Faculty of Engineering & Technology

Linux Laboratory – ENCS3130

Python Project – Medical Test Management System

Prepared by:

Name: Francis Miadi **Number:** 1210100

Name: Miar Taweel **Number:** 1210447

Instructor: Dr. Mohammad Jubran

Teaching Assistant : Eng. Ahed Mafarjeh

Section: 1

Hand in date : 23/08/2024

Abstract:

In this project, the goal is to develop a Python-based program that effectively manages medical test data for individual patients. The program should allow users to retrieve specific test results by patient ID or test type, add new test results, and update existing records. This system will function as a basic tool for managing patient records.

The system will work with and manage two primary files: one containing medical records and the other containing medical test data. The medical records file will include essential patient information, such as patient IDs, test results, and other relevant details. The medical tests file will store data related to various tests conducted at our medical center, including test types, dates, and results. The system will be designed to efficiently access and update the medical record file, enabling the seamless addition of new test results, updating of existing records, deletion of outdated or incorrect entries, and retrieval of specific test results based on patient ID, test type, or other criteria.

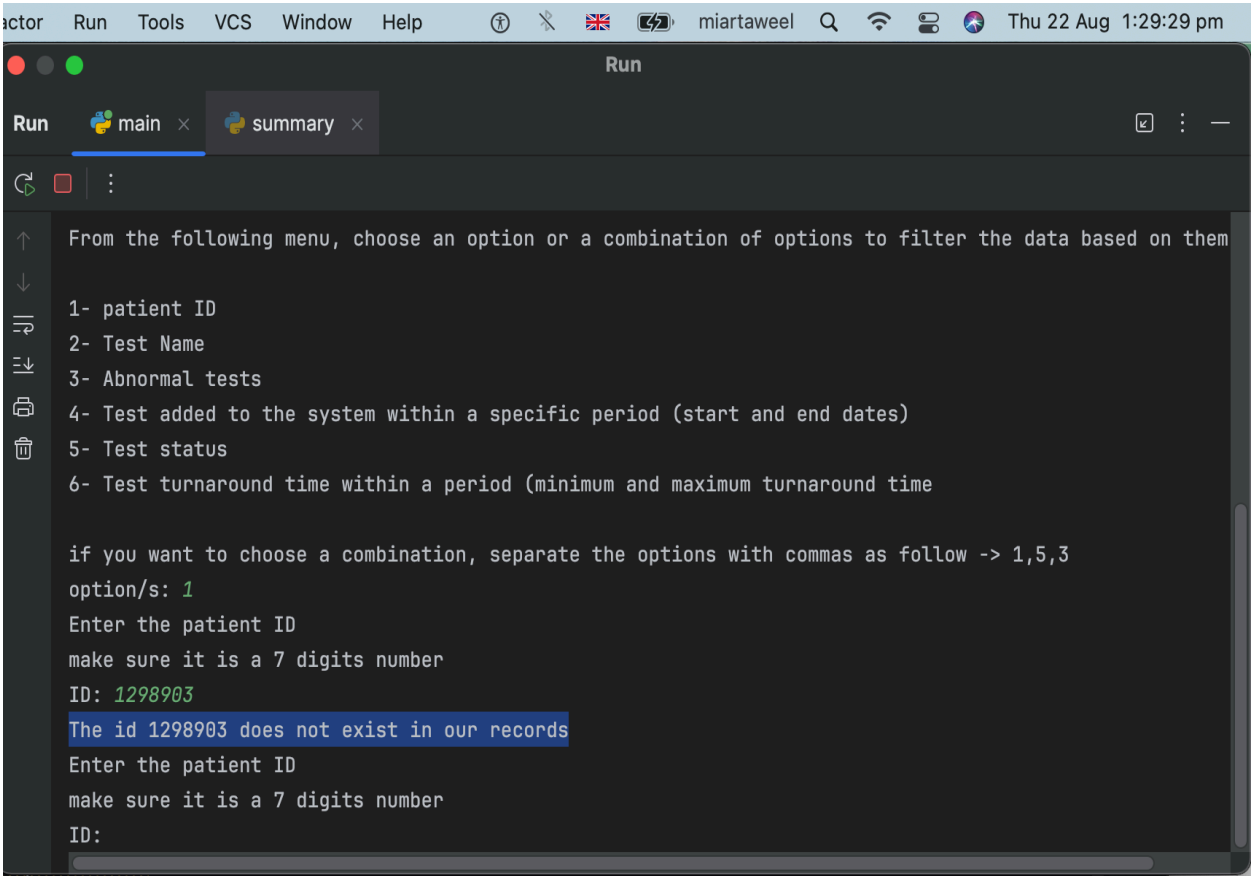
- This report presents a series of test cases and their results to demonstrate the efficiency and effectiveness of the developed system.

Table of Contents

1. Error Handling:

1.1. Non Existent Patient:

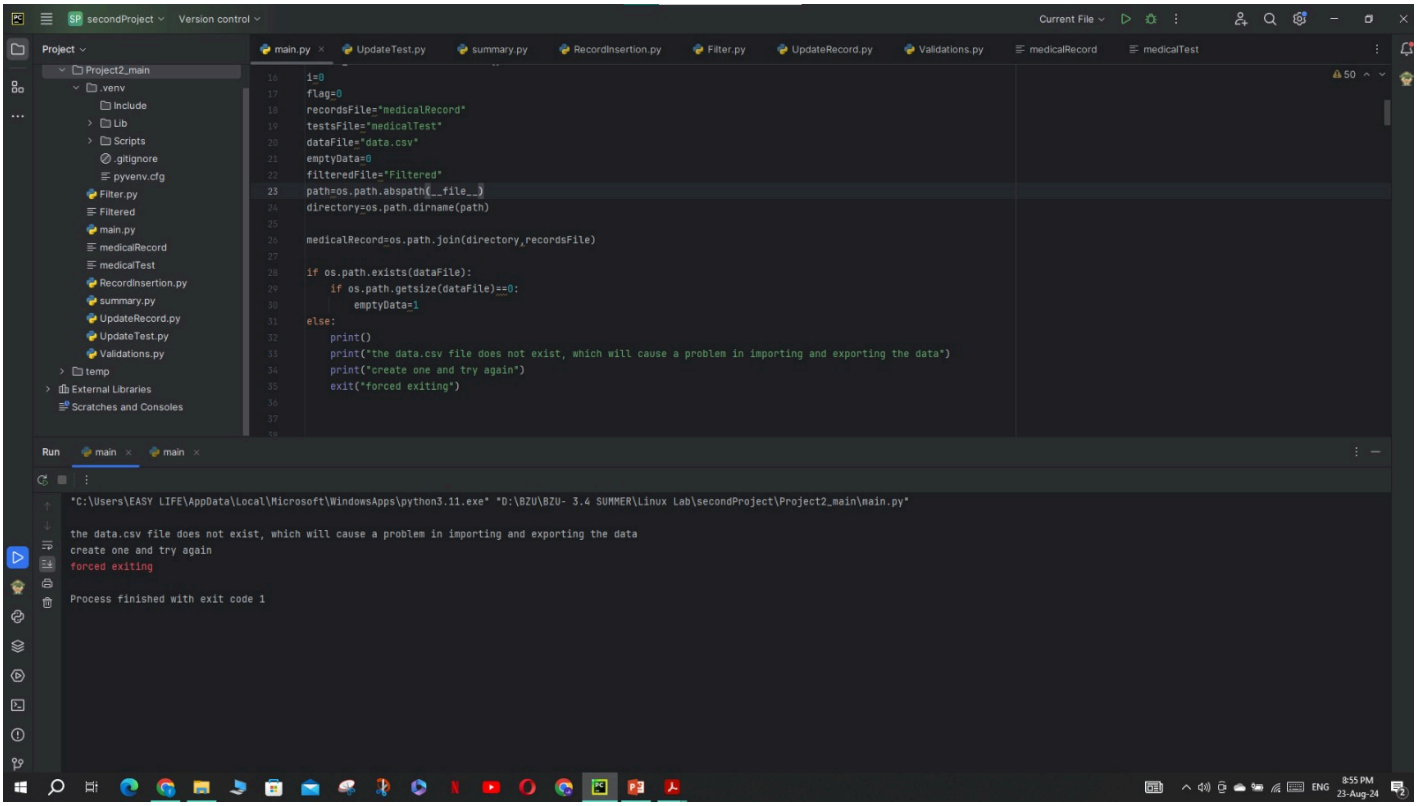
When performing a specific operation (Add, Update ,filter) on the record file , the patient ID is inserted to determine which record to operate on , the following image represents **the handling of the error of updating a non existent patient** :



As viewed in the previous figure , the non-existence of the record with that patient ID was detected , thus the program requests the user to enter it again.

1.2. Non Existent .CSV file:

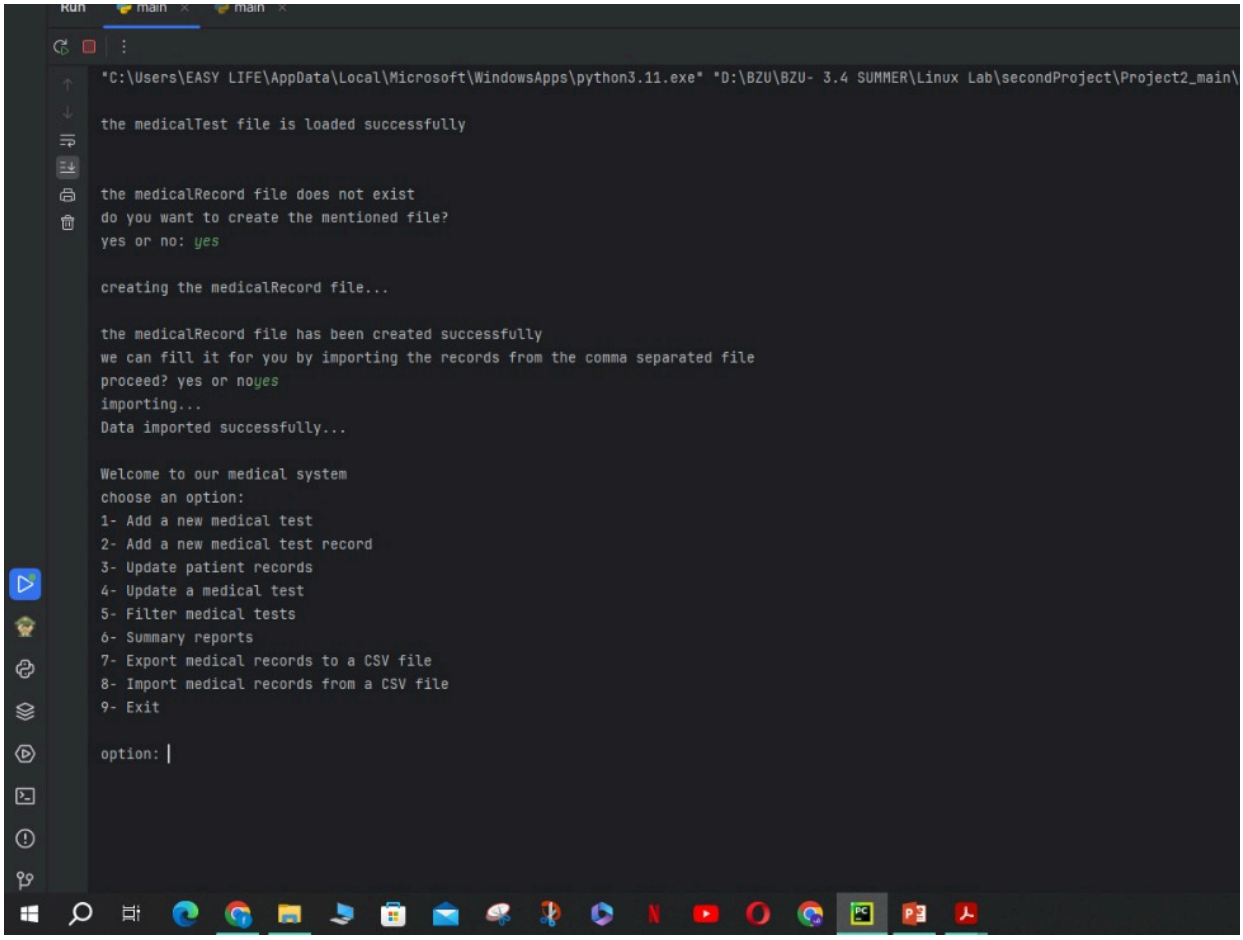
When importing or exporting from the record file, a .csv file should exist to import or export to, thus their existence is checked before starting the program ,the following image s represents **the handling of the error of the non existence of the .csv** :



As viewed in the previous figure , the non-existence of the .csv was detected , but since the medical record files' content depends on it , the program quits since it can't be executed without data.

1.3. Non Existent Record file:

When performing any operation on the record file, the record file and the test file should exist , thus their existence is checked before starting the program ,the following image s represents **the handling of the error of the non existence of the record file** :

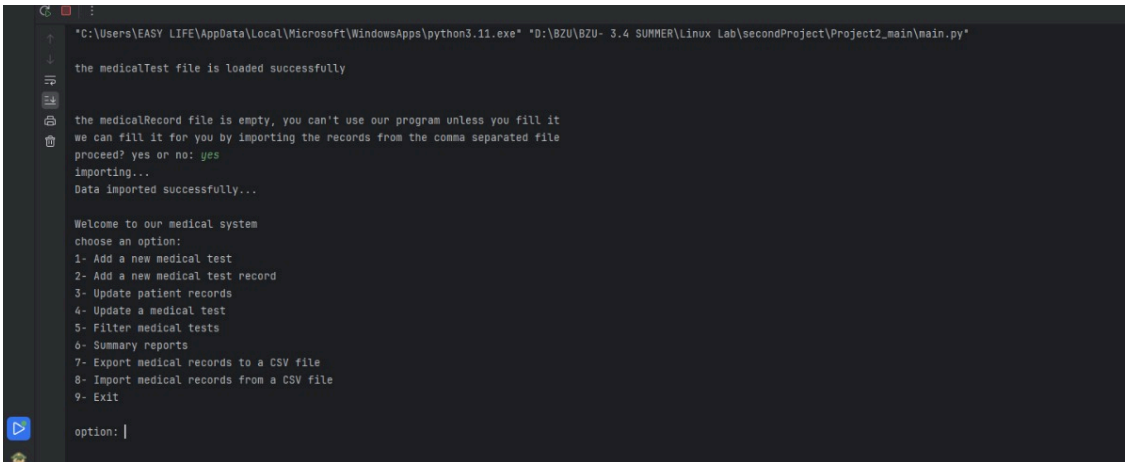


As viewed in the previous figure , the non-existence of the file was detected , thus the user is given the choice to create a new file for each, since the files empty the user is required to fill the files with data then use the program, the user is given the choice to import the data from the .csv file to fill the file.

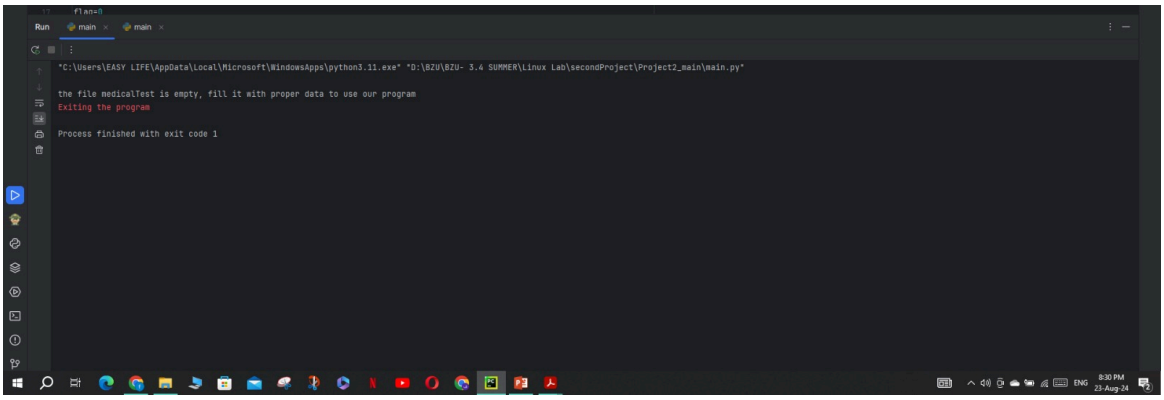
1.4. Empty files:

When performing any operation on the record file, both the medical record and medical test files should be filled with data, thus their emptiness is checked before starting the program ,the following images represents **the handling of the error of the empty files**:

- **An empty medical record file :**



- **An empty medical Test file :**

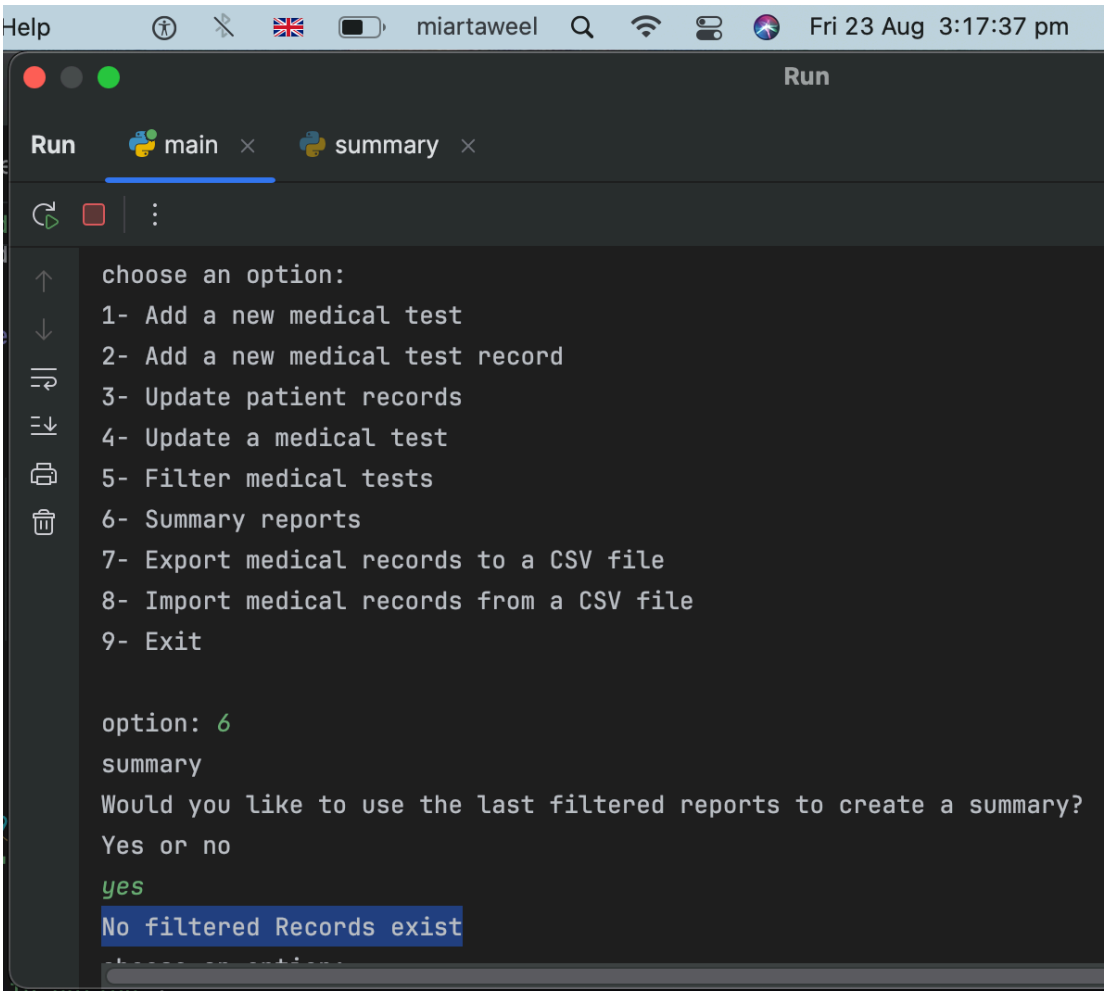


As viewed in the previous figures, the emptiness of both files in both cases was detected , thus the files are empty the user is required to fill the files with data then use the program, where in the case of the record file , the user is given the choice to import the data from the .csv file to fill the file.

1.5.Empty filtering file:

When performing the filter operation on the record file, such as the summary for instance, the file with the filtered values should be full should be filled with data , thus their emptiness is checked before starting the program ,the following images represents **the handling of the error of the empty filtering file**:

- **An empty filtering record file :**

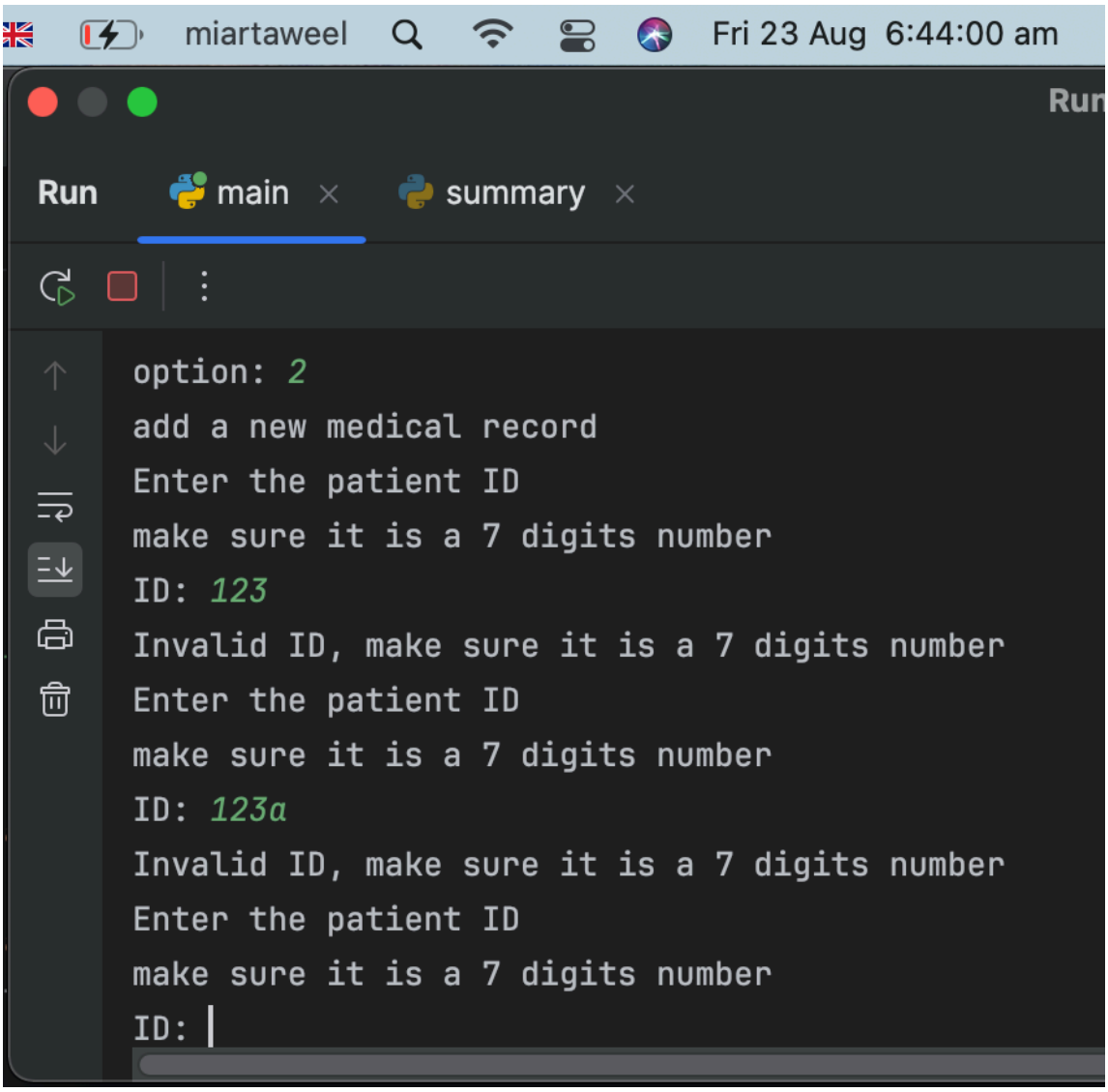


As viewed in the previous figures, the non-existence of filtered records was detected , thus the user is required to filter again by another attribute , then create another summary.

2. Data Validation:

2.1.Patient ID:

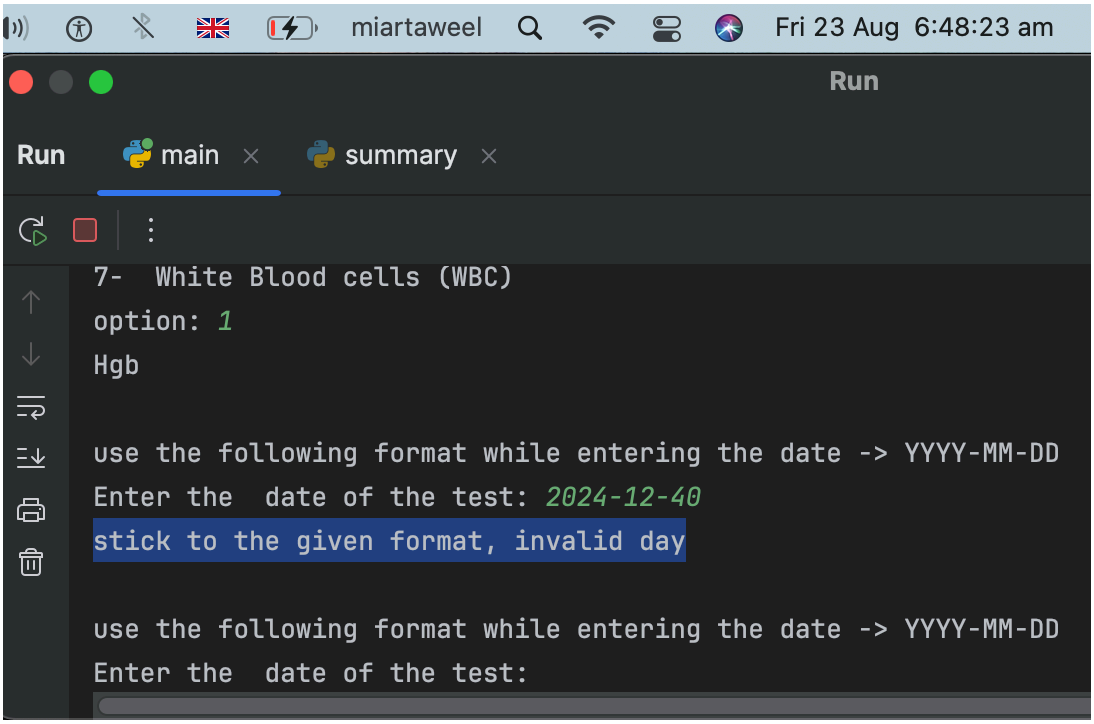
When performing a specific operation on the record file , the patient ID is inserted to determine which record to operate on ,thus the patient ID should be checked whether it's a valid 7 value number, the following image represents the checking of its validity :



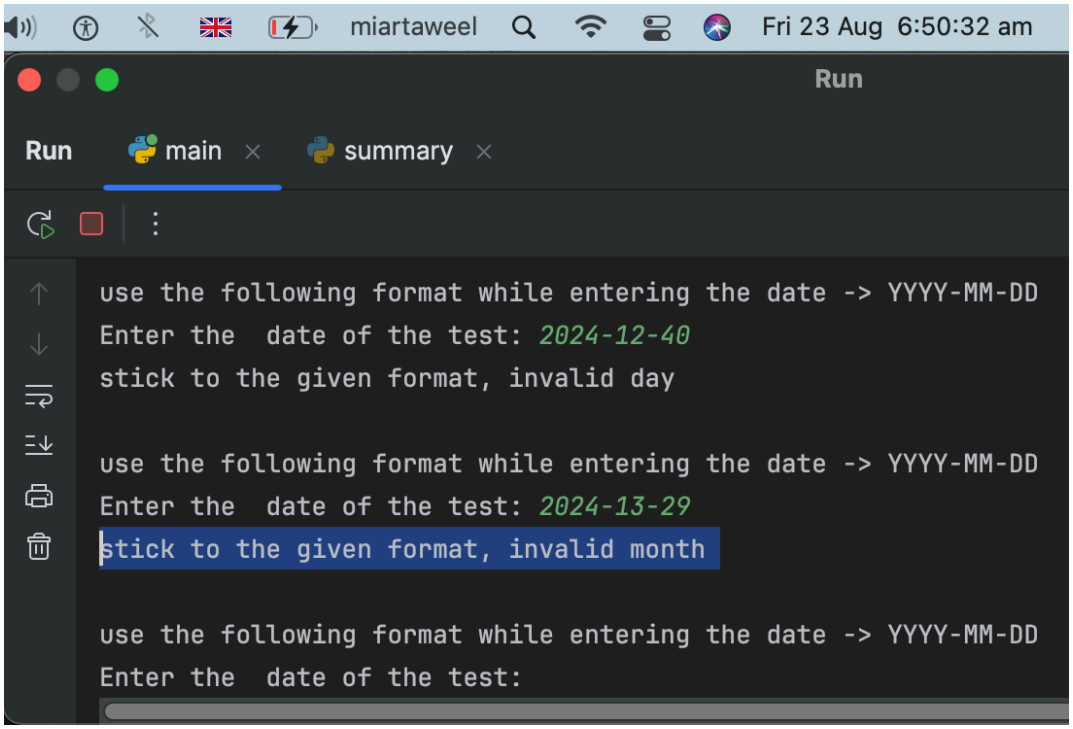
2.2.Date Checking:

When inputting a date to insert into a new record or when updating a record , the date should be checked whether it's a valid date written in the YYYY-MM-DD format, the following image represents the checking of its validity :

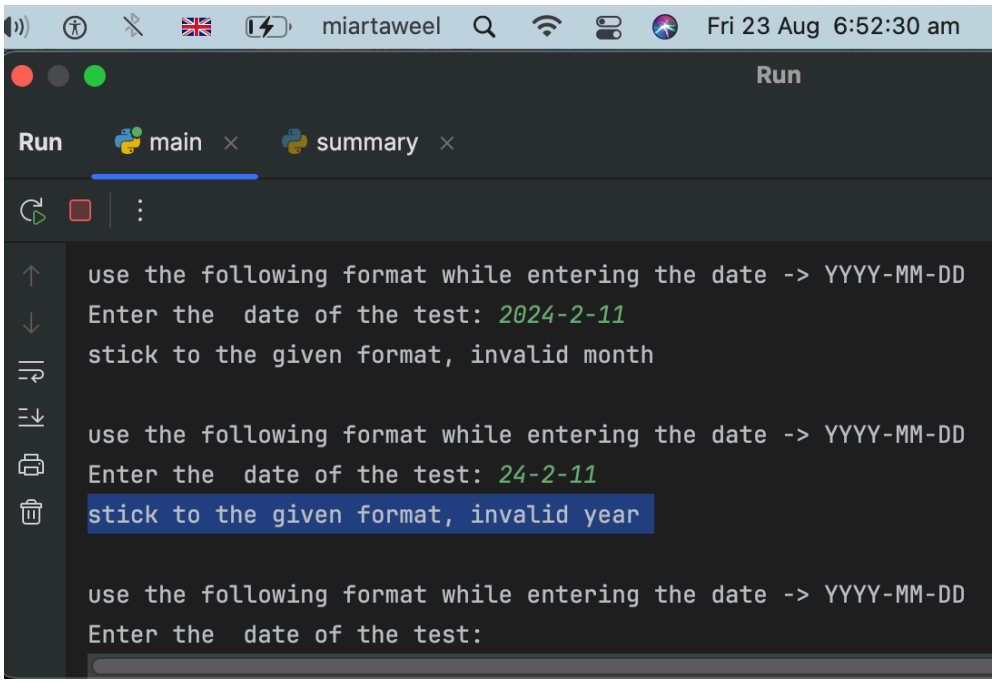
- **Day Validity :** An invalid day was inputted



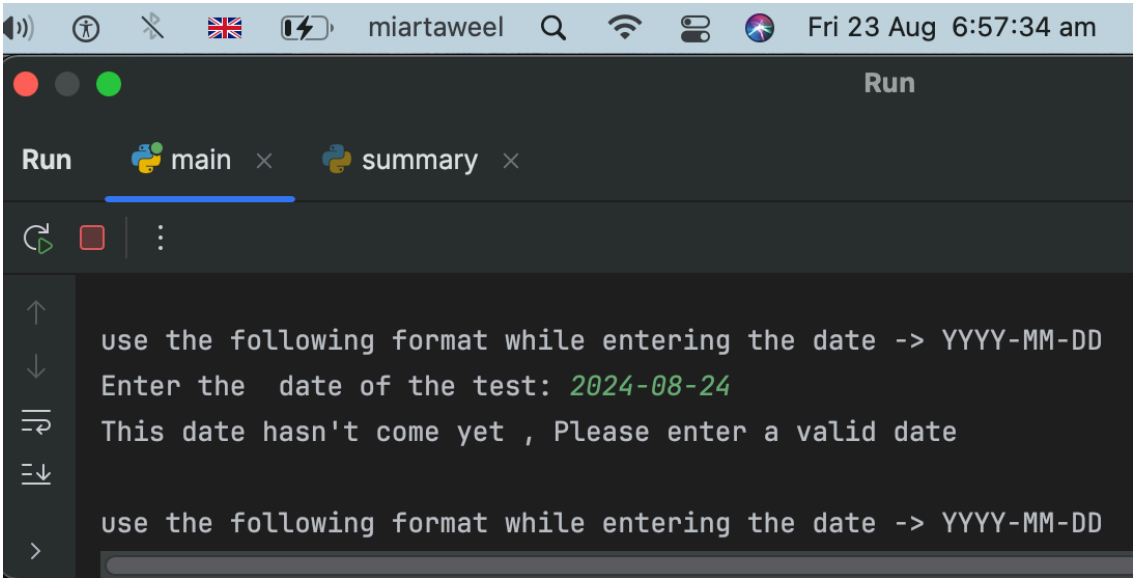
- **Month Validity :** An invalid month was inputted



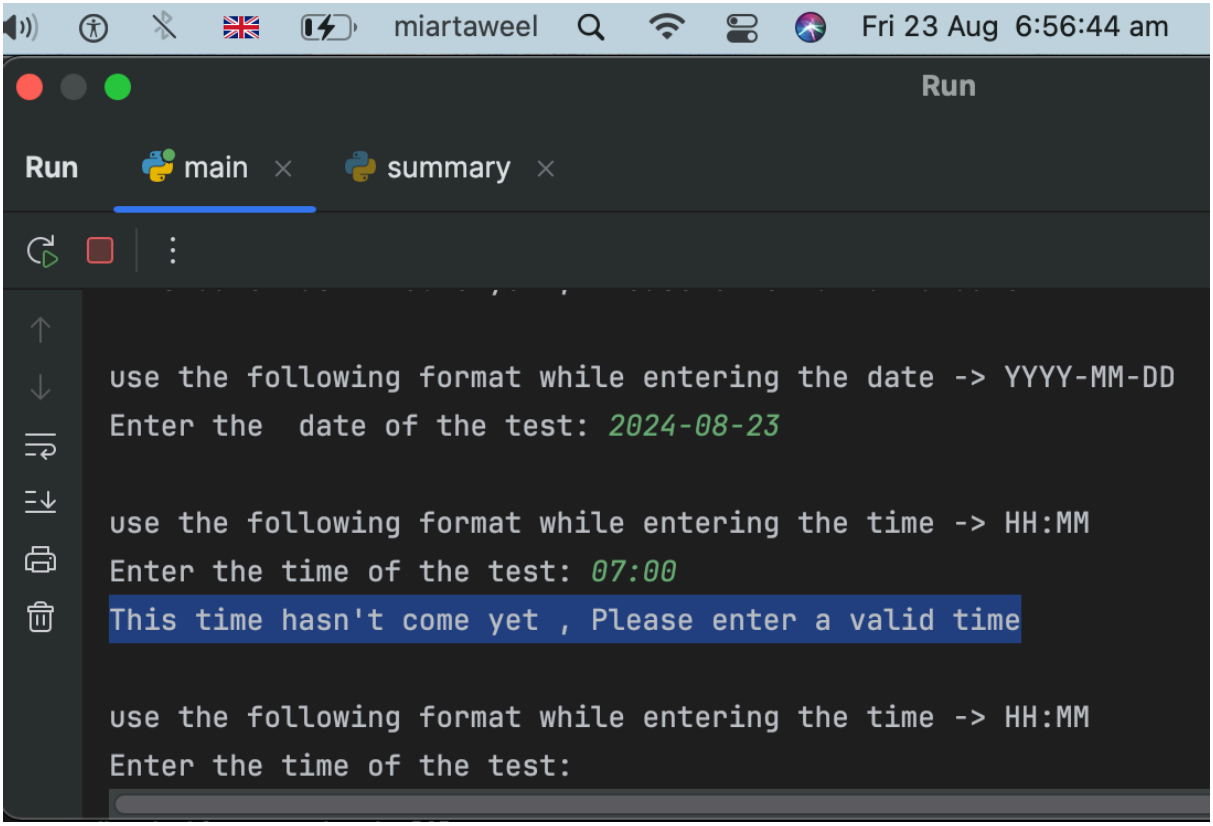
- **Date Format Validity :** An invalid date format was inputted



- **checking for invalid future dates :** An invalid date which hasn't come yet was inputted



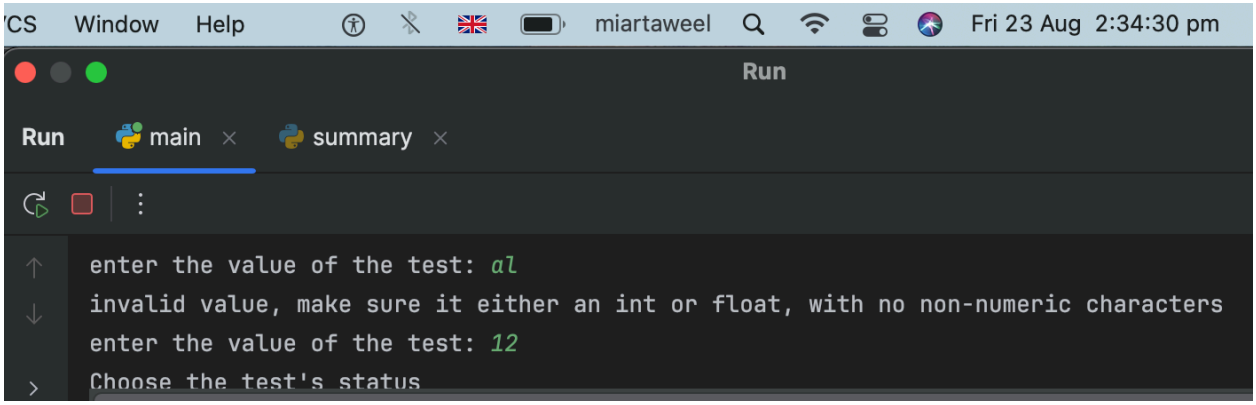
As viewed in the image , an invalid date from the future value was entered , since the date today is 2024-08-23 , and the inputted date was 2024-08-24 , which is tomorrows date as in it didn't come yet thus it was rejected and required the user to enter a valid value



As also viewed in the image , a date and an invalid time from the future value was entered , since the date today is 2024-08-23 , the inputted date was 2024-08-23, which is a valid date, but the time is invalid since the current time was 6:56 am and the inputted time is 7am as in it didn’t come yet thus it was rejected and required the user to enter a valid value

2.3.Test Value Checking:

When inputting a value to insert into a new record or when updating a record , the value should be checked whether it's a valid number, the following image represents the checking of its validity :

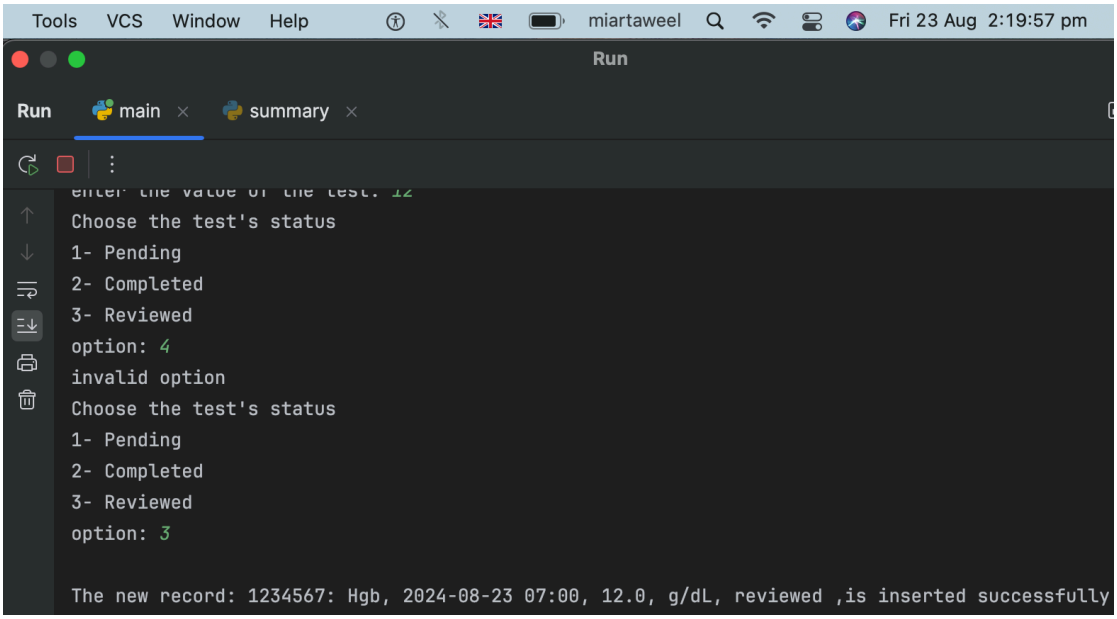


As viewed in the image , an invalid value was entered , thus it was rejected and required the user to enter a valid value, an integer or a float.

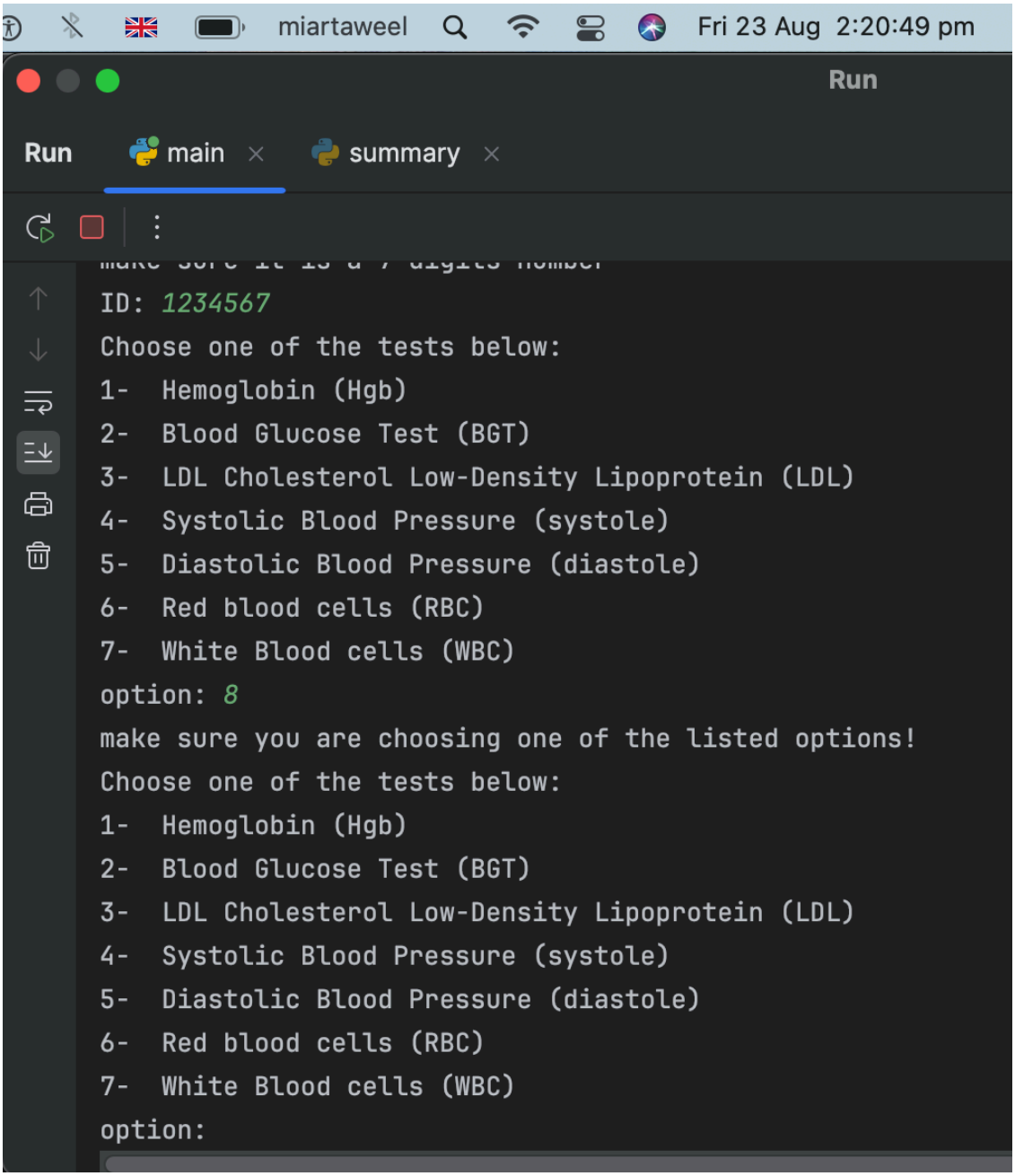
2.4. Test Type and status checking:

When inputting the type and status od the test to insert into a new record or when updating a record , the date should be checked whether it's a valid type and status chosen from the existing types in the medical test files , thus for each of the tyoe and status a menu is printed to chose a valid existing value from it , as shown in the following images :

- **Test Status:**



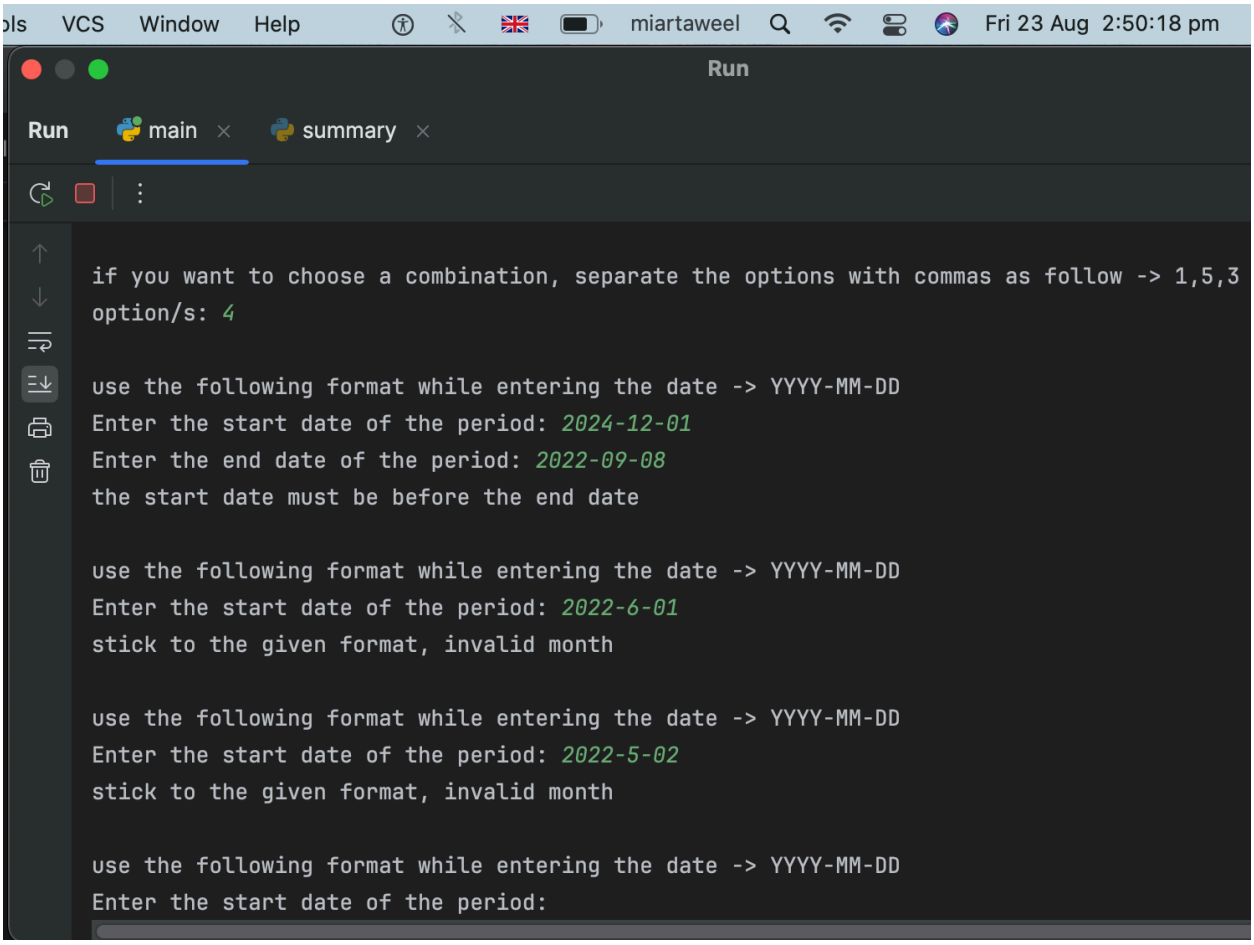
- **Test Type:**



As viewed in the previous figures , it is not allowed to choose a value different from the one in the drop down menu since it will only demand the user to choose from there, which assures for both cases that a valid test status and tyoe will be chosen accurately .

2.5.Valid Date Range Checking :

When inputting a specific range of date to retrieve a number of records from , the range should be checked whether it's a valid range , as in the first date should be a date before the second date ,which is checked as shown in the following image:

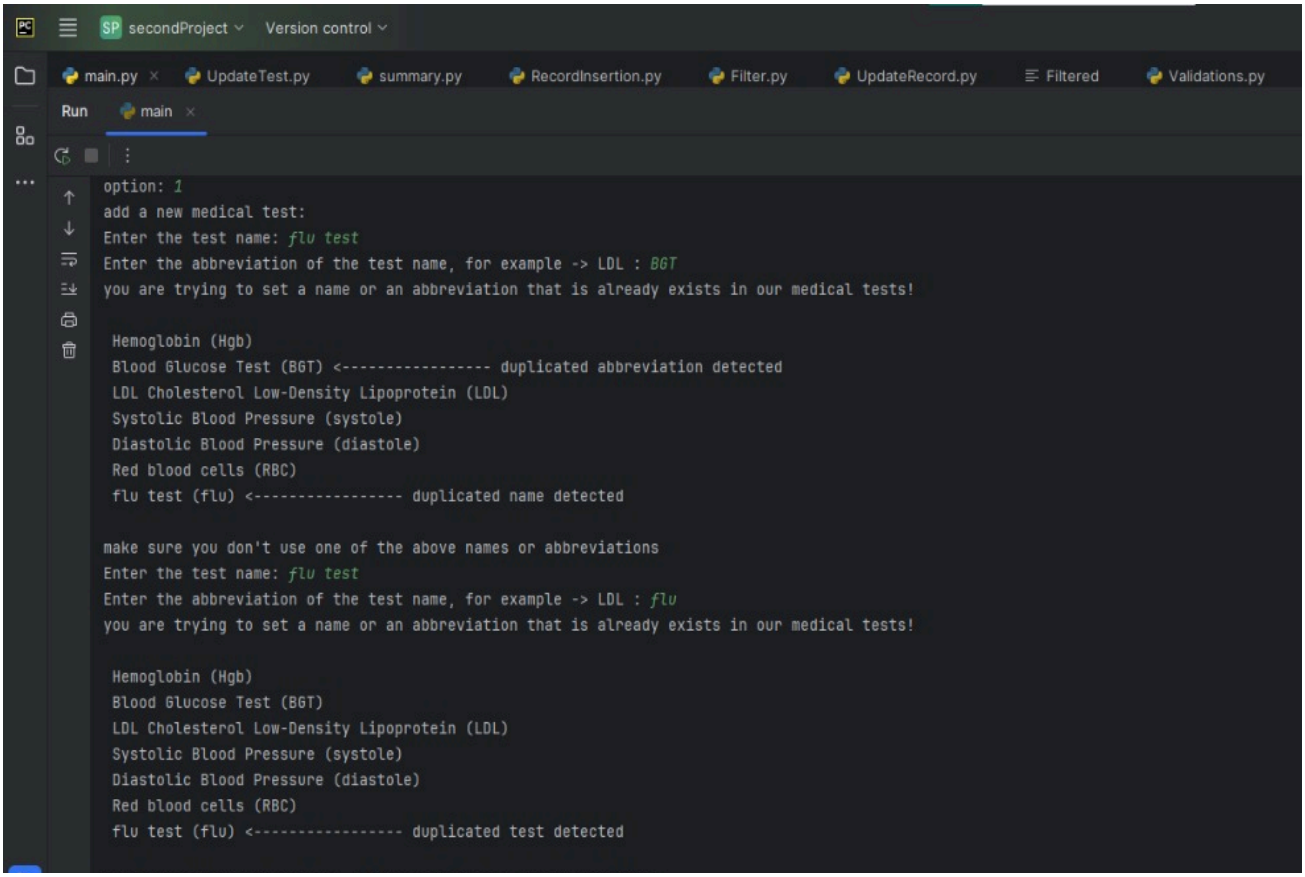


As viewed in the picture , an invalid range of dates was entered, in the first attempt to enter , the second dates' year was before the first one ,and in the second attempt to enter , the range the second dates' month was before the first one thus it was rejected and required the user to enter a valid range

2.6.Valid Test name and abbreviation:

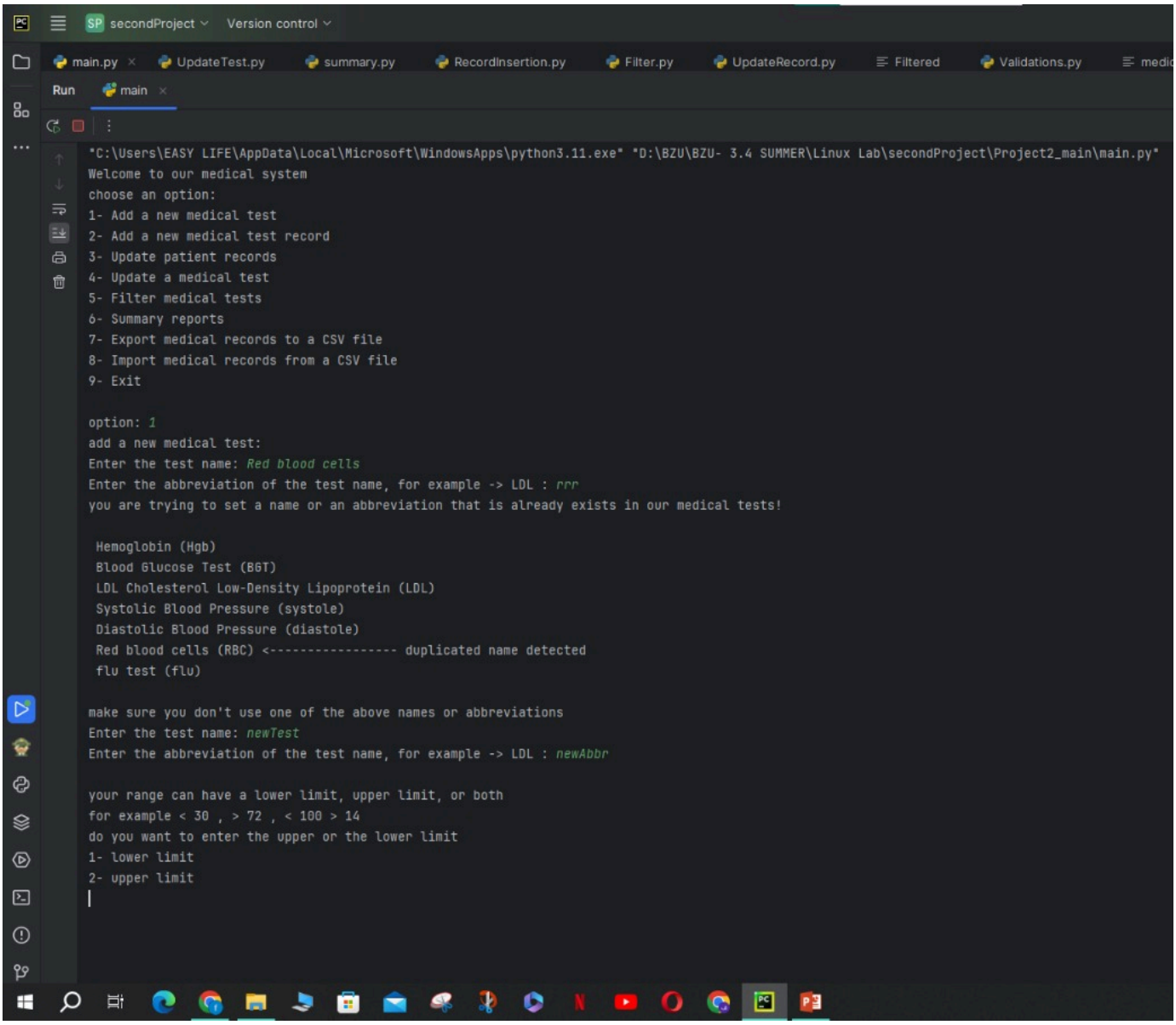
When inputting a specific test name to insert/update the medical test file , the name and abbreviation should be checked whether they're valid, as in it shouldn't already exist in the medical test file ,which is checked as shown in the following image:

- **Invalid Name and Invalid abbreviation:**



As viewed in the previous image, the entered name and abbreviation already existed in the medical test file in two different medical tests , in the second iteration both the name and abbreviation already existed in the medical test file in the same medical test ,thus it was required to be entered again.

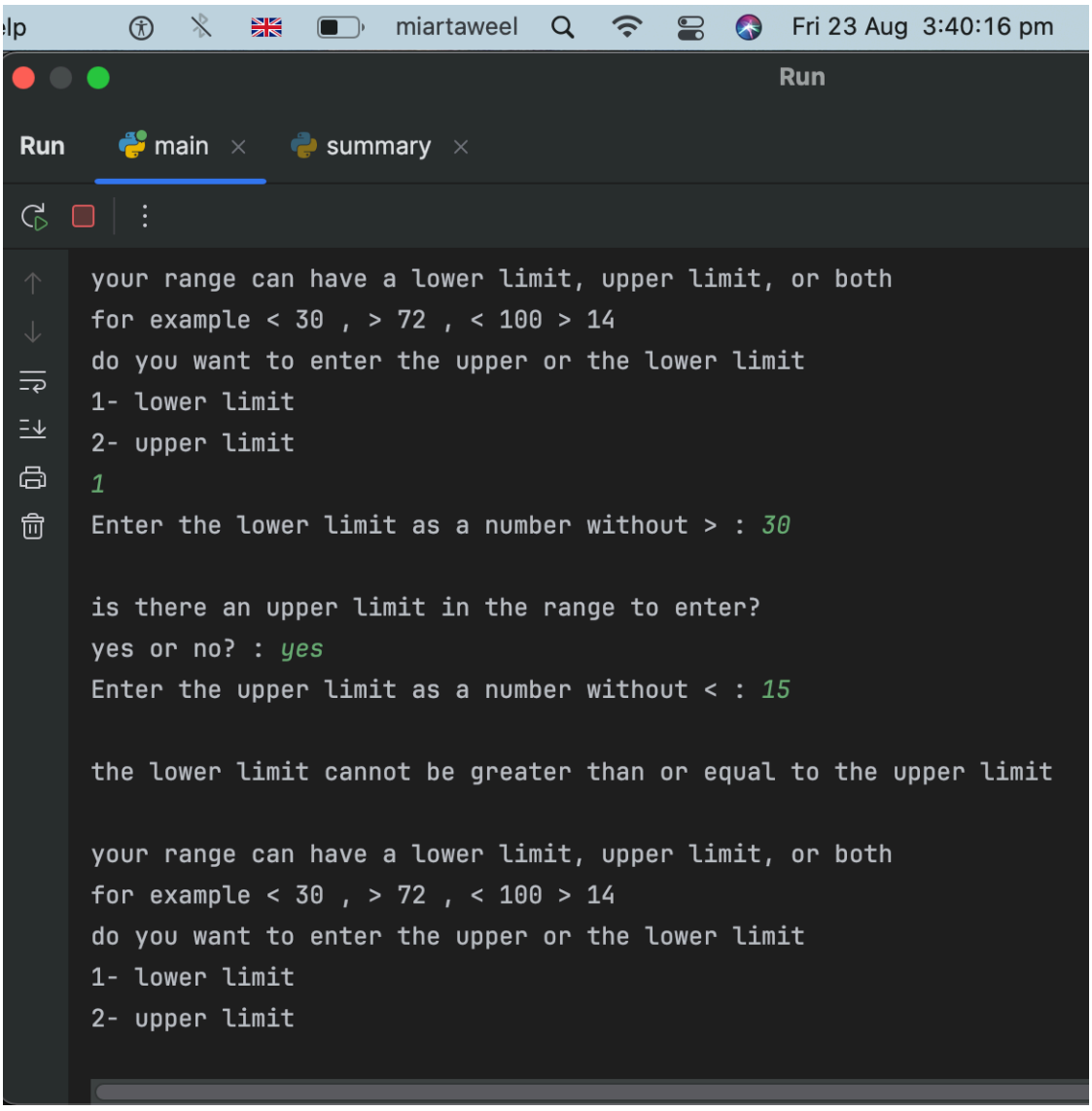
- **Invalid Name and valid abbreviation:**



As viewed in the previous image, the entered name only already existed in the medical test file and even though the abbreviation is valid the error was still detected ,thus it was required to be entered again. in the second iteration both the name and abbreviation were correctly entered thus it was terminated successfully.

2.7.Valid Range Checking :

When inputting a specific range of values **to insert into a medical test** , the range should be checked whether it's a valid range **, as in the lower limit should be in fact less than the upper limit** ,which is checked as shown in the following image:

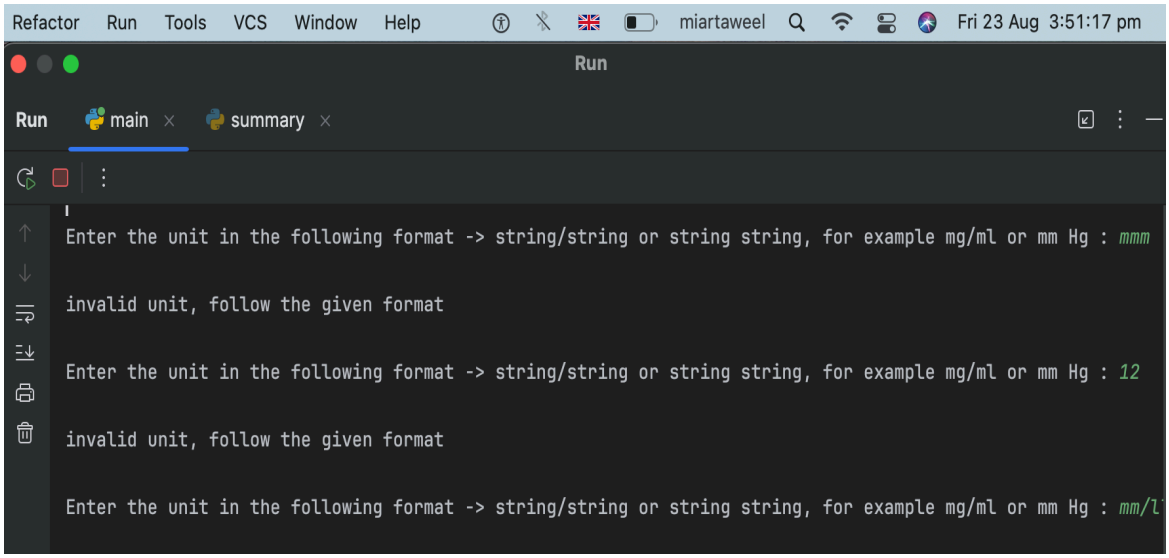


As viewed in the picture , an invalid range was entered, the lower limit entered (30) was not less than the upper limit (15), thus the user was required to re-enter a valid value range.

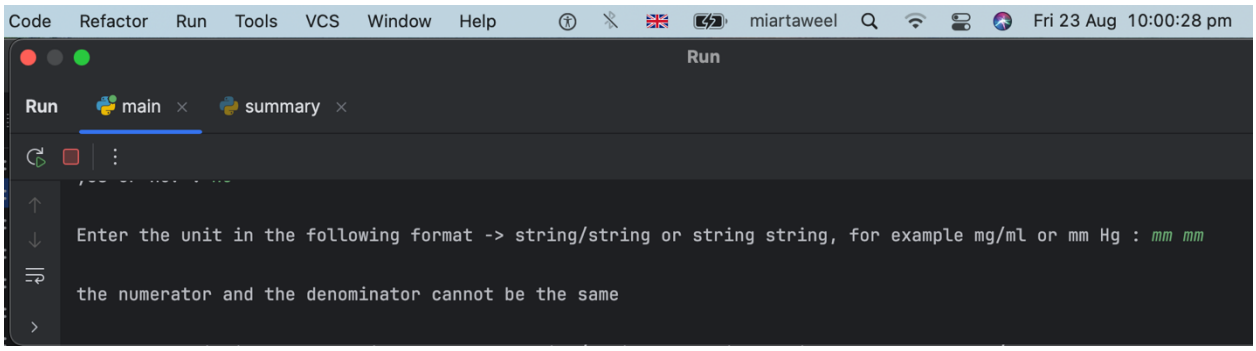
2.8.Unit Checking:

When inputting a unit to insert into a new medical test or when updating a test, the unit should be checked whether it's a valid unit written in the "char/char" or "char char" format, the following image represents the checking of its validity :

- **Invalid format:**



- **The Numerator is equal to the Denominator:**

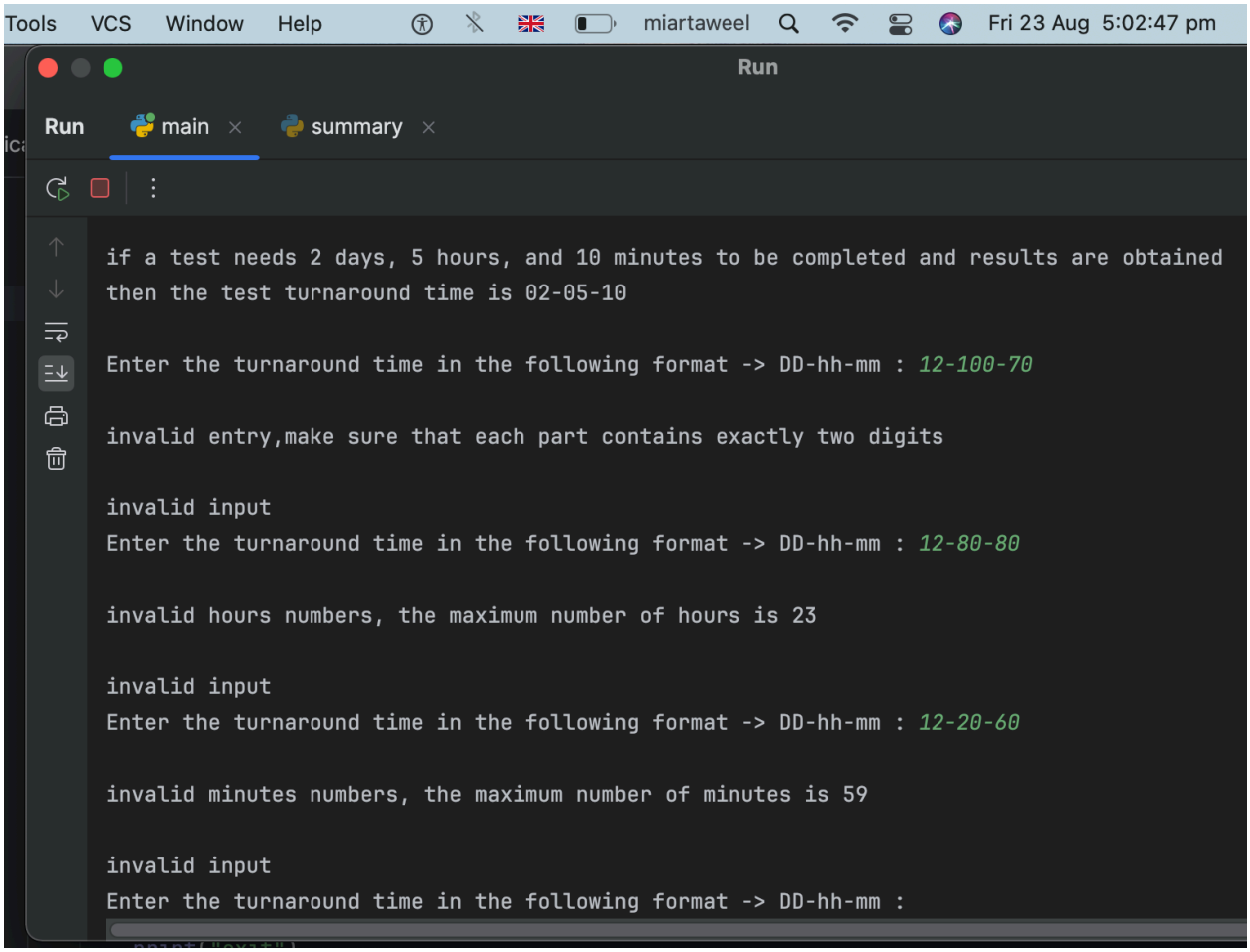


As viewed in the previous figure, the first input entered wasn't inserted in the correct format , ,the user entered another input which was also invalid since in contained integers not chars, the third attempt the numerators value was equal to the denominator ,thus the user was requested to enter a valid unit again.

2.9. Turnaround Time Checking:

When inputting a turnaround time to insert into a new medical test or when updating a test, the turnaround time should be checked whether it's valid written in DD-HH-MM format,

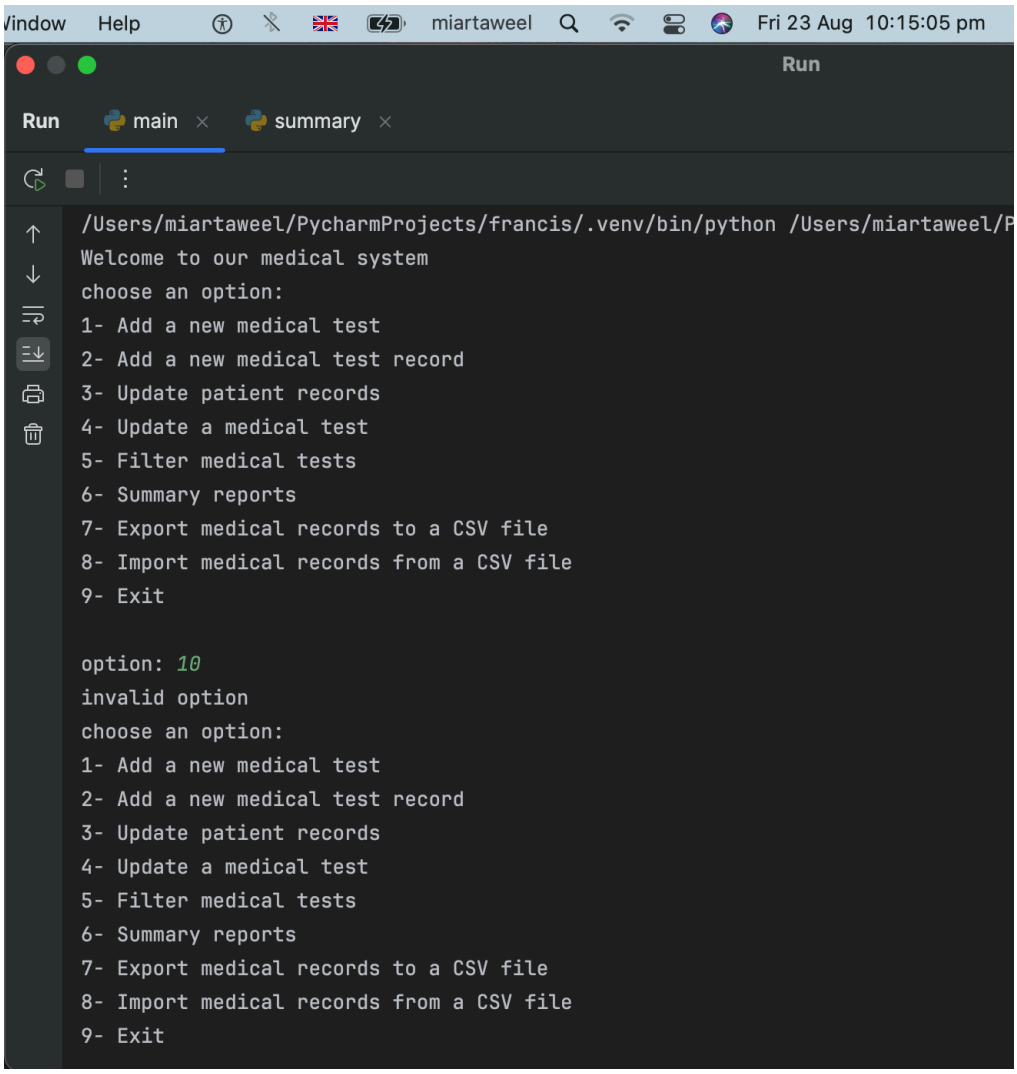
Where the hours aren't allowed to exceed 23 hours , and minutes aren't allowed to exceed 59 minutes , the following image represents the checking of its validity :



As viewed in the previous figure, the first input entered wasn't inserted in the correct format , thus the user entered another input which was also invalid since in contained hours that exceed 23, and in the third attempt minutes that exceed 59 were entered , thus the user was requested to enter a valid time again.

2.10. Invalid menu option :

When inputting an option to choose from the menu to perform any kind of operation , the option should be checked if it exists in the menu or not , the following image represents the checking of its validity :



As viewed in the previous figure, the index of the option chosen doesn't exist in the list , thus the user is required to re-enter the index so no error would occur.

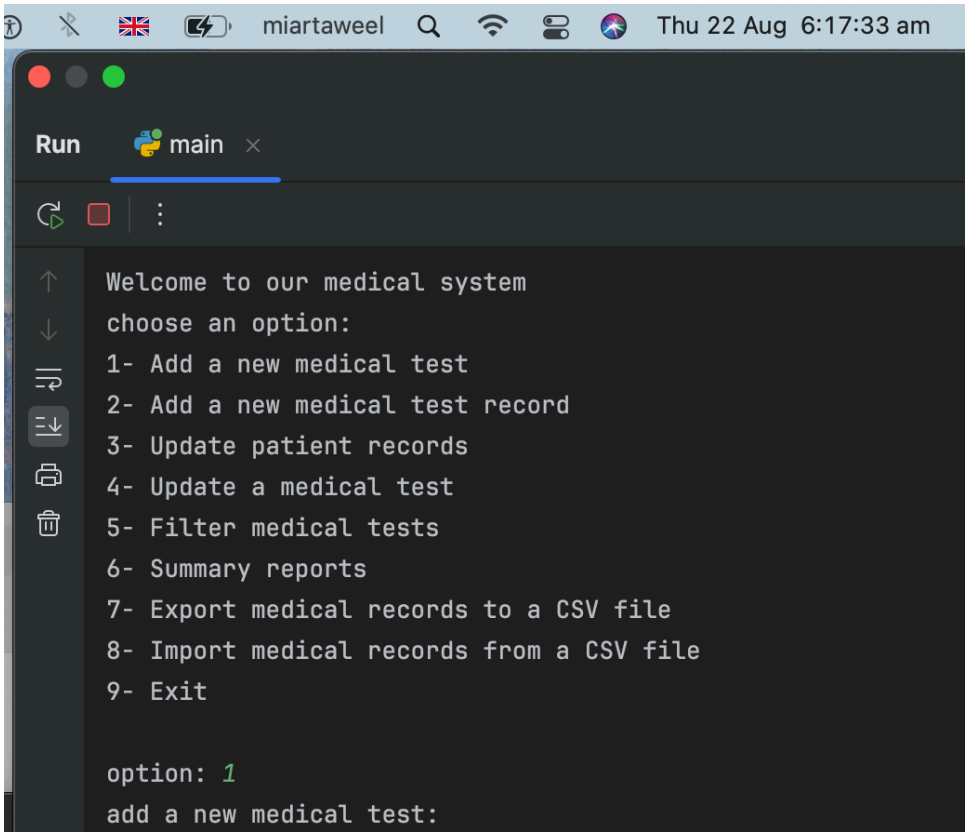
Note: This Validation check occurs in every menu existing in this program.

3. Execution :

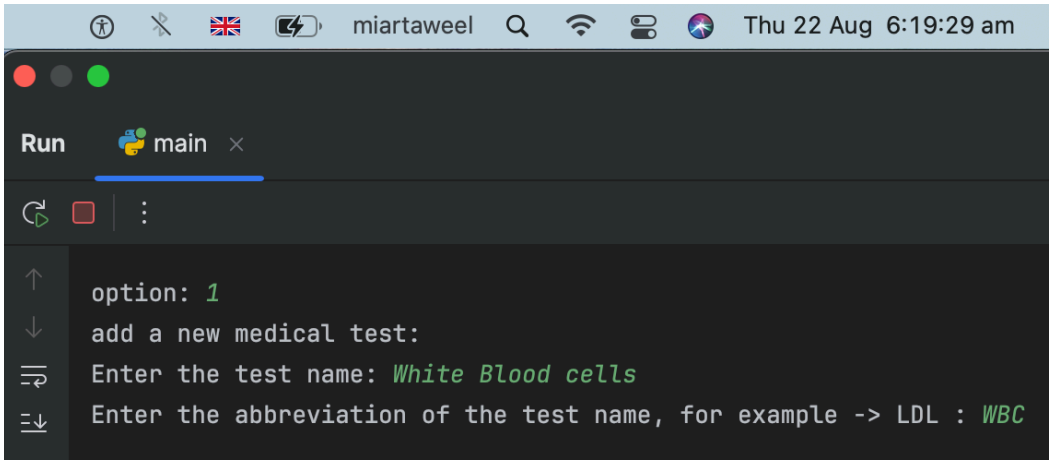
3.1.Inserting a medical test :

The following images represent inserting a new test into the medical test file with valid inputs , the output will be observed in the file after executing :

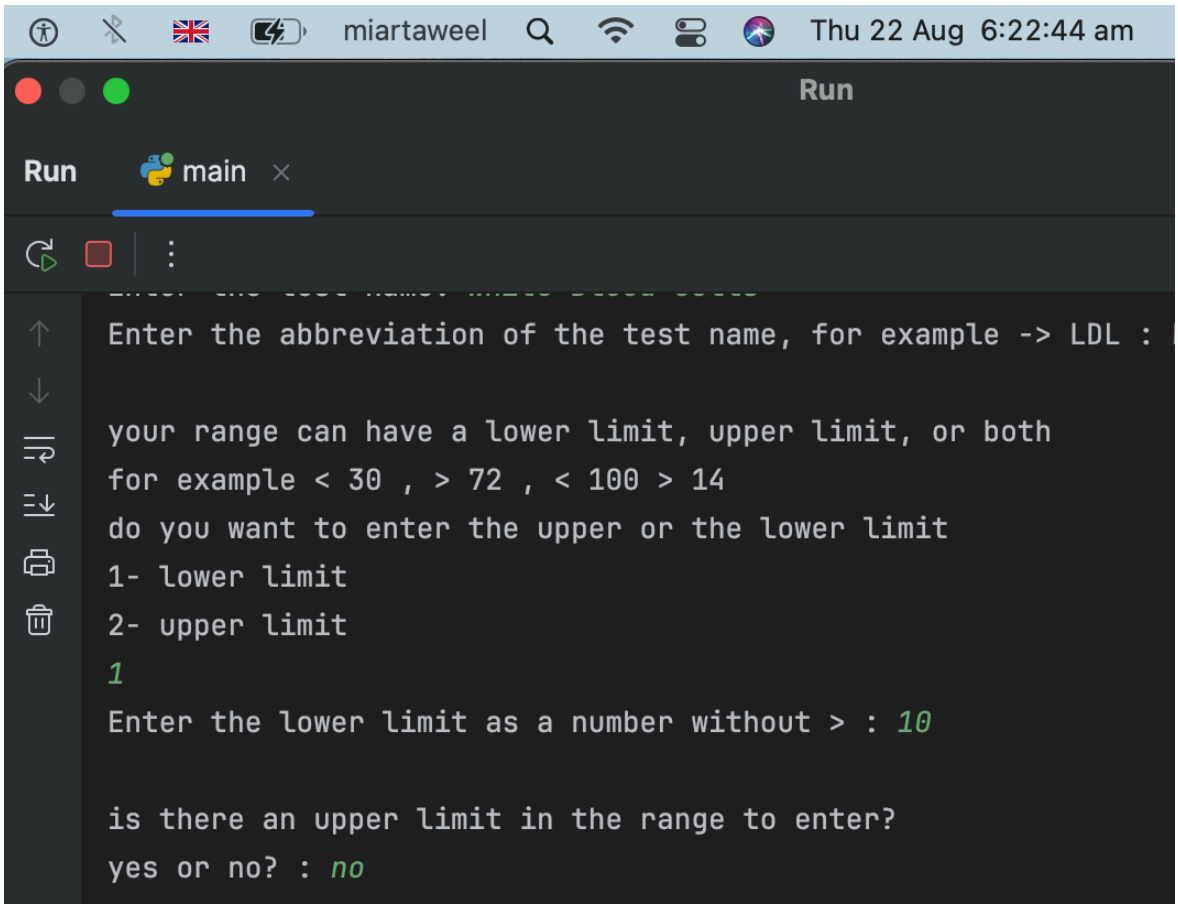
The operation starts by selecting the “Add a new medical test option” from the option menu:



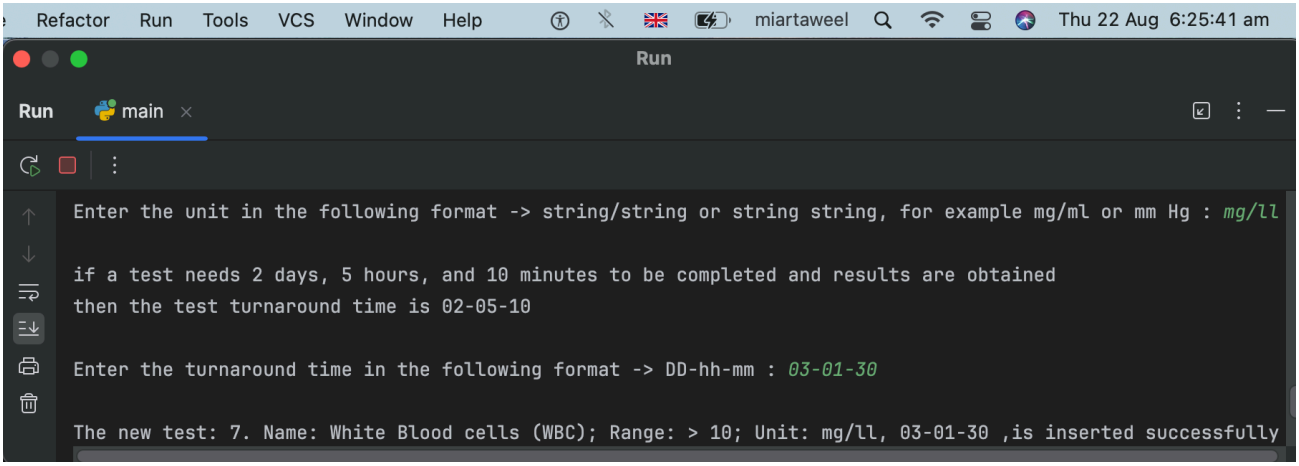
After Selecting the insert option from the option menu , a valid Test name and its abbreviation is entered:



Also valid values for the range which in this case is chosen to be a lower limit range , was inserted as viewed down below :



In addition to valid values for the unit, and turnaround times which were entered in their specified forma and inserted as viewed down below :

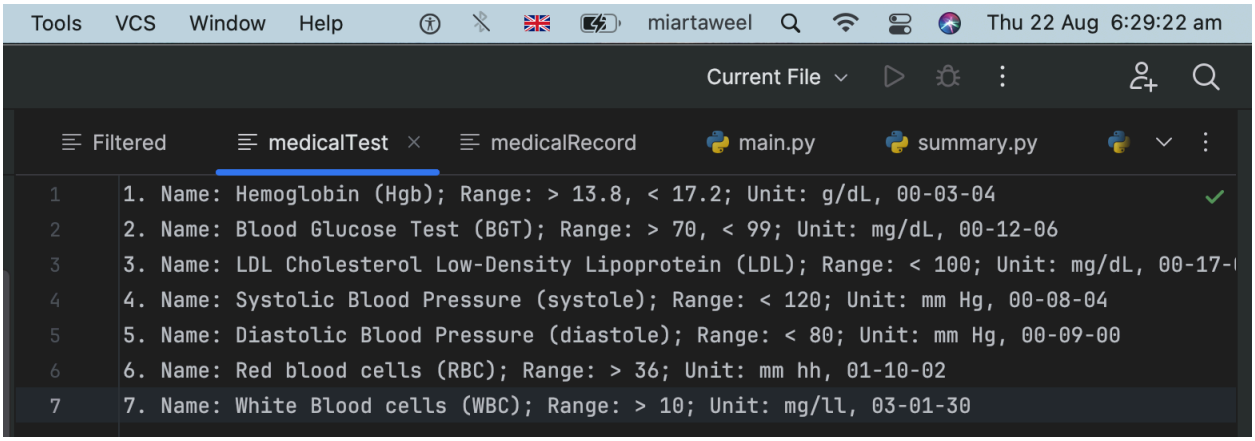


- **Output :**

According to the previous images' inputs , the following record should be inserted :

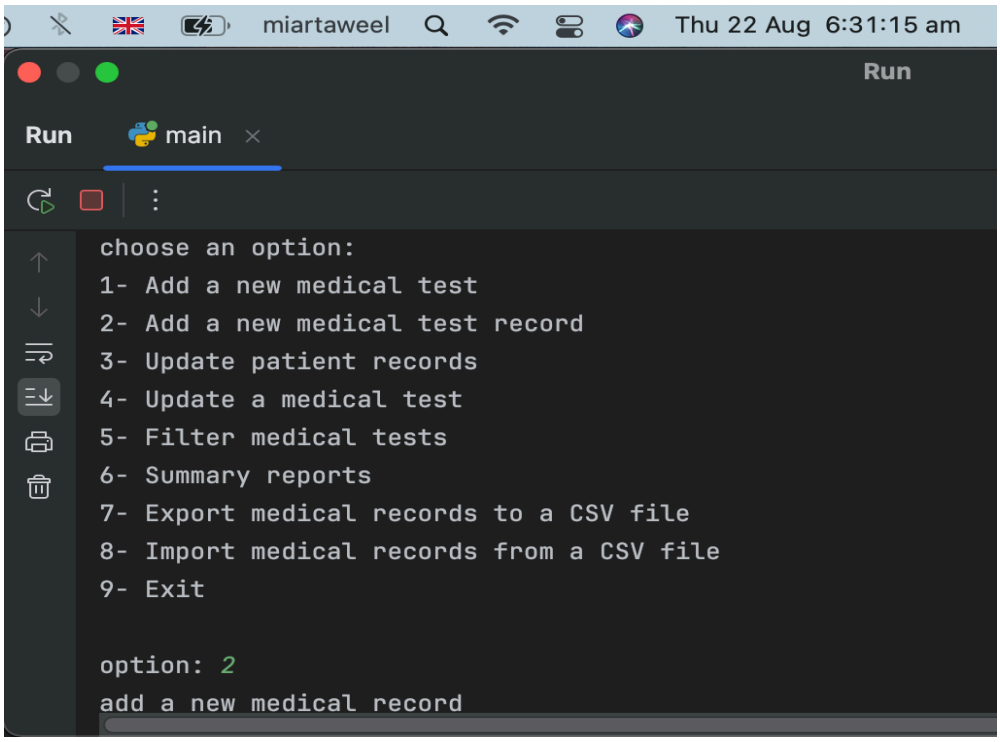
Name: White Blood cells (WBC); Range: > 10; Unit: mg/ll, 03-01-30

Which was in fact successfully inserted as viewed down below in the medical test file::

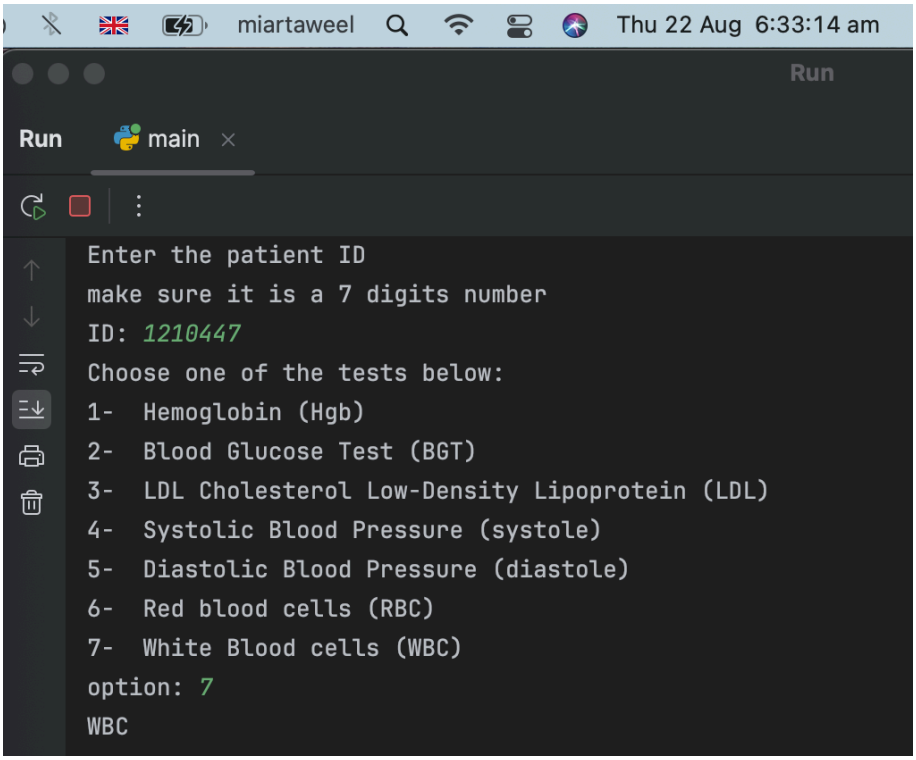


3.2.Inserting a medical record :

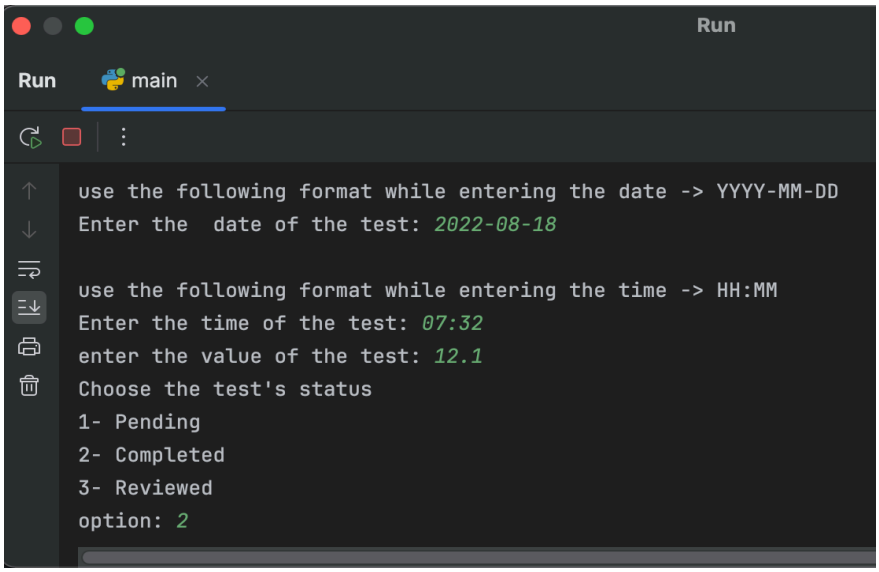
The following images represent inserting a new record into the medical record file with valid inputs , the output will be observed in the file after executing :



After Selecting the insert option from the option menu , a valid 7 integer number is inserted as the patient ID, in addition to selecting a valid test type from the test type menu.



Also valid values for the date time ,test value and the status were inserted as viewed down below :



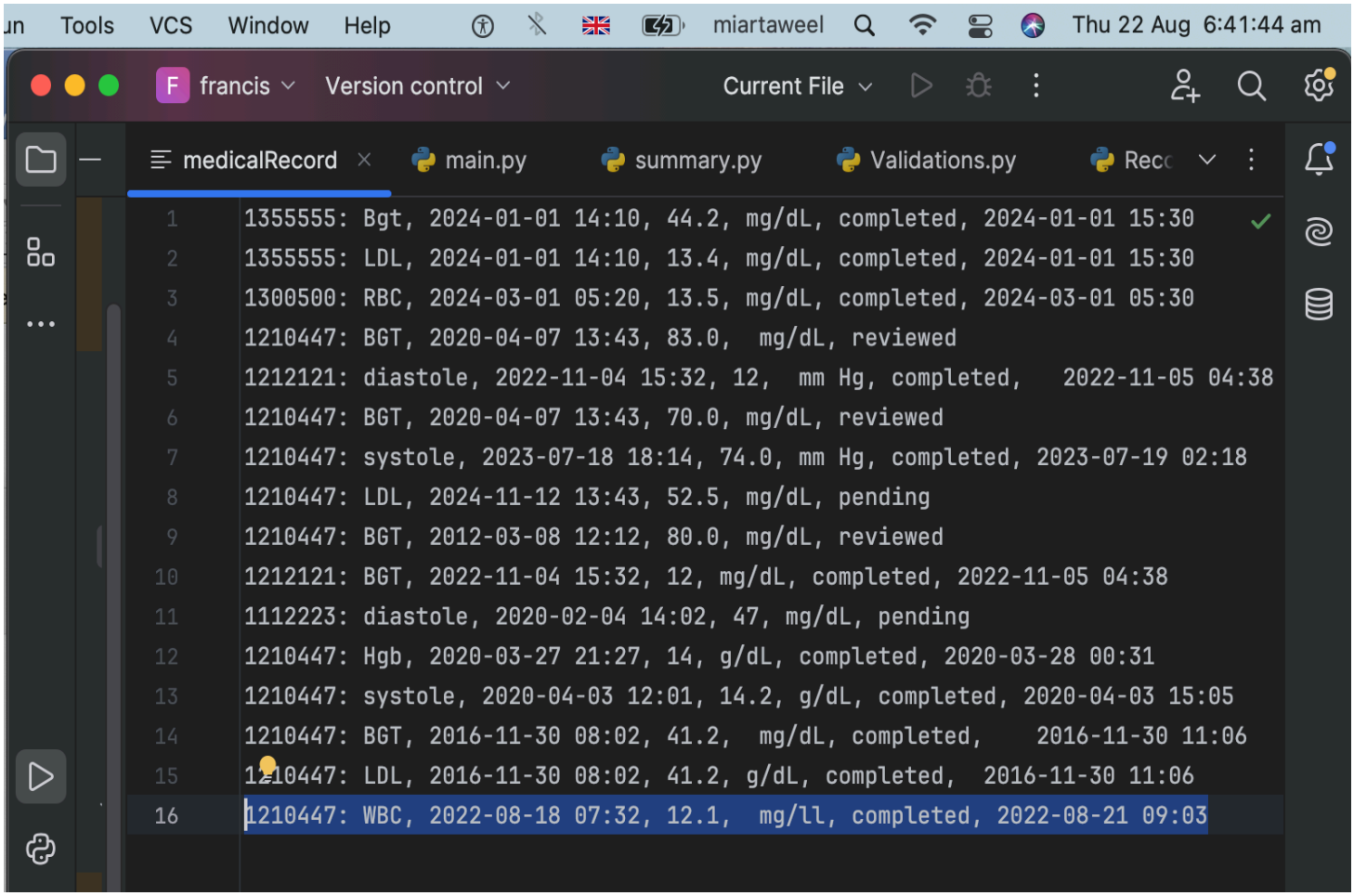
Note: The unit value isn't required to be inserted since its determined by the test type and is taken directly from the file .

- **Output :**

According to the previous images' inputs , the following record should be inserted :

1210447: WBC, 2022-08-18 07:32, 12.1, mg/ll, completed, 2022-08-21 09:03

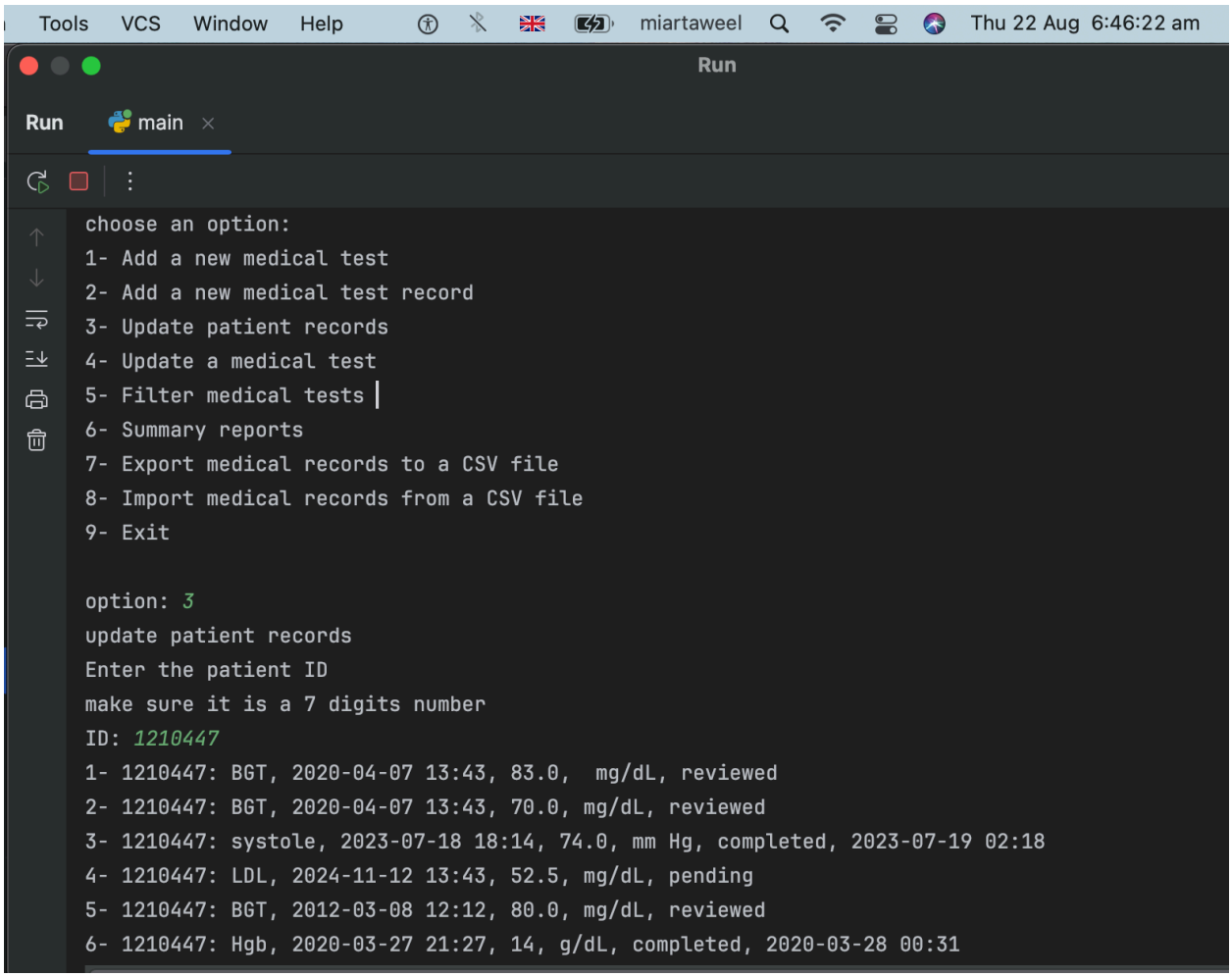
As viewed in the previous record the completion time was found by adding the turnaround time from the medical test file to the date entered currently, which was in fact successfully inserted as viewed down below :



3.3. Update Record:

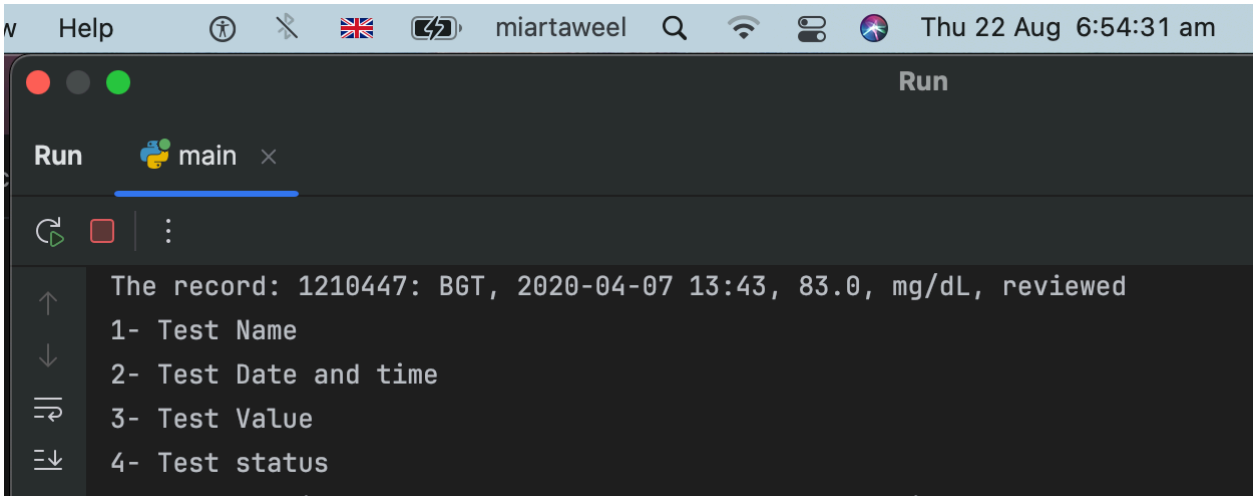
The following images represents updating all the fields of a specific test record based on choice as viewed down below :

The operation starts by selecting update option from the option menu, then inserting an existing patient ID to update its record :

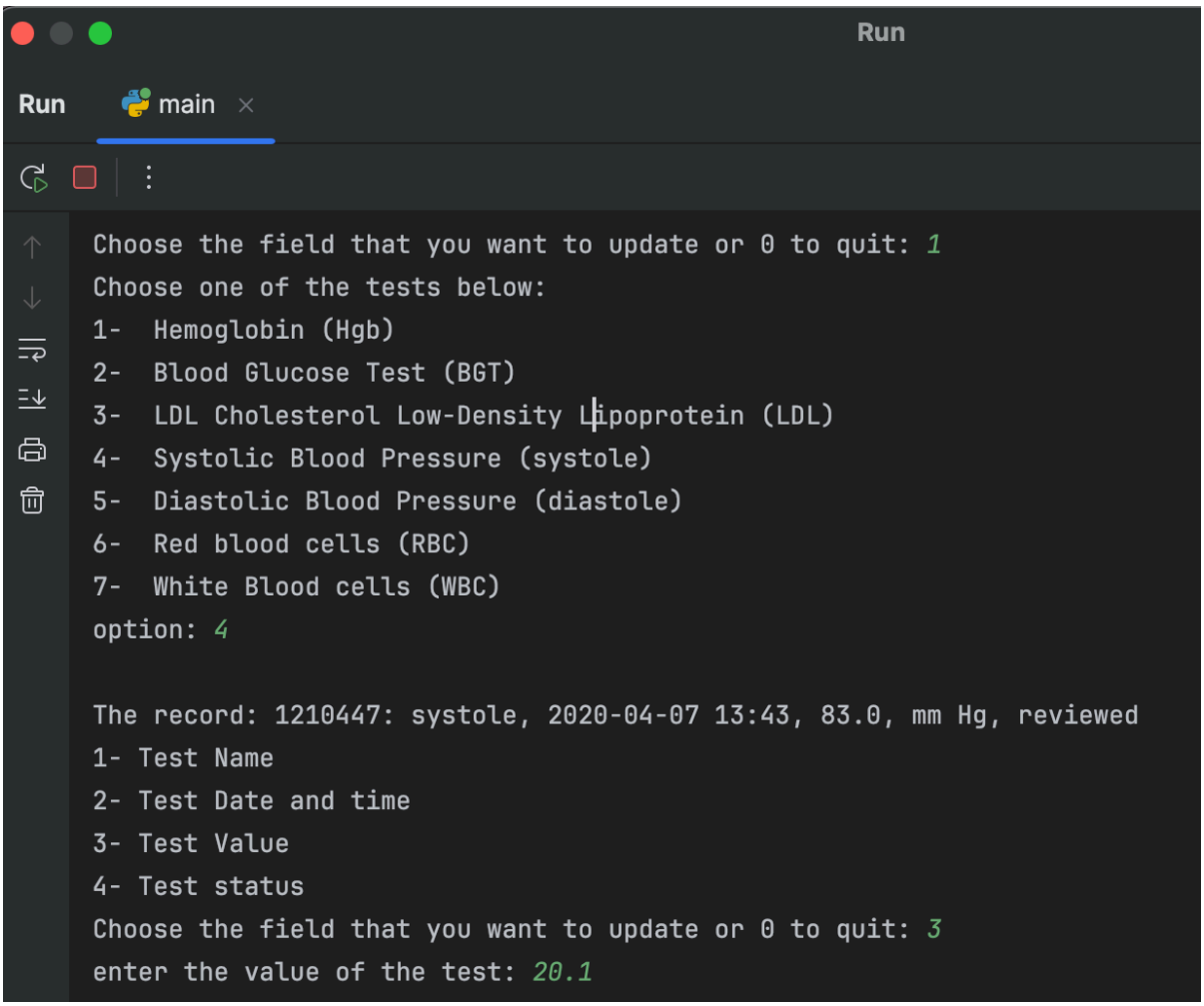


After inserting the patients ID , all the records of that patient is shown as viewed in the previous picture , thus one record must be chosen to update it .

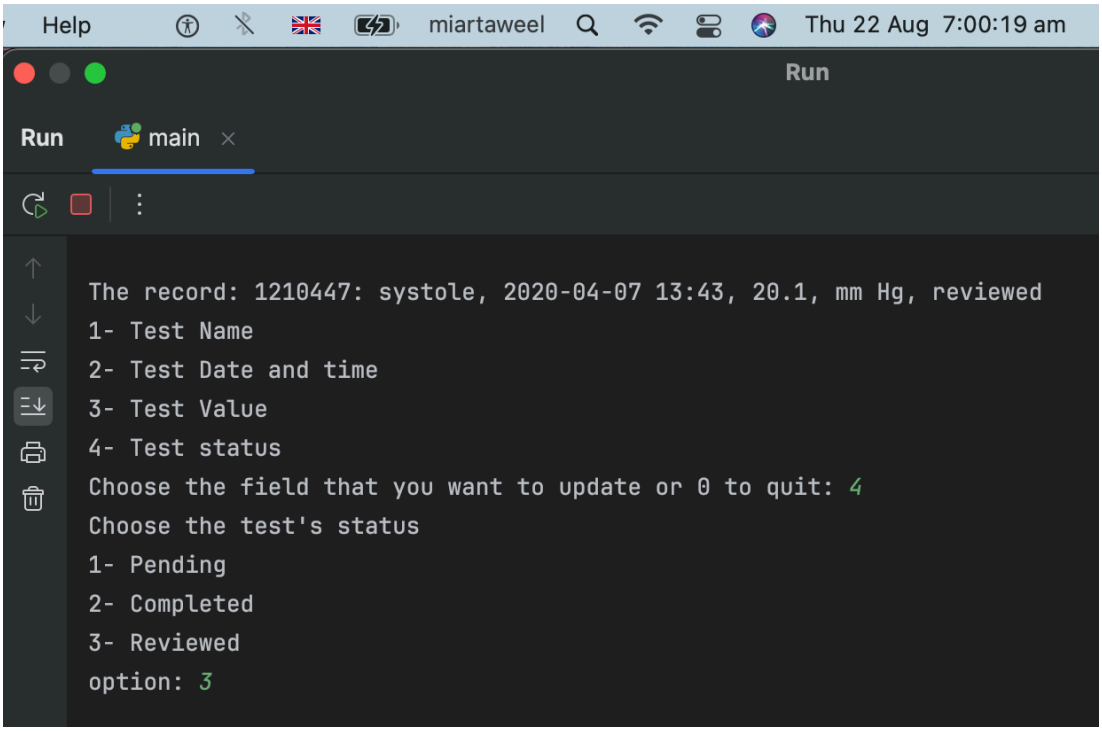
After a record is chosen ,a menu is shown to let the user choose which aspect they should update :



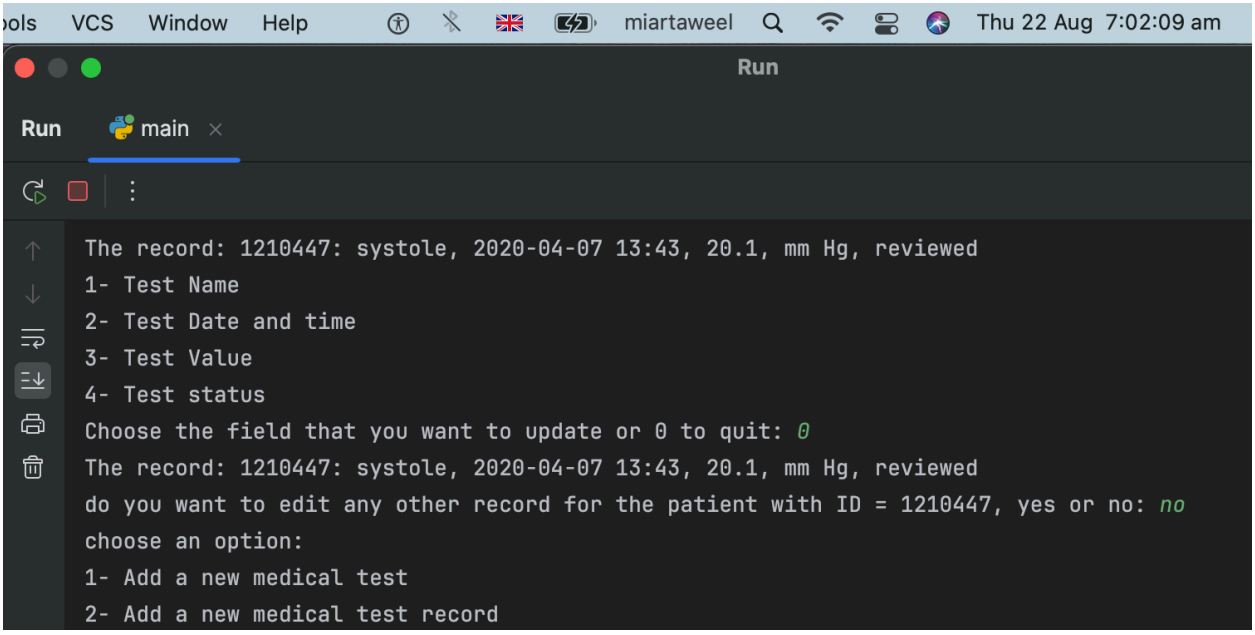
After choosing which aspect to update which in this case is the test type ,a new test type is chosen from its menu to update it , which then proceeds to replace the old value successfully as viewed down below:



After updating the test type , the update menu reappears,, then another aspect is chosen to be updated from :



After entering a new test status to update and executing it successfully , to exit and save the new record the “Quit” option is terminated by entering ‘0’, as viewed down below :



3.4. Update Medical Test:

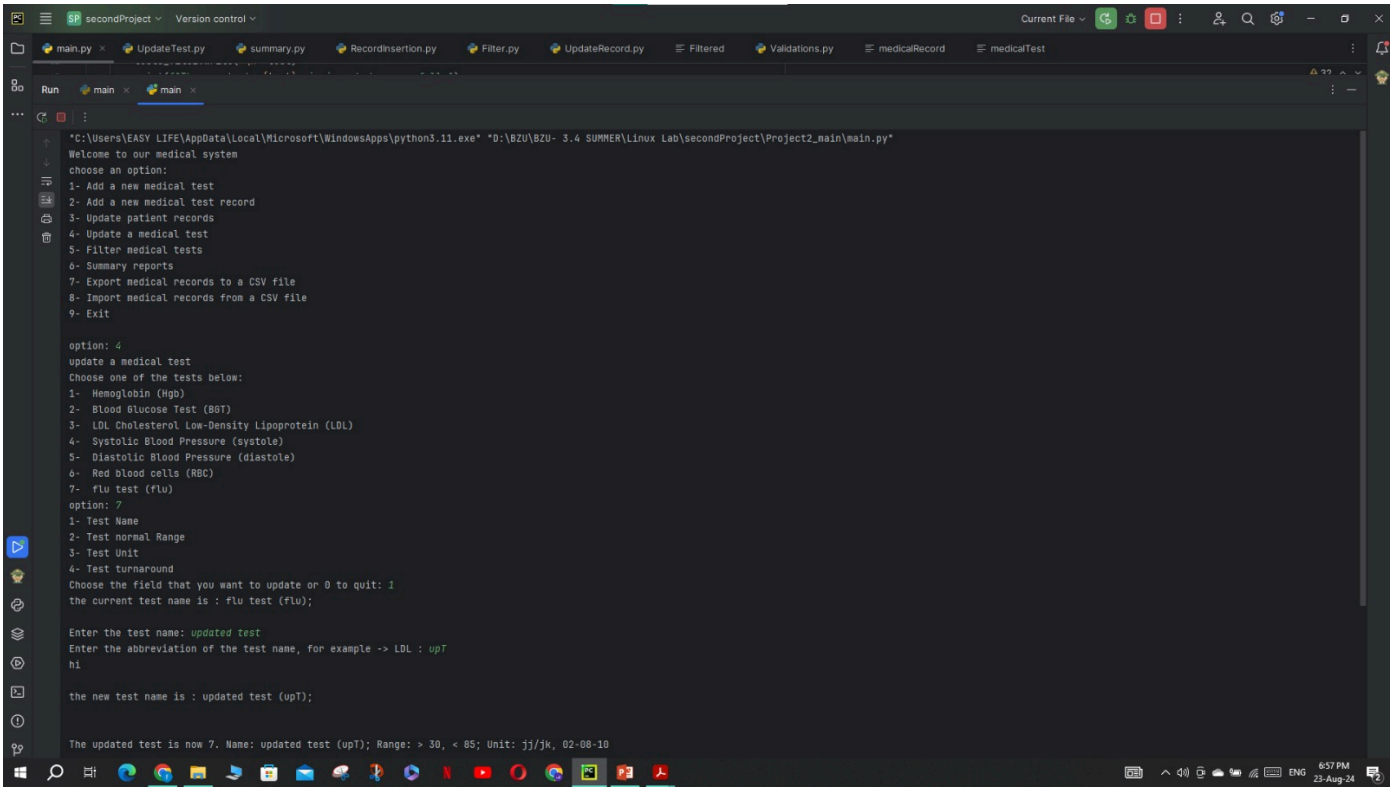
The following images represents updating all the fields of a specific medical test based on choice as viewed down below, where each update operation would also affect the data in the medical test files in fact :

The operation starts by selecting “update a medical test” option from the option menu:

The Old Record chosen to update:

Name: flu test (flu); Range: > 30, < 85; Unit: jj/jk, 02-08-10

- **Updating the test name :**

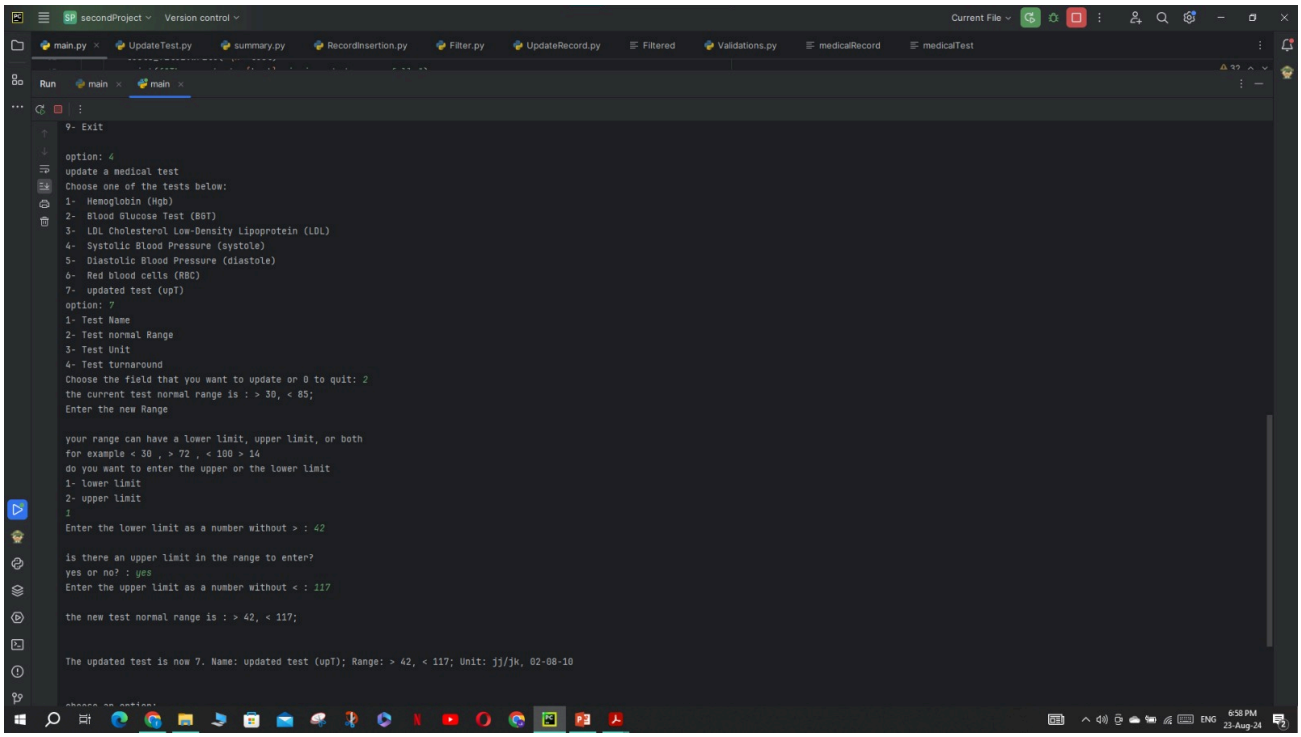


After updating the test name of that test, based on the inputted data viewed in the previous picture , the test became :

Name: updated test (upT); Range: > 30, < 85; Unit: jj/jk, 02-08-10

This change was also reflected onto the medical record file.

- **Updating the normal range:**

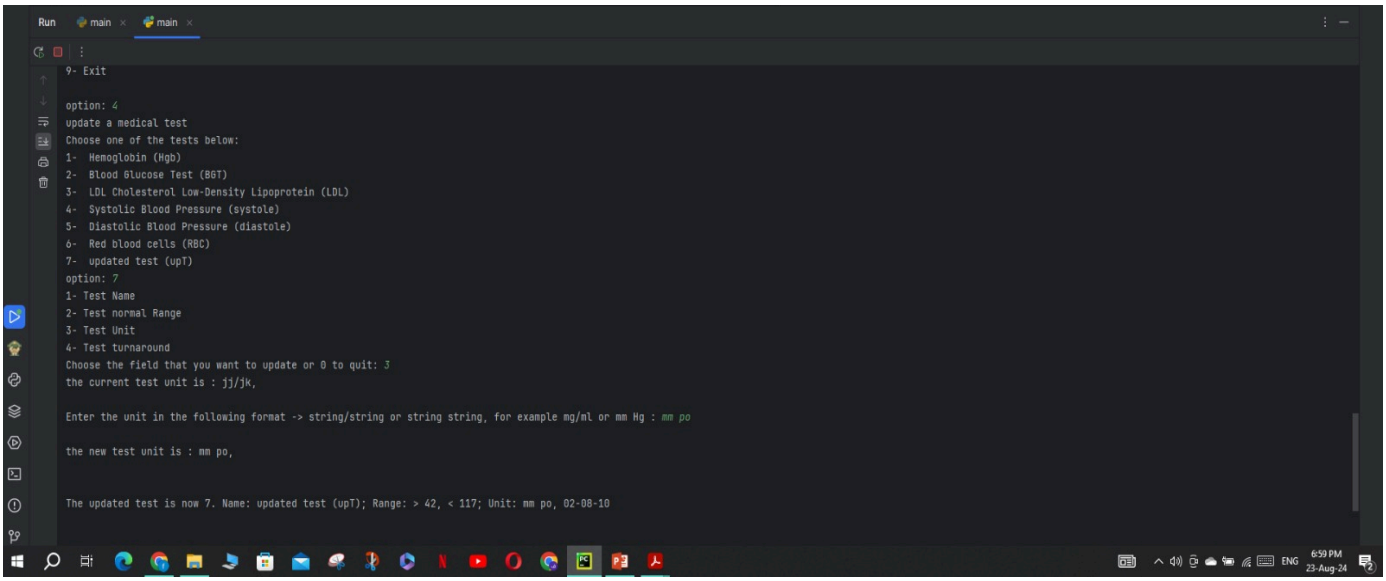


After updating the normal range of that same test, based on the inputted data viewed in the previous picture , the test became :

Name: updated test (upT); Range: > 42, <117; Unit: jj/jk, 02-08-10

This change was also reflected onto the medical record file.

- **Updating the unit:**

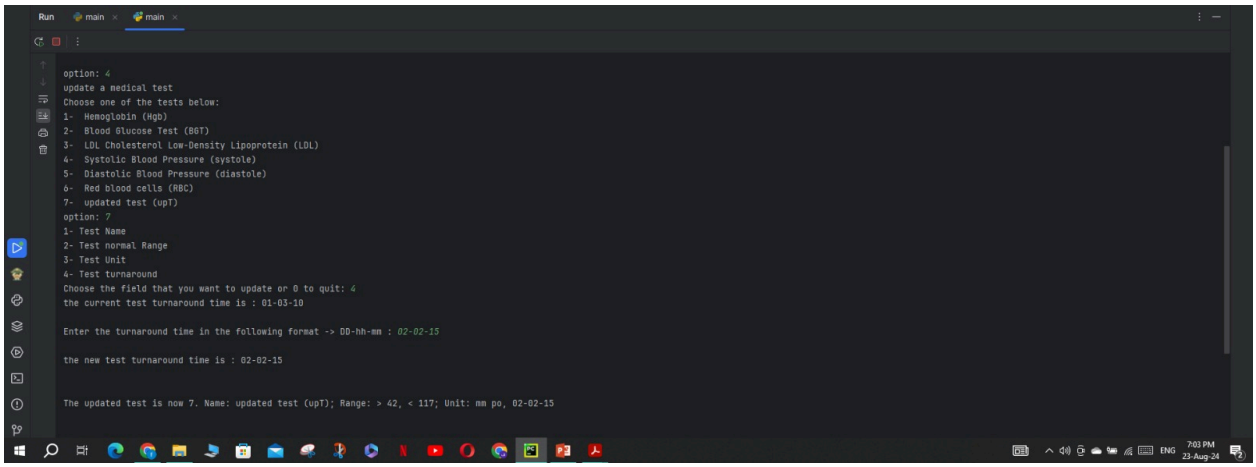


After updating the unit of that same test, based on the inputted data viewed in the previous picture , the test became :

Name: updated test (upT); Range: > 42, <117; Unit: mm po, 02-08-10

This change was also reflected onto the medical record file.

- **Updating the turnaround time:**



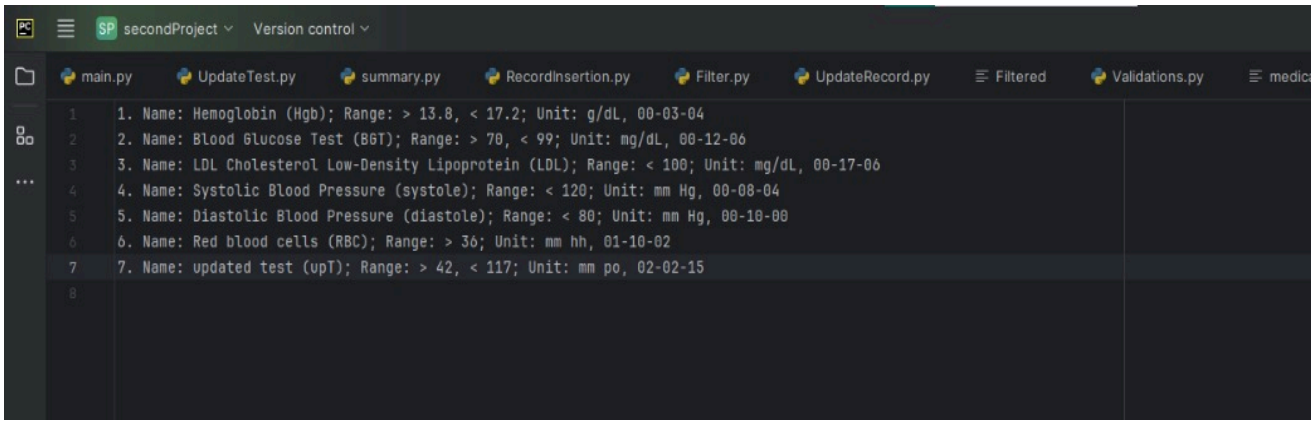
After updating the unit of that same test, based on the inputted data viewed in the previous picture , the test became :

Name: updated test (upT); Range: > 42, <117; Unit: mm po, 02-02-15

- **The medical Test file after the updation:**

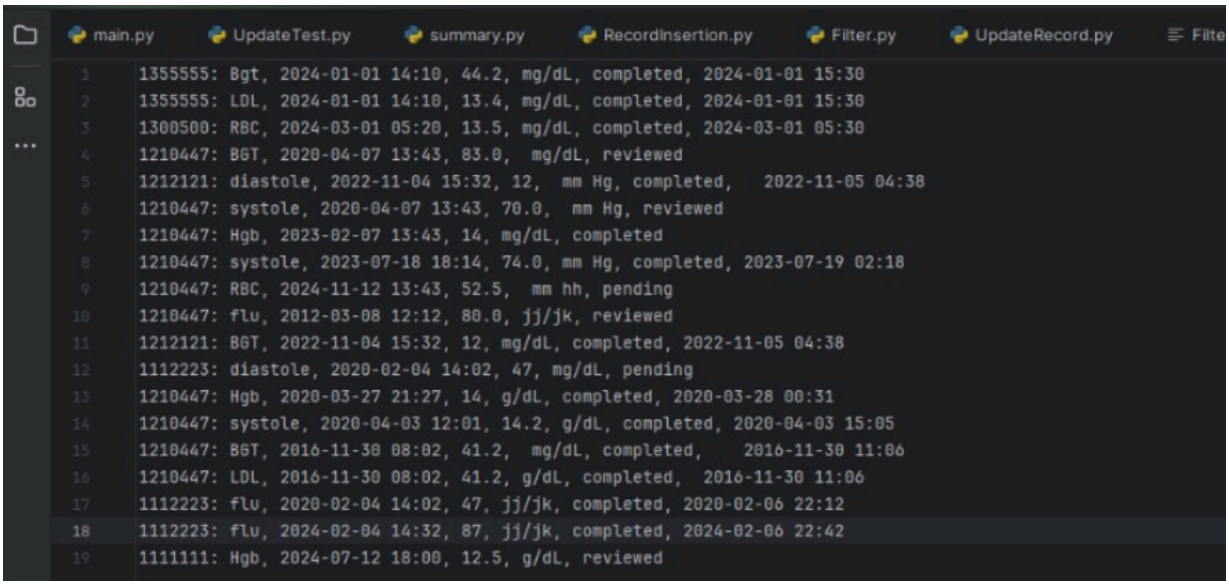
Observing the medical test file after the update occurred, the old flu test has been completely replaced by the new record :

Name: updated test (upT); Range: > 42, <117; Unit: mm po, 02-02-15

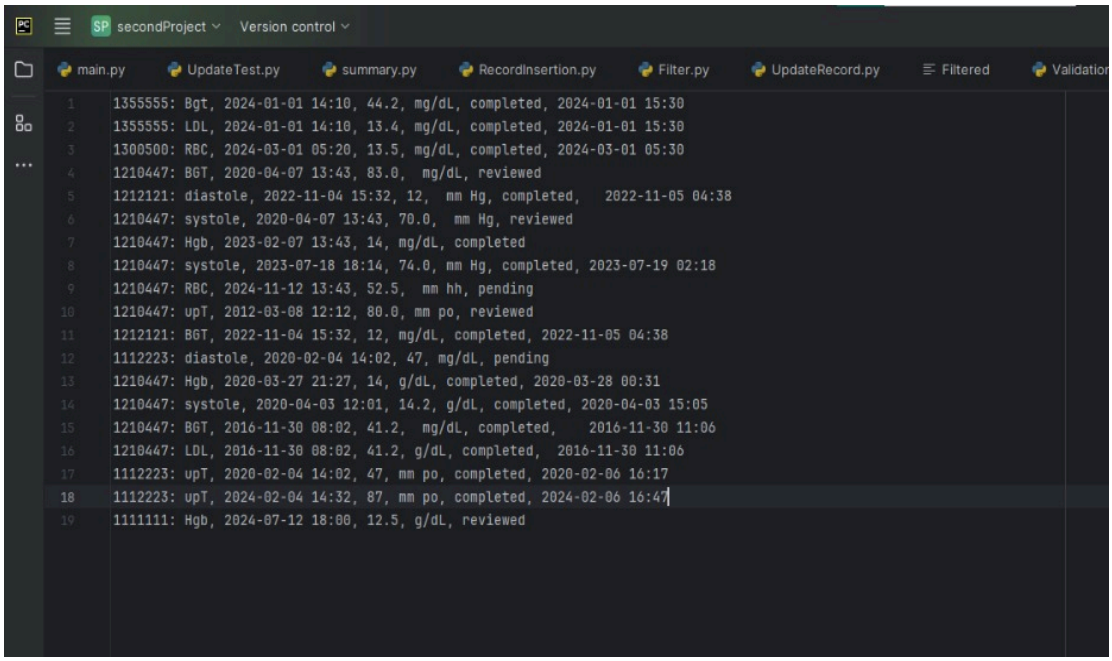


The changes were reflected onto the medical record file, and observed as viewed down below.

- **The old medical record Content:**



- **The new medical record Content:**

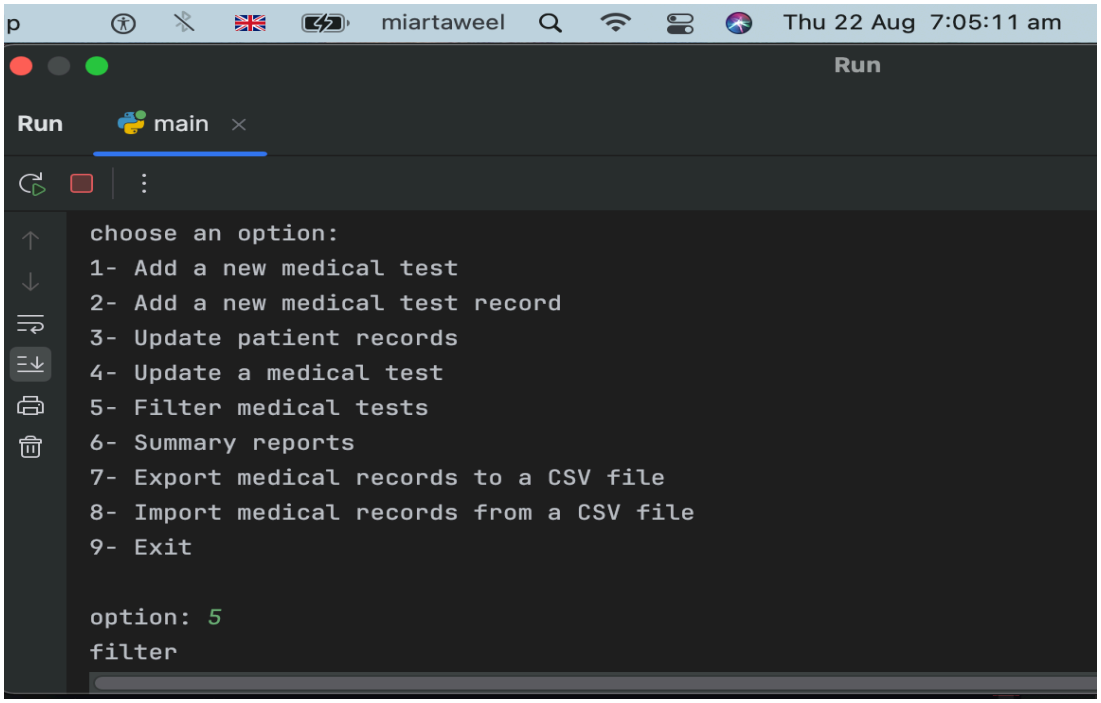


Observing the medical record file after the update occurred , all the previous flu test records have been altered to fit the data of the updated medical test , where the abbreviation name , unit and completion time has been updated accordingly .

3.5. Filtering:

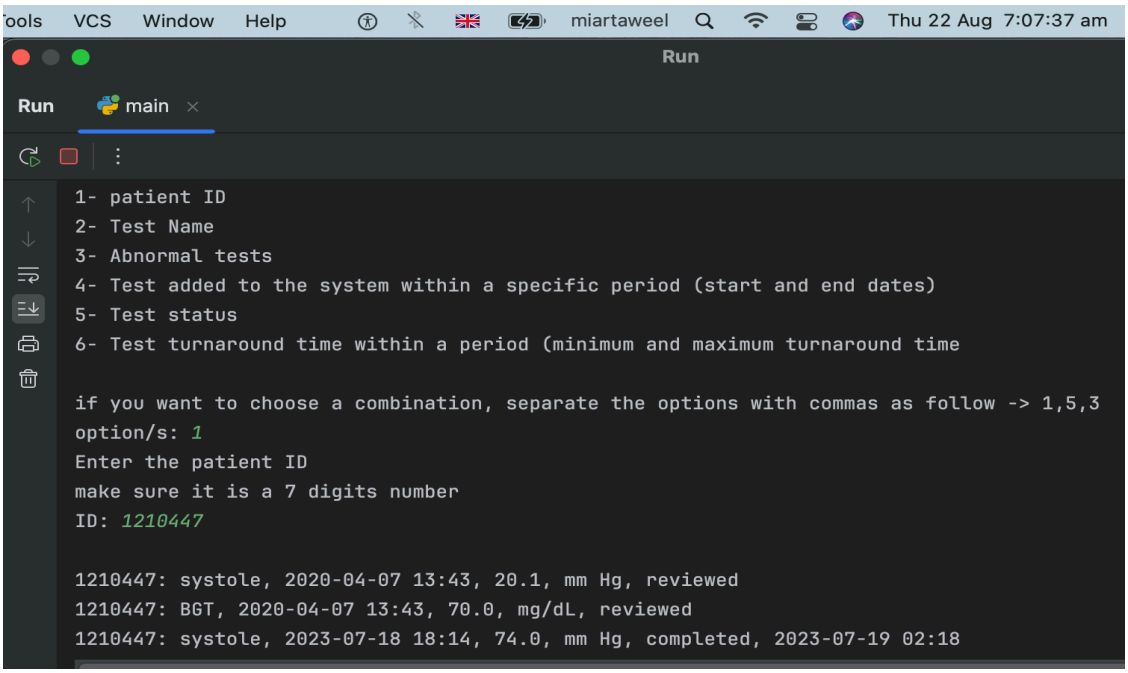
The following images represent searching for a record in the medical record file:

The operation starts by selecting the search by patient ID option from the option menu, then inserting an existing patient ID to search for :



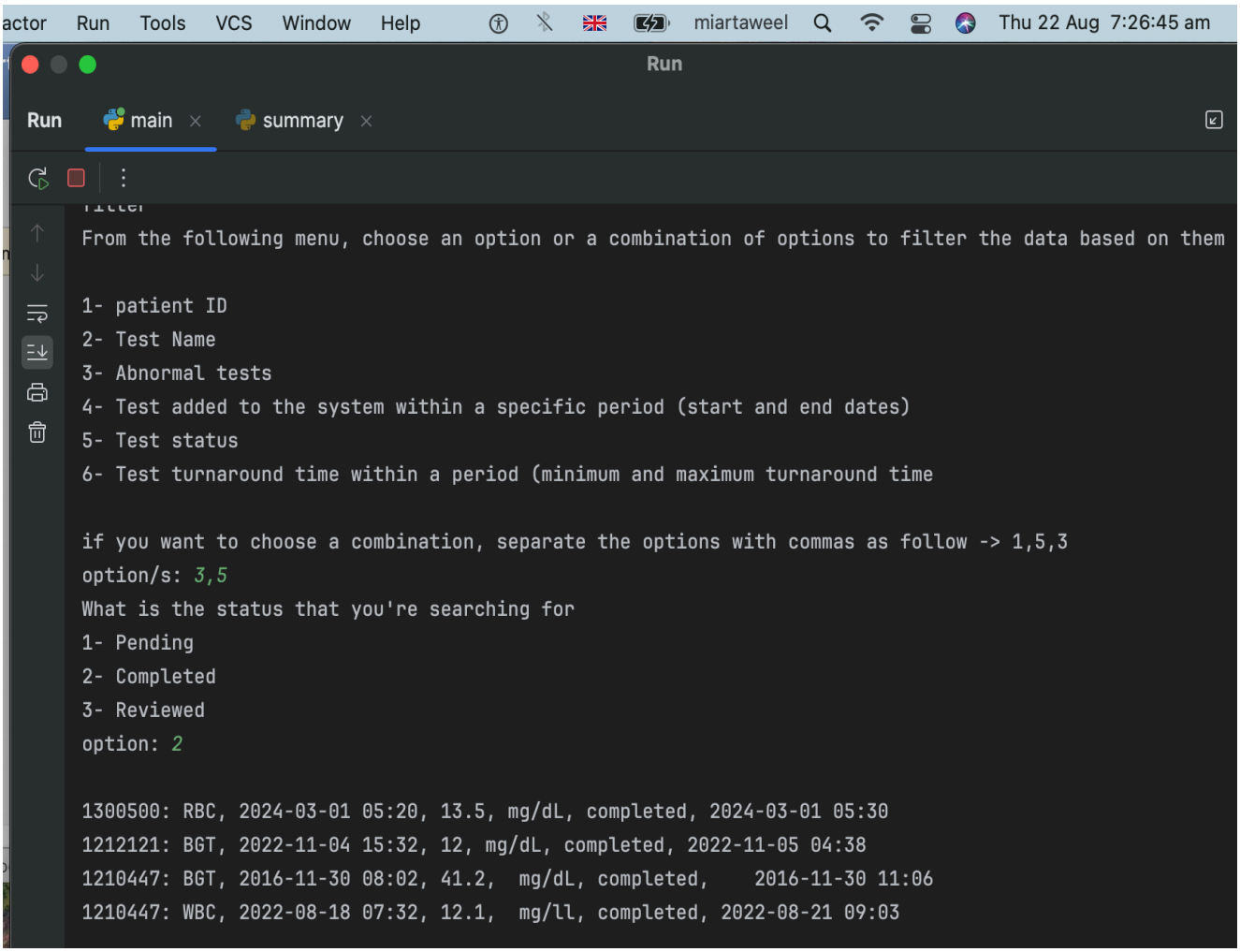
- **All Patients Retrieved:**

An option menu then appears, then the “patient ID” filter option is chosen which retrieves all the records of that patient as viewed down below :



- **Multiple filters selected :**

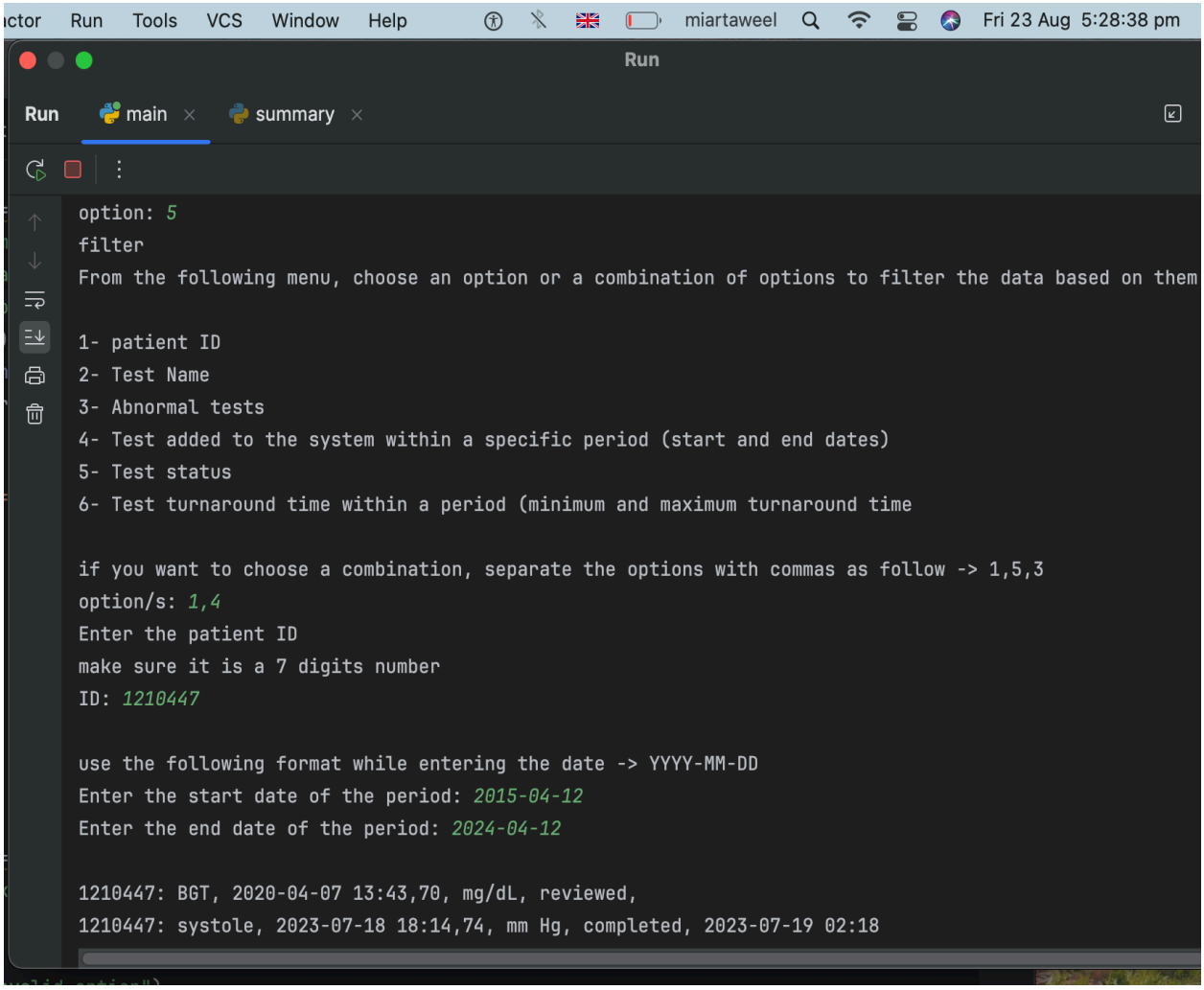
  Since as observed in the menu a combination of filters can be selected to filter the records , the following image represents choosing two filters , **Abnormality and status,** which were chosen by entering their indexes in the following format “index1,index2”as seen below :



As viewed in the image , the resulting records are records which are both abnormal and are also completed , thus choosing multiple filters was executed successfully

- **All Patients Retrieved in a specific period**

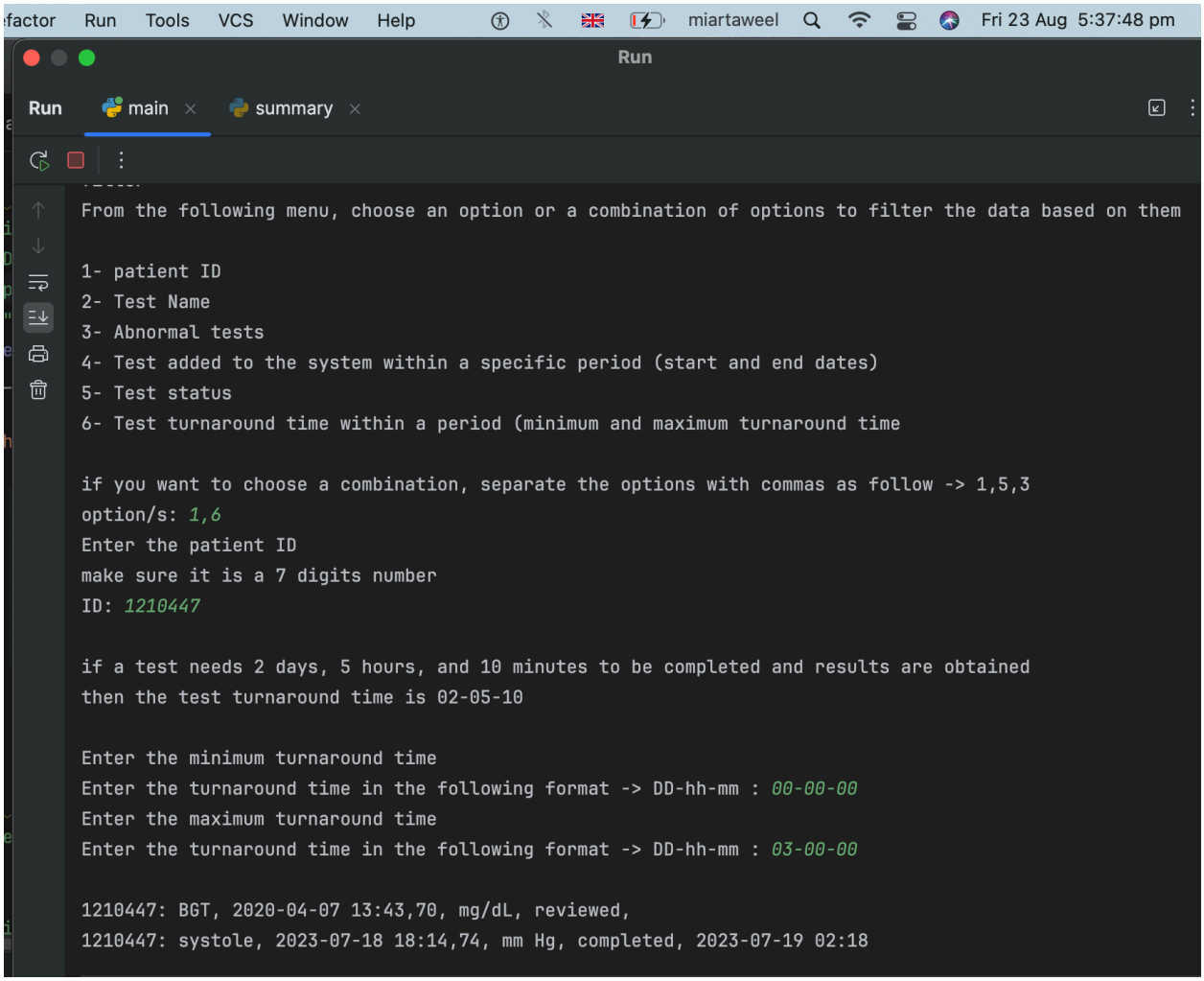
  In this case , as applied previously a combination of filters can be selected to filter the records , the following image represents choosing two filters : **the ID** and **a specific period,** which were chosen by entering their indexes in the following format “index1,index2”as seen below :



As viewed in the image , the resulting records are records which are between 2015-04-12 and 2024-04-12, in addition to the ID being equal to 1210447, thus filtering by choosing multiple filters was executed successfully.

- **All Patients Retrieved for a specific turnaround range :**

  In this case , as applied previously a combination of filters can be selected to filter the records , the following image represents choosing two filters : **a specific ID and a specific turnaround range,** which were chosen by entering their indexes in the following format “index1,index2”as seen below :

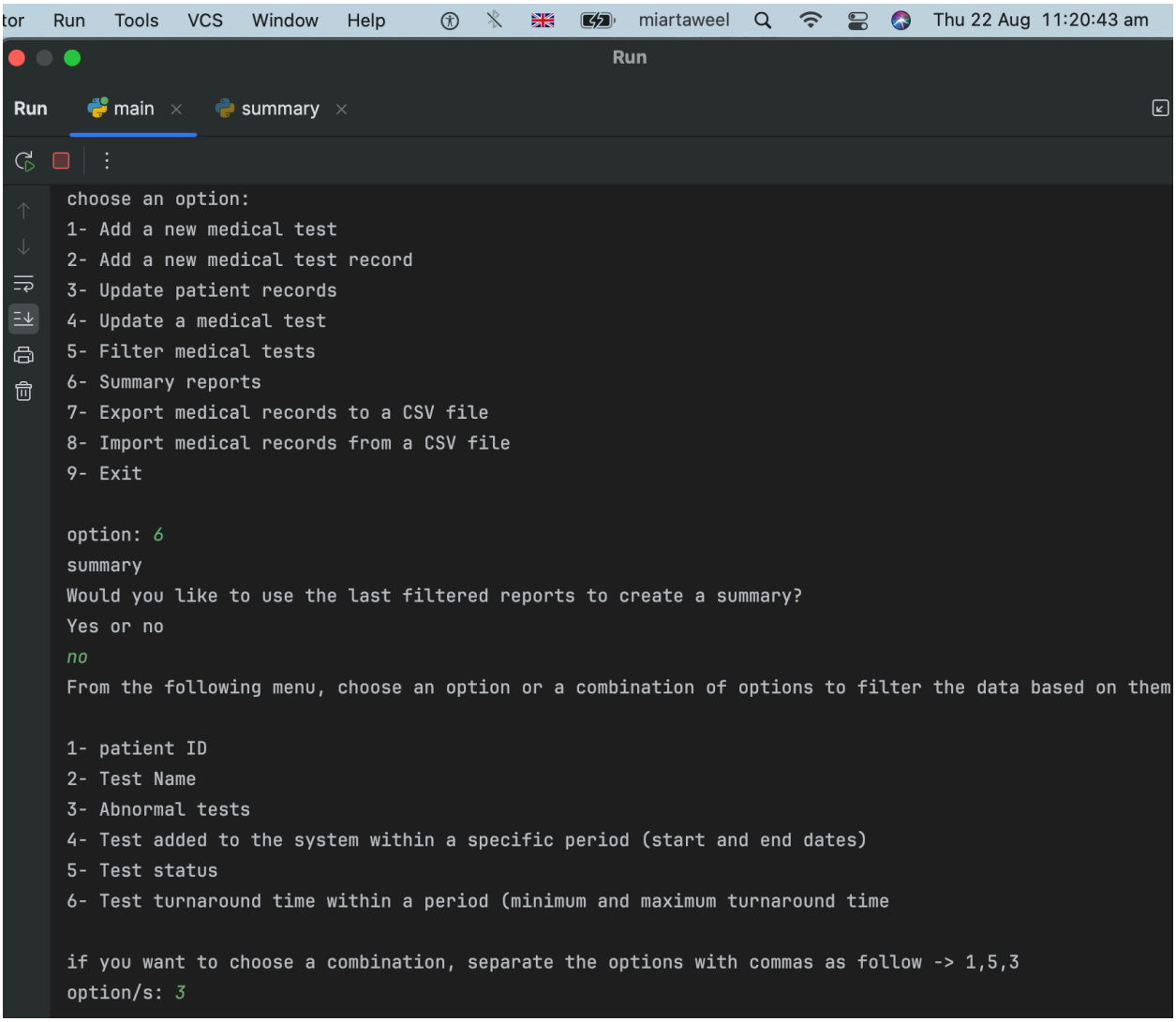


  As viewed in the image , the resulting records are records which have an ID of 1210447, in addition to the turnaround time being up to three days , thus filtering by choosing multiple filters was executed successfully.

3.6.Summary Reports:

Summary reports are generated by filtering the records as done in the previous part, then finding the minimum maximum and average value for each test type in the records and turnaround time , viewed down below :

The operation starts by selecting the “A Summary reports” option from the drop down menu, which then provides the user the choice of creating a report based on the last executed filtering operation, if the user rejects , a new filtering process based on the users choice is excecuted:



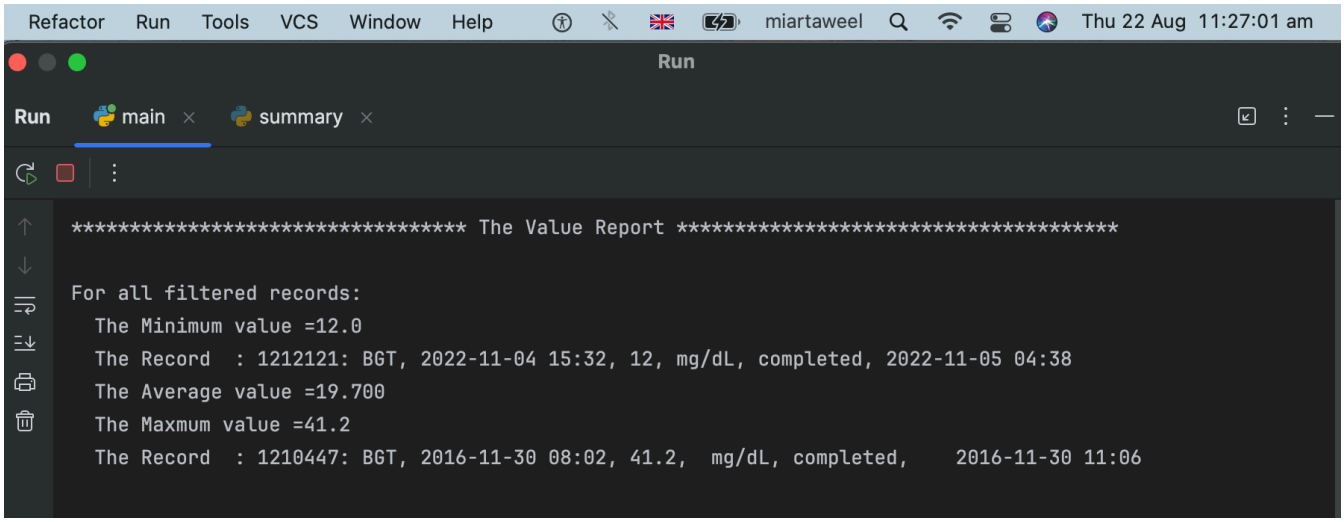
As viewed in the picture, the records were chosen to be filtered based on their abnormality.

```
if you want to choose a combination, separate the options with commas as follow -> 1,5,3
option/s: 3

1300500: RBC, 2024-03-01 05:20, 13.5, mg/dL, completed, 2024-03-01 05:30
1212121: BGT, 2022-11-04 15:32, 12, mg/dL, completed, 2022-11-05 04:38
1210447: BGT, 2016-11-30 08:02, 41.2,  mg/dL, completed,    2016-11-30 11:06
1210447: WBC, 2022-08-18 07:32, 12.1,  mg/ll, completed, 2022-08-21 09:03
```

After of filtering the records as seen in the previous image , Reports for each the value and the turnaround time are found :

- **The Value Summary Report : for all the records**



- **The Value Summary Report : for each test type**

```
      ------------------------------------------------
For each filtered Test type:

-Hemoglobin (Hgb):
  Records of this type dont exit

-Blood Glucose Test (BGT):
  The Minimum value =12.0
  The Minimum Record  : 1212121: BGT, 2022-11-04 15:32, 12, mg/dL, completed, 2022-11-05 04:38
  The Maximum value =41.2
  The Maximum Record  =1210447: BGT, 2016-11-30 08:02, 41.2,  mg/dL, completed,    2016-11-30 11:06
  The average  =26.600

-LDL Cholesterol Low-Density Lipoprotein (LDL):
  Records of this type dont exit

-Systolic Blood Pressure (systole):
  Records of this type dont exit
```

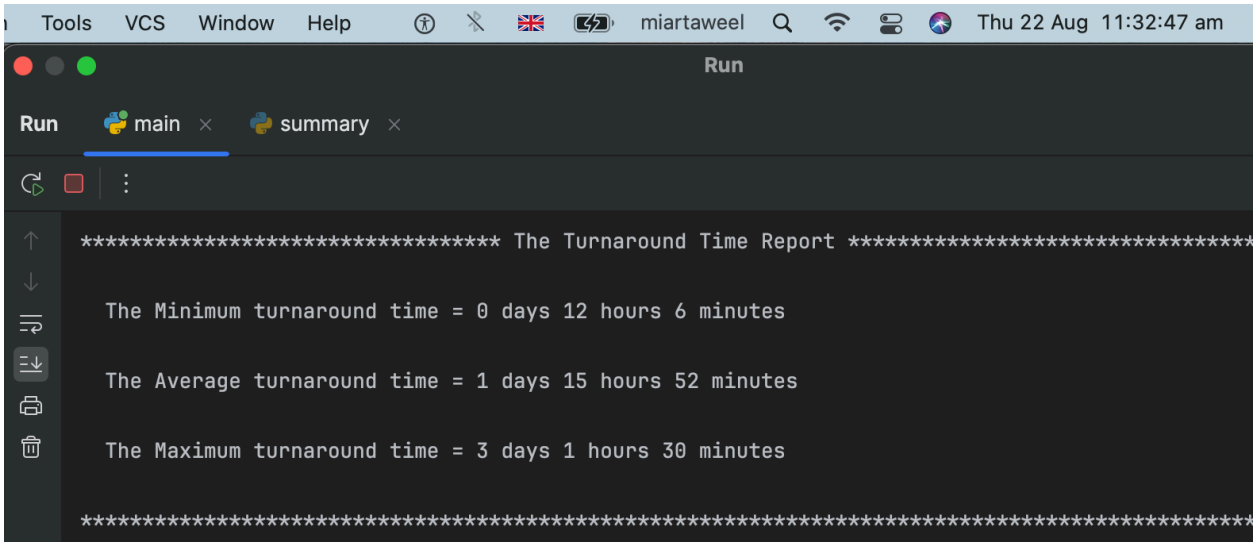
```
-Diastolic Blood Pressure (diastole):
  Records of this type dont exit

-Red blood cells (RBC):
  The Minimum value =13.5
  The Minimum Record  : 1300500: RBC, 2024-03-01 05:20, 13.5, mg/dL, completed, 2024-03-01 05:30
  The Maximum value =13.5
  The Maximum Record  =1300500: RBC, 2024-03-01 05:20, 13.5, mg/dL, completed, 2024-03-01 05:30
  The average  =13.500

-White Blood cells (WBC):
  The Minimum value =12.1
  The Minimum Record  : 1210447: WBC, 2022-08-18 07:32, 12.1,  mg/ll, completed, 2022-08-21 09:03
  The Maximum value =12.1
  The Maximum Record  =1210447: WBC, 2022-08-18 07:32, 12.1,  mg/ll, completed, 2022-08-21 09:03
  The average  =12.100
```

As viewed in the previous images, for each of the test types the minimum value ,maximum value and the records which contains each were found and printed , in addition to calculating the average of each.

- **The turnaround time Summary Report : for all the Test types**

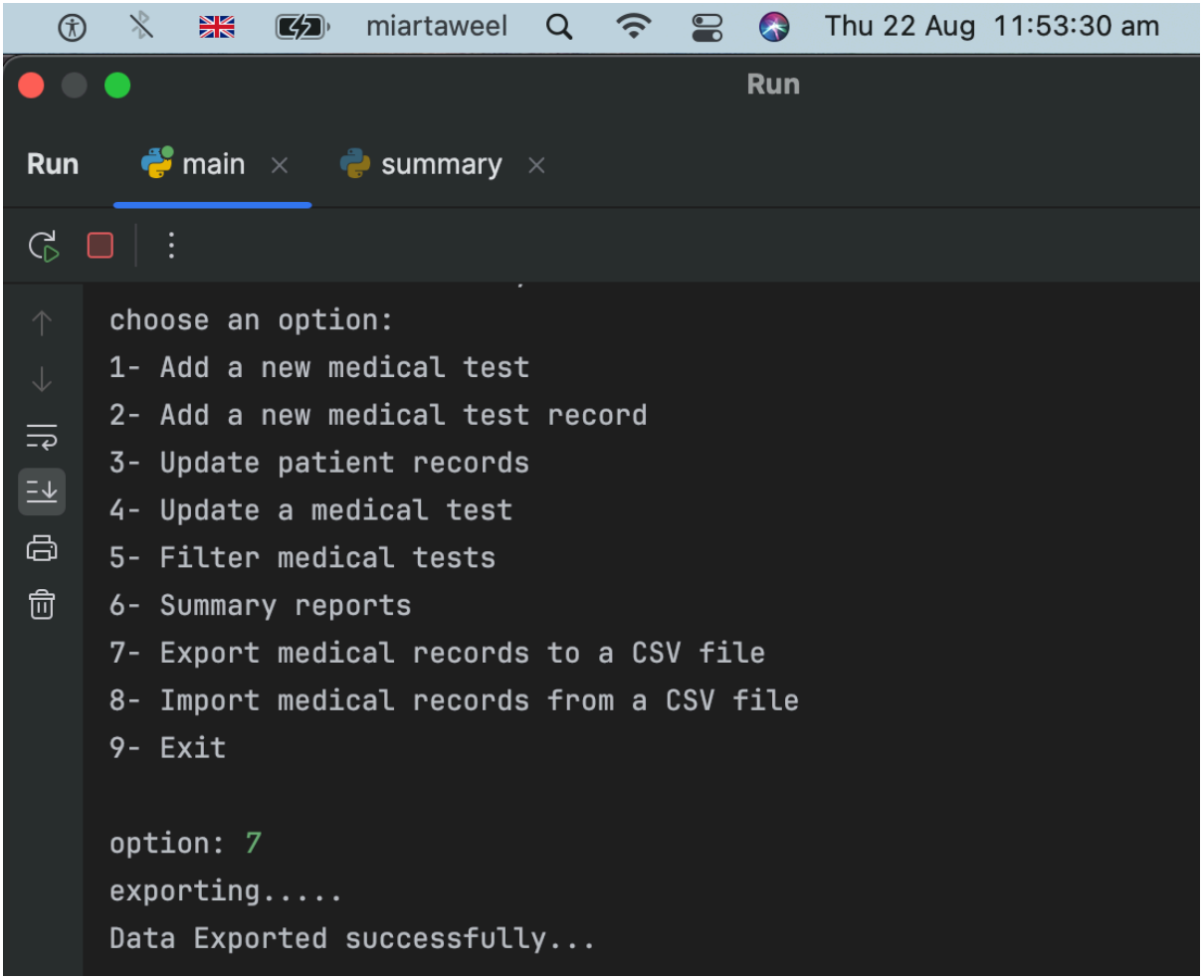


In the turnaround report, for each of the test types the minimum turnaround time ,maximum turnaround time were found and printed , in addition to calculating the average of each time of all the test types.

3.7. Exporting Records to a CSV file:

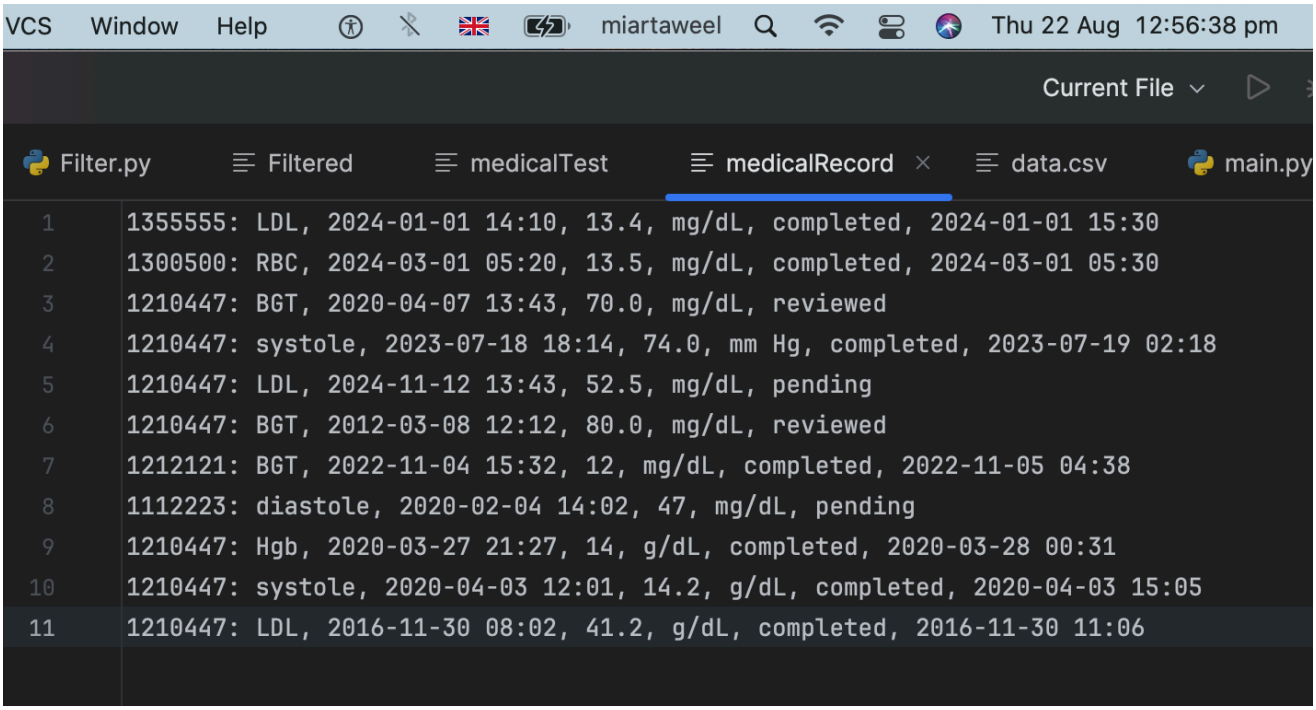
After performing multiple operations on the medical record file , the new resulting records should be saved to a .csv file , to organize the data efficiently.

The operation starts by selecting the “Export medical records to a CSV file “ option from the drop down menu:



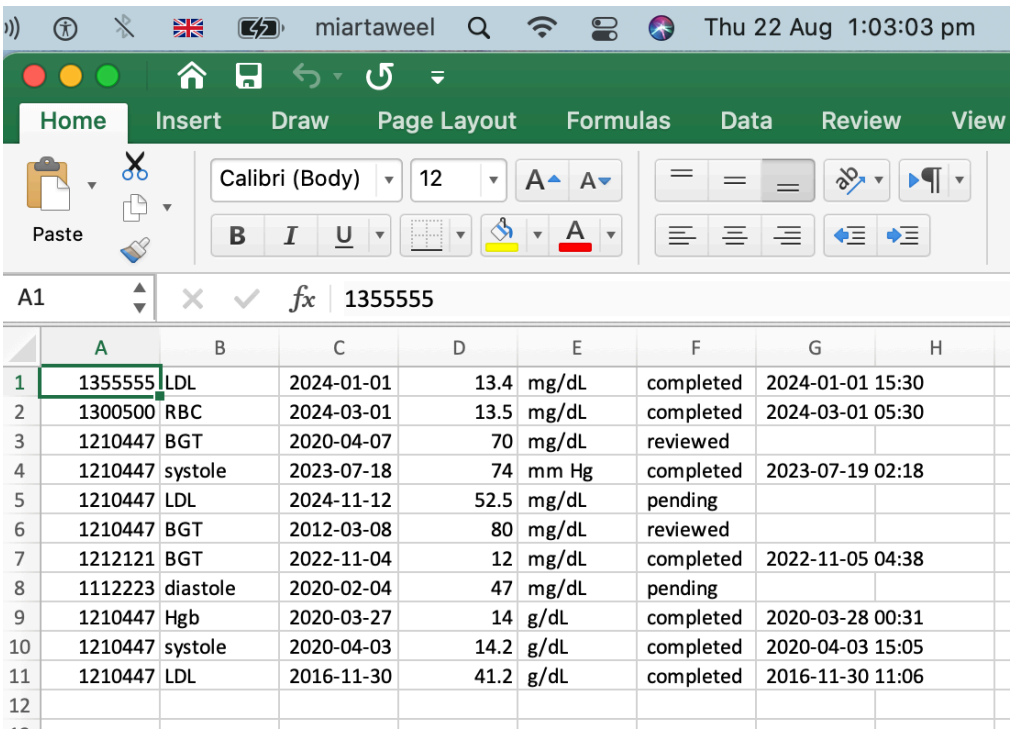
After choosing the export option form the menu , the data from the medical record file is exported to the csv file

- **Medical Record file Data::**



- **Output viewed in CSV file :**

According to the previous image the data from the file should be found in the export file down below :

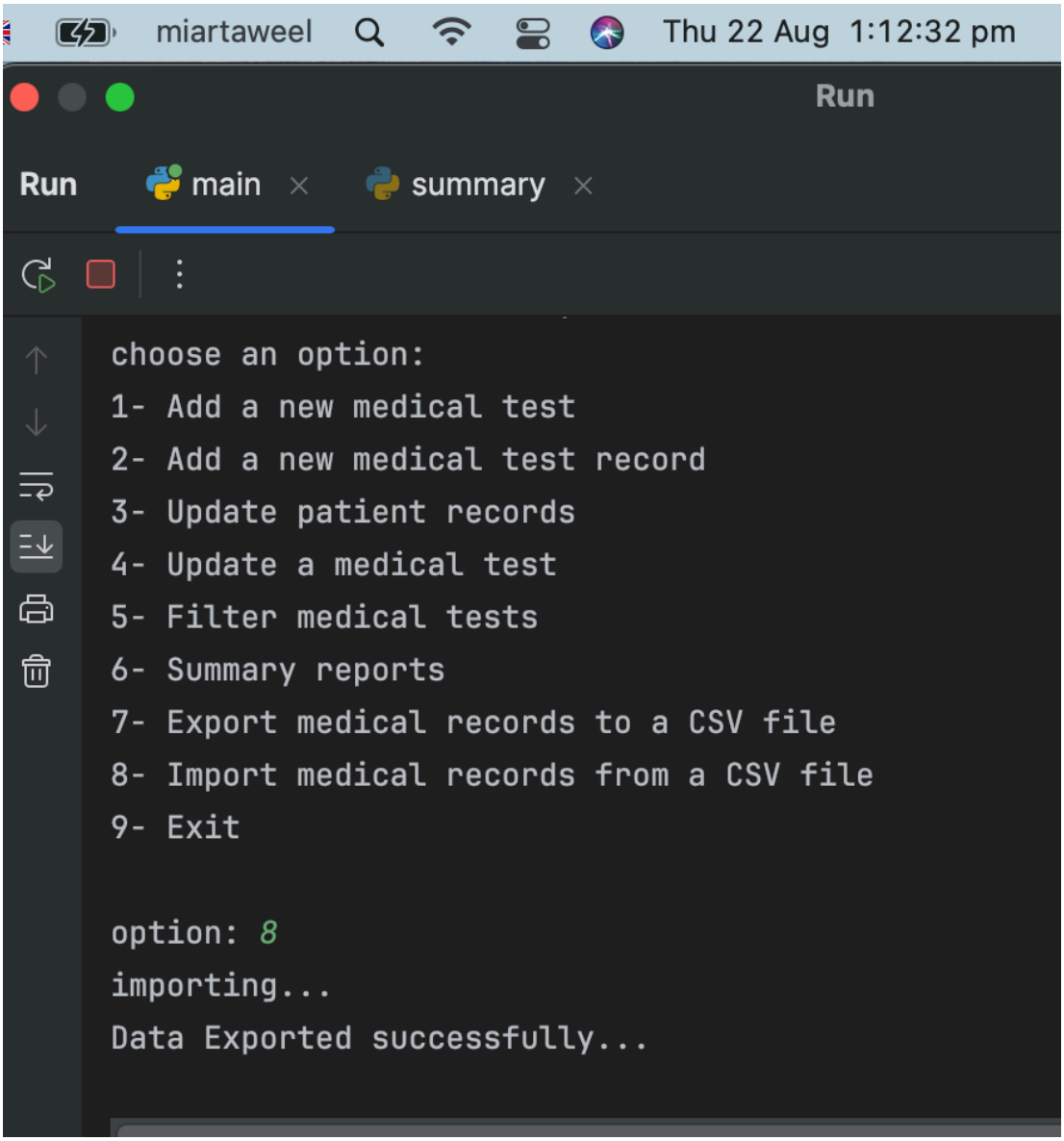


	A	B	C	D	E	F	G	H
1	1355555	LDL	2024-01-01	13.4	mg/dL	completed	2024-01-01 15:30	
2	1300500	RBC	2024-03-01	13.5	mg/dL	completed	2024-03-01 05:30	
3	1210447	BGT	2020-04-07	70	mg/dL	reviewed		
4	1210447	systole	2023-07-18	74	mm Hg	completed	2023-07-19 02:18	
5	1210447	LDL	2024-11-12	52.5	mg/dL	pending		
6	1210447	BGT	2012-03-08	80	mg/dL	reviewed		
7	1212121	BGT	2022-11-04	12	mg/dL	completed	2022-11-05 04:38	
8	1112223	diastole	2020-02-04	47	mg/dL	pending		
9	1210447	Hgb	2020-03-27	14	g/dL	completed	2020-03-28 00:31	
10	1210447	systole	2020-04-03	14.2	g/dL	completed	2020-04-03 15:05	
11	1210447	LDL	2016-11-30	41.2	g/dL	completed	2016-11-30 11:06	
12								

3.8. Importing Records from a CSV file:

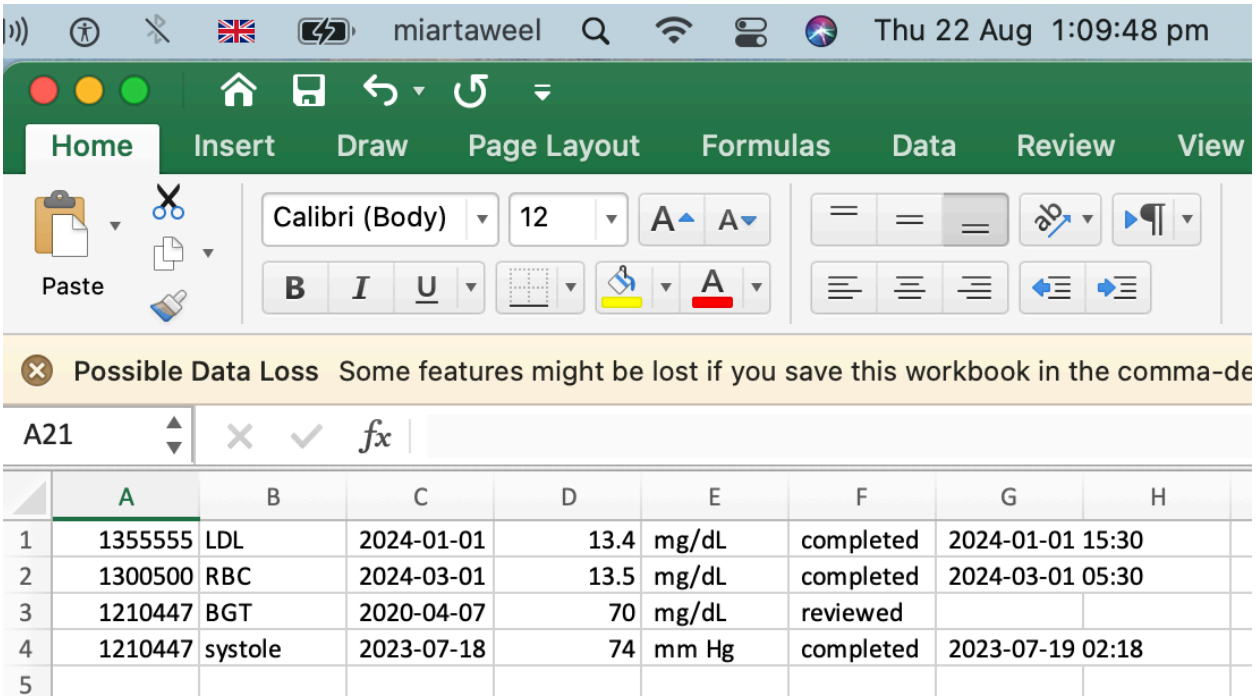
Before performing operations on the medical record file , the records should be imported from a .csv file

The operation starts by selecting the “Import medical records from a CSV file “ option from the drop down menu:



After choosing the option form the menu , the data from the medical record file is filled with the csv files data

- **CSV file Data::**



	A	B	C	D	E	F	G	H
1	1355555	LDL	2024-01-01	13.4	mg/dL	completed	2024-01-01 15:30	
2	1300500	RBC	2024-03-01	13.5	mg/dL	completed	2024-03-01 05:30	
3	1210447	BGT	2020-04-07	70	mg/dL	reviewed		
4	1210447	systole	2023-07-18	74	mm Hg	completed	2023-07-19 02:18	
5								

- **Output viewed in medical Record file :**

According to the previous image the data from the file should be found in the medical records file as seen down below :