**Adam Heavey**

**Coded the following files:**

* volume.py
* mass.py
* area.py
* test\_area.py
* test\_distance.py
* test\_mass.py
* test\_temperature.py
* test\_volume.py

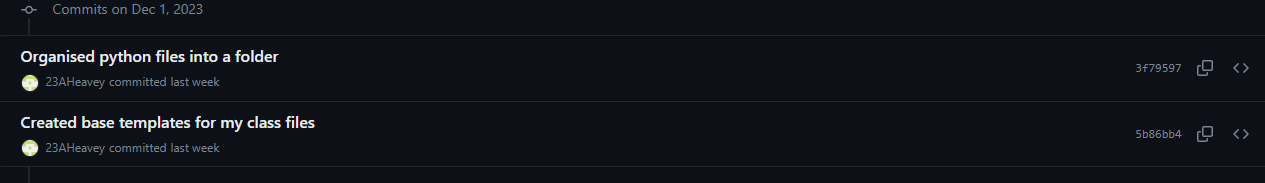
**Worked on the following documents:**

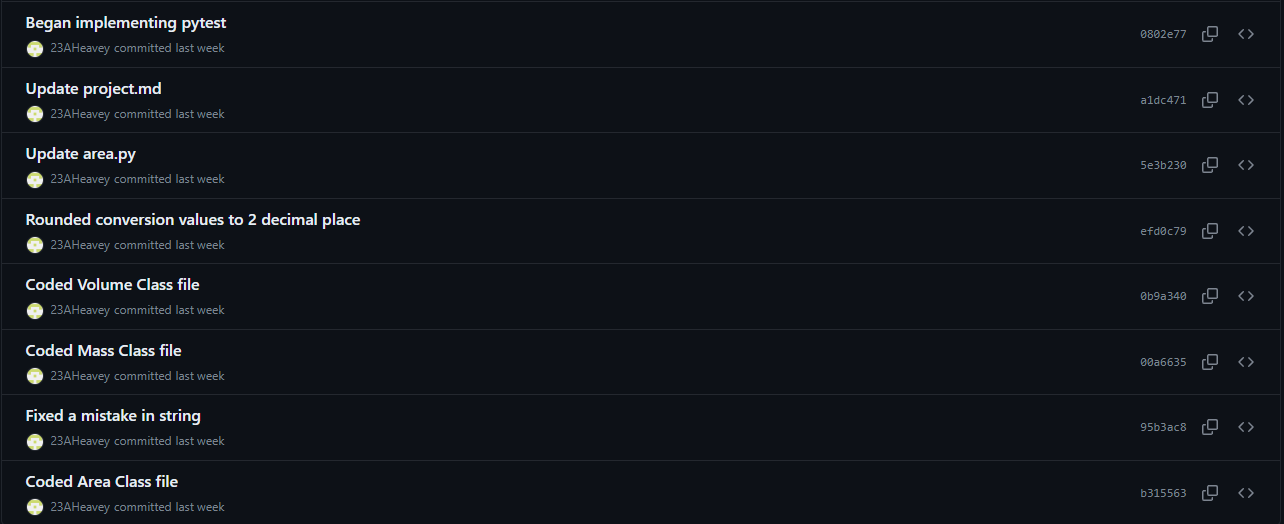
* Algorithm: Pseudocode
* Teamworking Document
* Testing & Screenshots Document

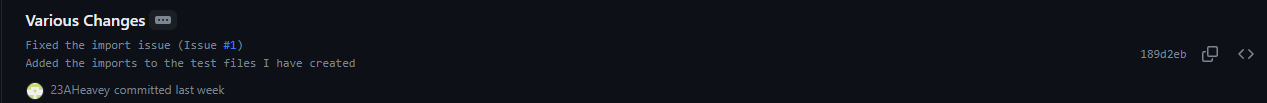
**Other contributions made to the project:**

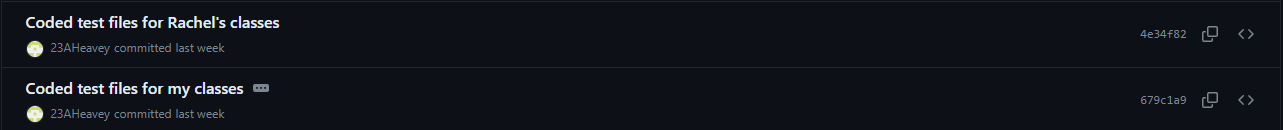
* Used GitHub and GitHub Desktop, along with my IDE to code and make any needed changes to the program. This helped us as a team to keep track of how the project was coming along.
* Helped to finish Data Dictionary.
* Optimised main menu file, as well as fixing bugs, which includes:
  + Try-Except-Else clauses.
    - This fixed the program from breaking if the user inputted a string instead of an integer/float value.
  + Changed user input for the conversion options to a float, instead of an integer.
    - It was causing an issue where if you entered a float value to convert, since it was not an integer it would print an error message. It will now accept decimal values.
  + Changed logic of the conversion functions completely.
    - The main issue was when asked to convert again, when given the choice of “y” and “n” for yes and no, any other input than “y” was considered as “n” regardless. You were able to just enter whatever you wanted to input, and it would return you to the main menu. I made the following changes to fix this issue:
      * I made a while True loop for the convert again input.
      * If “y” was entered, it would break out of the loop.
      * If “n” was entered, it would call the parent function.
      * Anything other than “y” or “n” entered would print an error message and continue to ask you if you would like to convert.
  + Made slight formatting change to conversion screens.

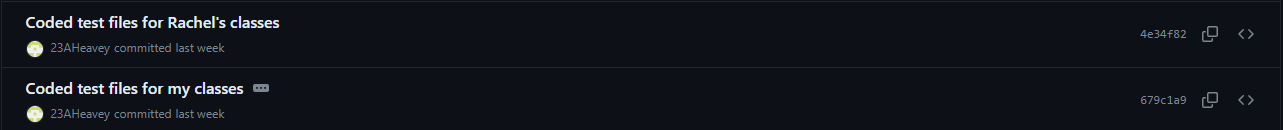
**Adam Heavey: Individual Effort Evidence (Major commits showcased)**

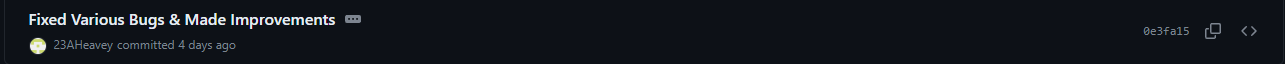


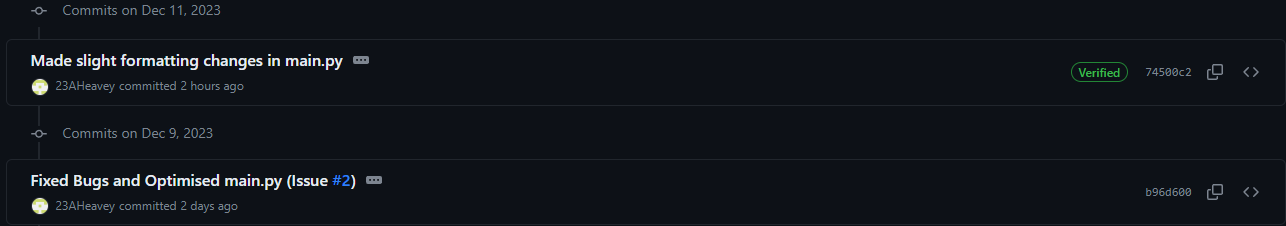












**Rachel Gillespie**

**Coded the following files:**

* distance.py
* temperature.py
* main.py

**Worked on the following documents:**

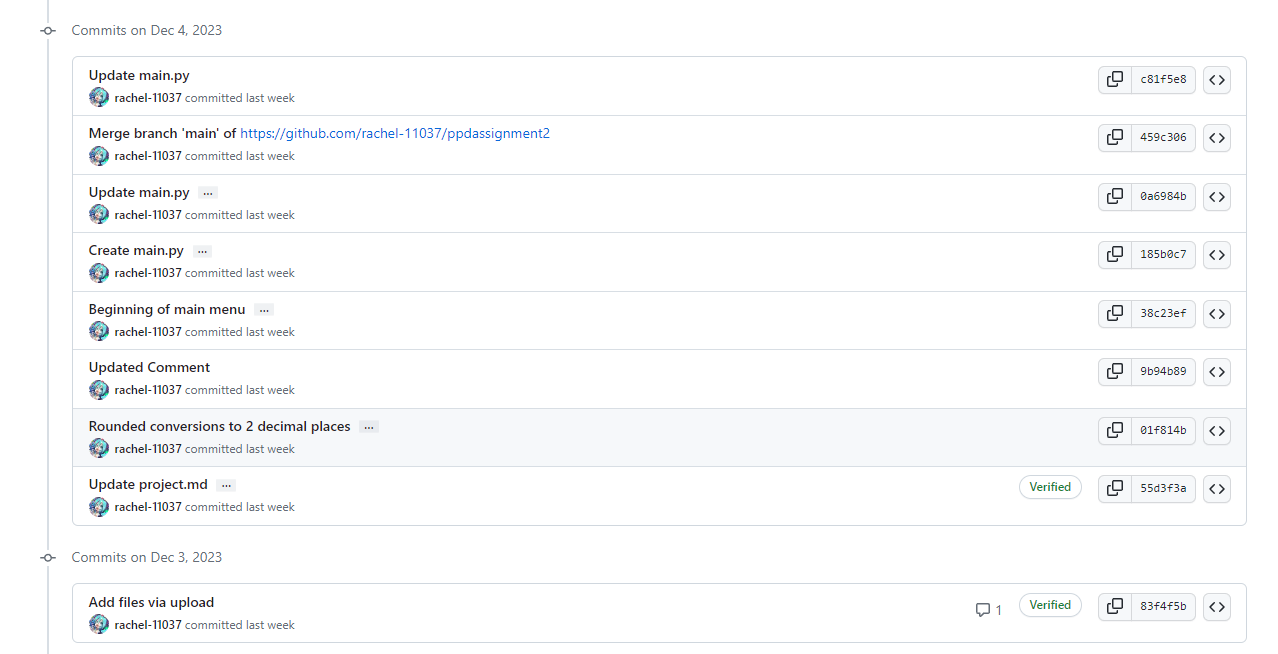
* Algorithm: Data Dictionary
* Added to this document my contributions

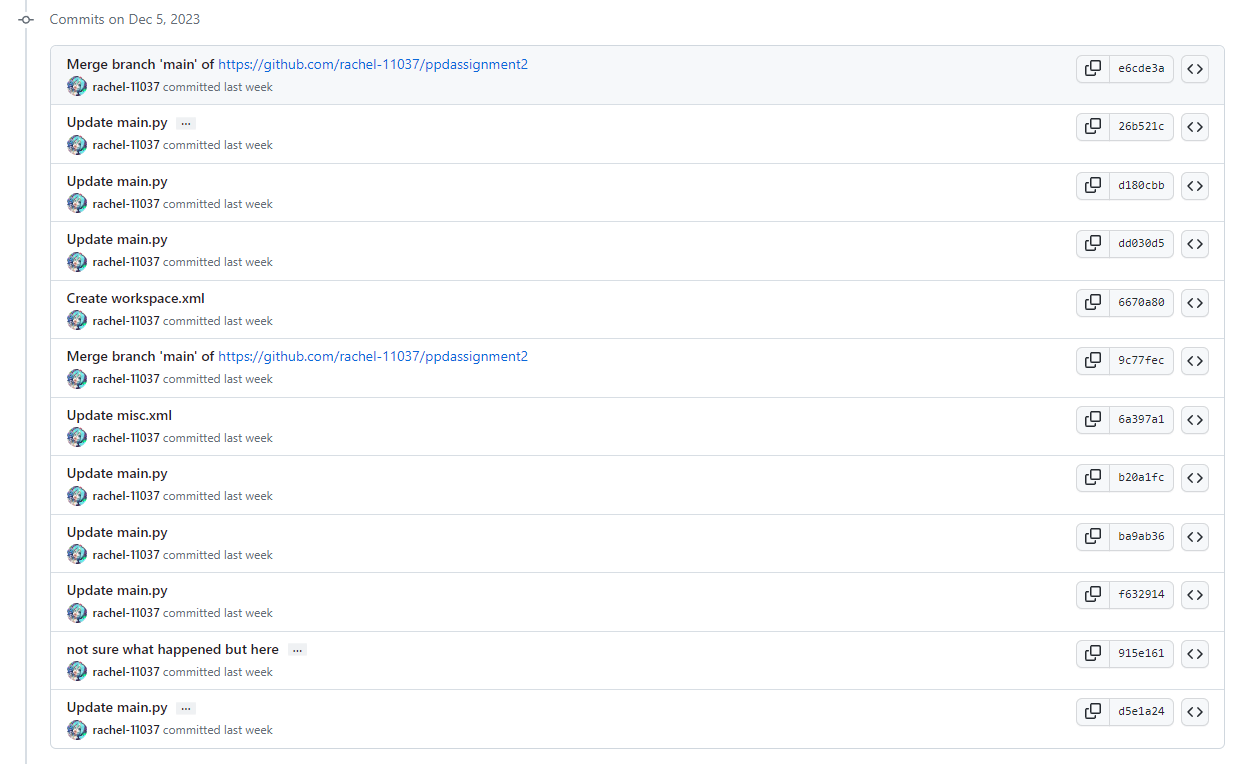
**Other contributions made to the project:**

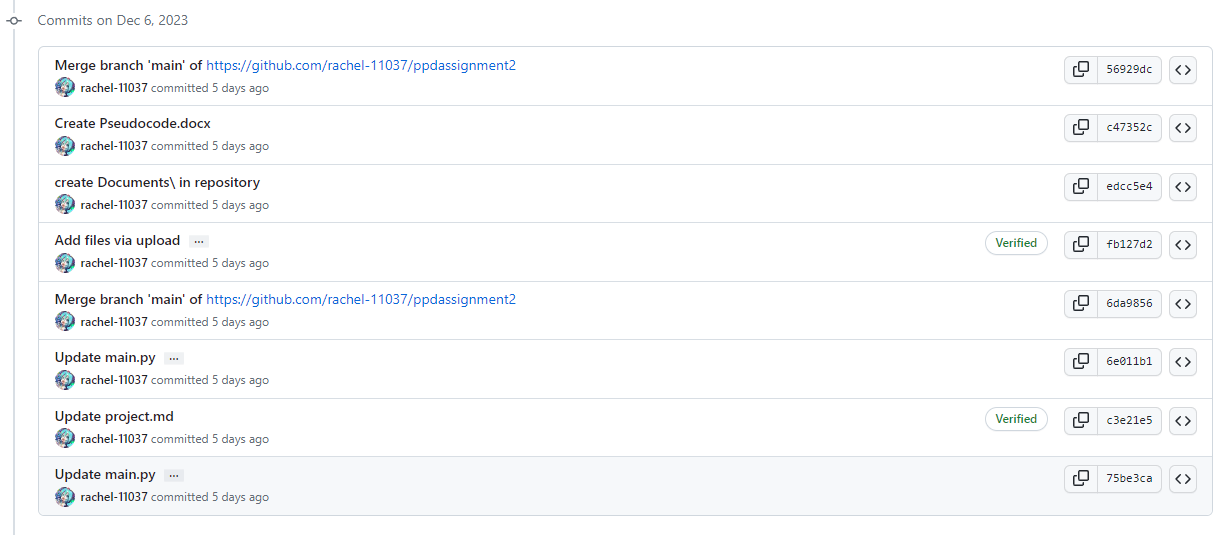
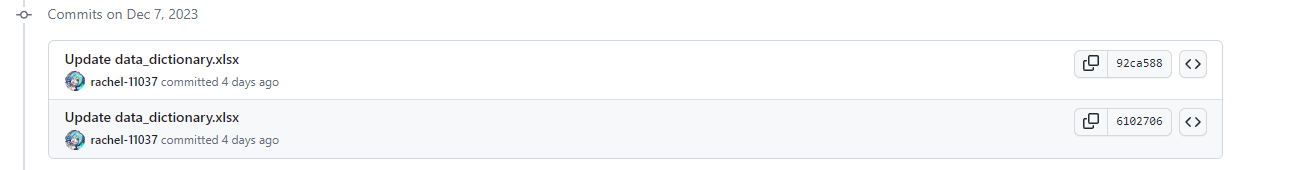
* Created and managed a GitHub repository to store and track all changes made to the program/project.
* Kept team updated with any commit that was made however minor.
* Made a few suggestions to optimise the main file code such as using PyInputPlus, a module that makes inputs in Python more efficient (they were declined, as we felt it was too much hassle to try and implement it)
* Created the main file which Adam then improved upon.

**Rachel Gillespie: Individual Effort Evidence (Major commits showcased)**

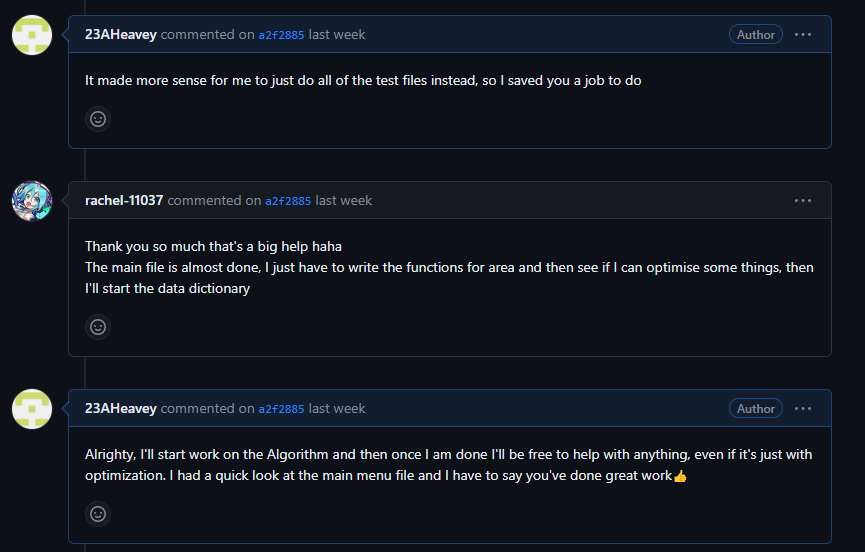
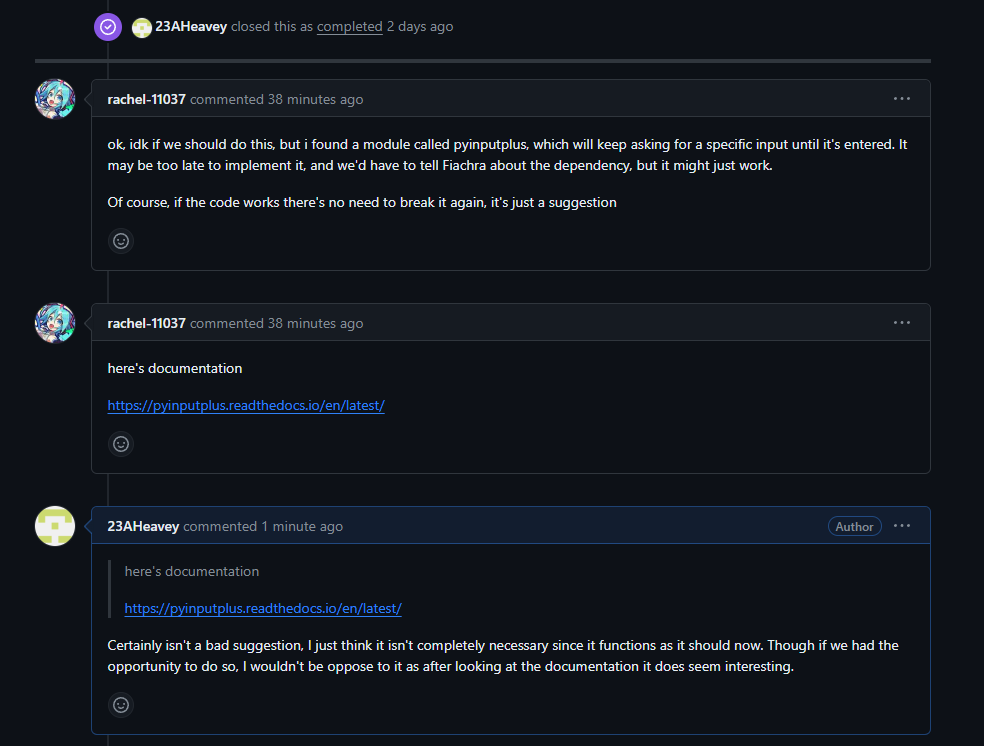








**Team Discussion Evidence**



**Project.md File**

A screenshot of a computer

Description automatically generated

This markdown file was used to give more clarity on how the project should look on completion, and who should do what tasks. This file was continuously updated throughout the lifetime of the project. It allows us as a team to designate tasks to each other and give us both a better idea of where we are each at in the project.