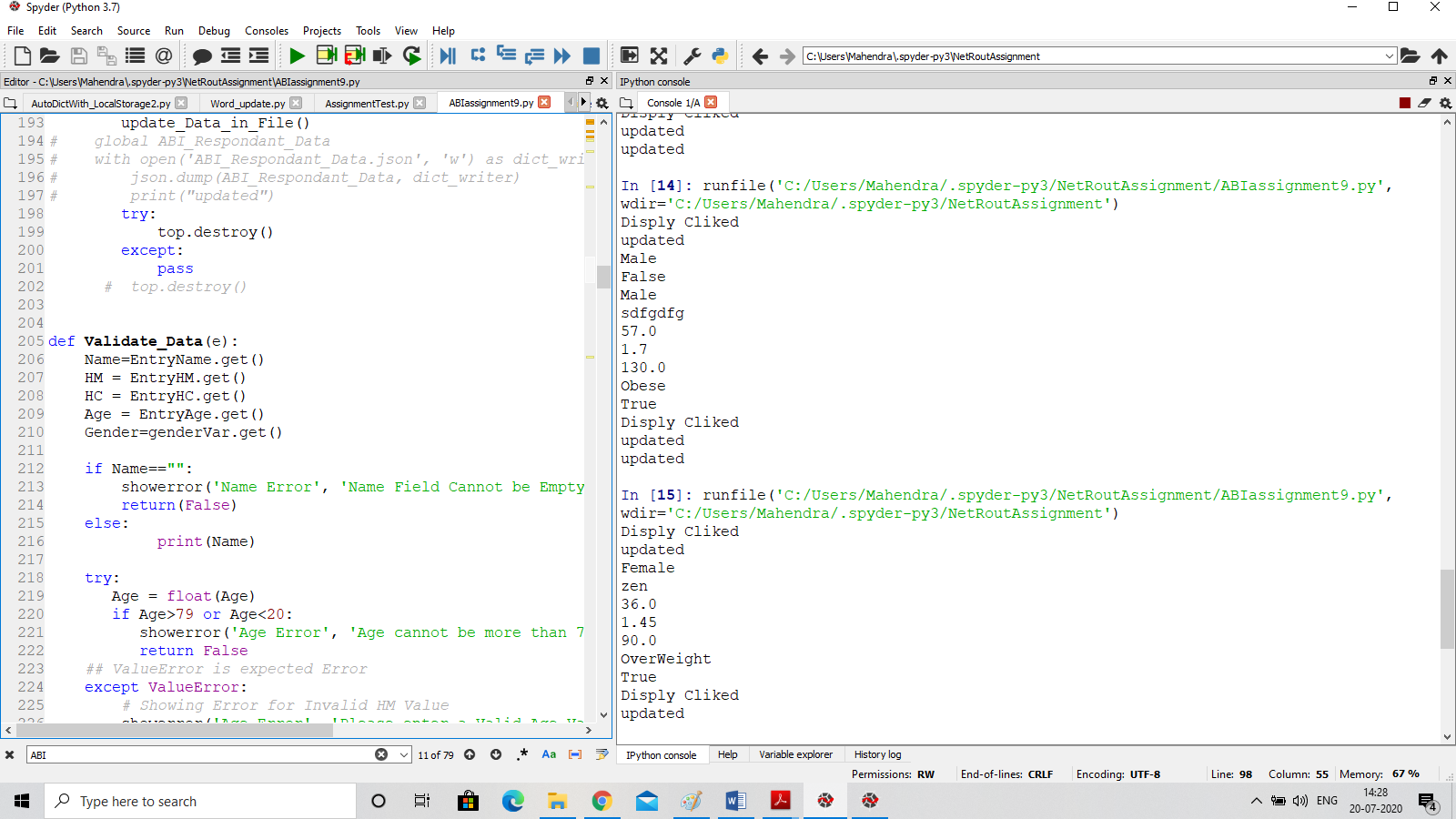




**Final Complete Window.**



**Console outputs to keep track errors If Any**



# Testing and Analysis

For testing I use few real data and analyzed the output. After few correction it starts working fine.

* **Situation1**: I forget to consider different scales of HM and HC, initially it was not doing good but handling scale it was fine.
* I tested same data with different scale , it was working fine,
* I tracked data entries in the local data dict and json file as.
* **Situation2**: Initially I forget to handle lower and upper limit of age, in that case it was giving unexpected output.
* **Situation3**: While displaying the Summary section, after clicking “Display Summary Button” it was not showing current updated summary. So before getting summary, updated json data file it was showing latest summary of data.

Similarly few more problem face during testing and analysis of Application.

# Review

After completing this application, I find some personal skill development which are as follows:

* Some more functions and libraries of python I got to know during this project.
* I got to know how to write a short and precise project report.
* How to manage time and focus on work and ignore unnecessary things.
* Before I was doing programming but was not that much focused.
* BAI and ABI are used interchangeably by mistake.

# Demonstration

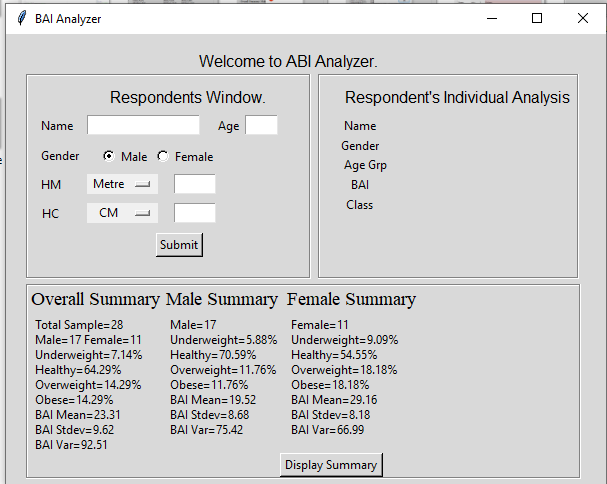
To use the application following steps.

1-Make a folder such as BAI\_application

2-Put *“BAIassignment10.py”* and “*ABI\_Respondant\_Data.json*” into above folder.

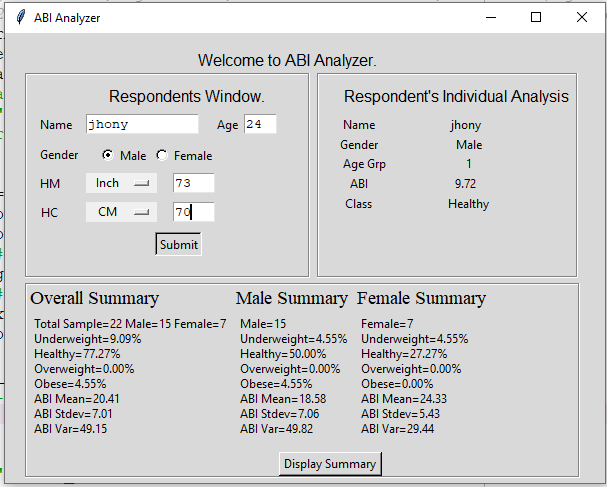
3- Open python file in Spyder and Run, or Run python file using CMD if python setup is already installed.

4- After running following window will open.

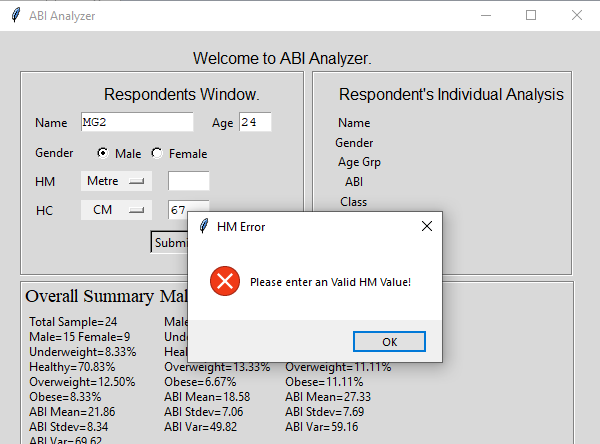
****

5-Fill Respondent window data correctly and click submit button. If all data is correct then following window will open.

In previous window Respondent Individual Analysis data was not present.



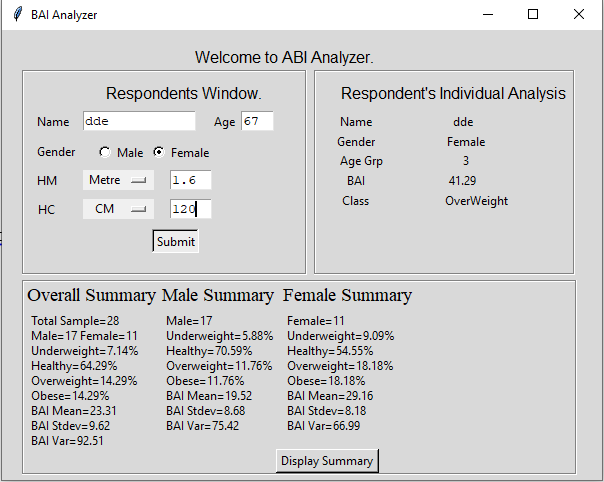
If there is any error in entry error message will show.



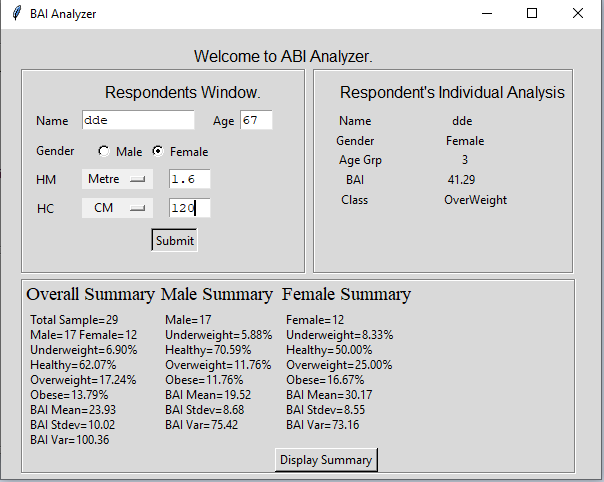
6- Repeat step 5 until you want submit the data.

7- Whenever you want to show overall summary click “Display Summary” button, updated summary will display.

Before display cliked



After Display clicked.



Note:

* Analysis of some other parameters such as Age, HM and HC can be done in similar way.
* These statistical value can also be displayed in console in case of no space in GUI.