



# BIRZEIT UNIVERSITY

**Faculty of Engineering & Technology**

**Linux Laboratory – ENCS3130**

**Python Project – Medical Test Management System**

---

**Prepared by:**

**Name:** Francis Miadi      **Number:** 1210100

**Name:** Miar Taweelel      **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the medical record file, enabling the seamless addition of new test results, updating of existing records, deletion of outdated or incorrect entries, and retrieval of specific test results based on patient ID, test type, or other criteria.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.

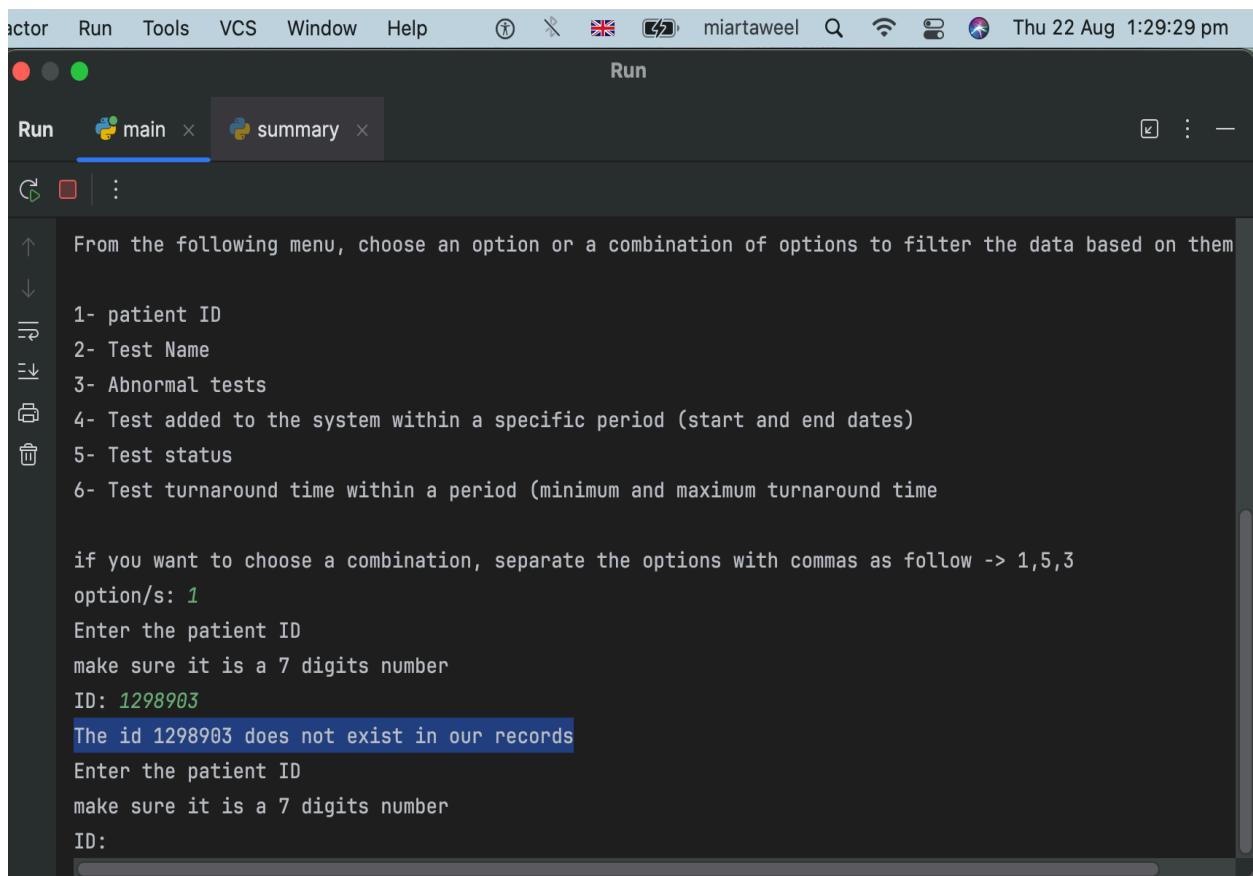
## Table of Contents

<b>1. Error Handling:</b> .....	<b>3</b>
1.1. Non Existant Patient: .....	3
1.2. Non Existant .CSV file: .....	4
1.3. Non Existant Record file:.....	5
1.4. Empty files:.....	6
1.5. Empty filtering file:.....	7
<b>2. Data Validation:</b> .....	<b>8</b>
2.1. Patient ID: .....	8
2.2. Date Checking:.....	9
2.3. Test Value Checking:.....	12
2.4. Test Type and status checking:.....	12
2.5. Valid Date Range Checking : .....	14
2.6. Valid Test name and abbreviation: .....	15
2.7. Valid Range Checking : .....	17
2.8. Unit Checking: .....	18
2.9. Turnaround Time Checking:.....	19
2.10. Invalid menu option : .....	20
<b>3. Execution :</b> .....	<b>21</b>
3.1. Inserting a medical test : .....	21
3.2. Inserting a medical record :.....	23
3.3. Update Record: .....	26
3.4. Update Medical Test:.....	29
3.5. Filtering:.....	33
3.6. Summary Reports: .....	37
3.7. Exporting Records to a CSV file: .....	40
3.8. Importing Records from a CSV file:.....	42

# 1. Error Handling:

## 1.1. Non Existant 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 :**



```
Run    Run main × summary ×
```

```
From the following menu, choose an option or a combination of options to filter the data based on them
1- patient ID
2- Test Name
3- Abnormal tests
4- Test added to the system within a specific period (start and end dates)
5- Test status
6- Test turnaround time within a period (minimum and maximum turnaround time)

if you want to choose a combination, separate the options with commas as follow -> 1,5,3
option/s: 1
Enter the patient ID
make sure it is a 7 digits number
ID: 1298903
The id 1298903 does not exist in our records
Enter the patient ID
make sure it is a 7 digits number
ID:
```

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 Existant .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** :

The screenshot shows a Python development environment with the following details:

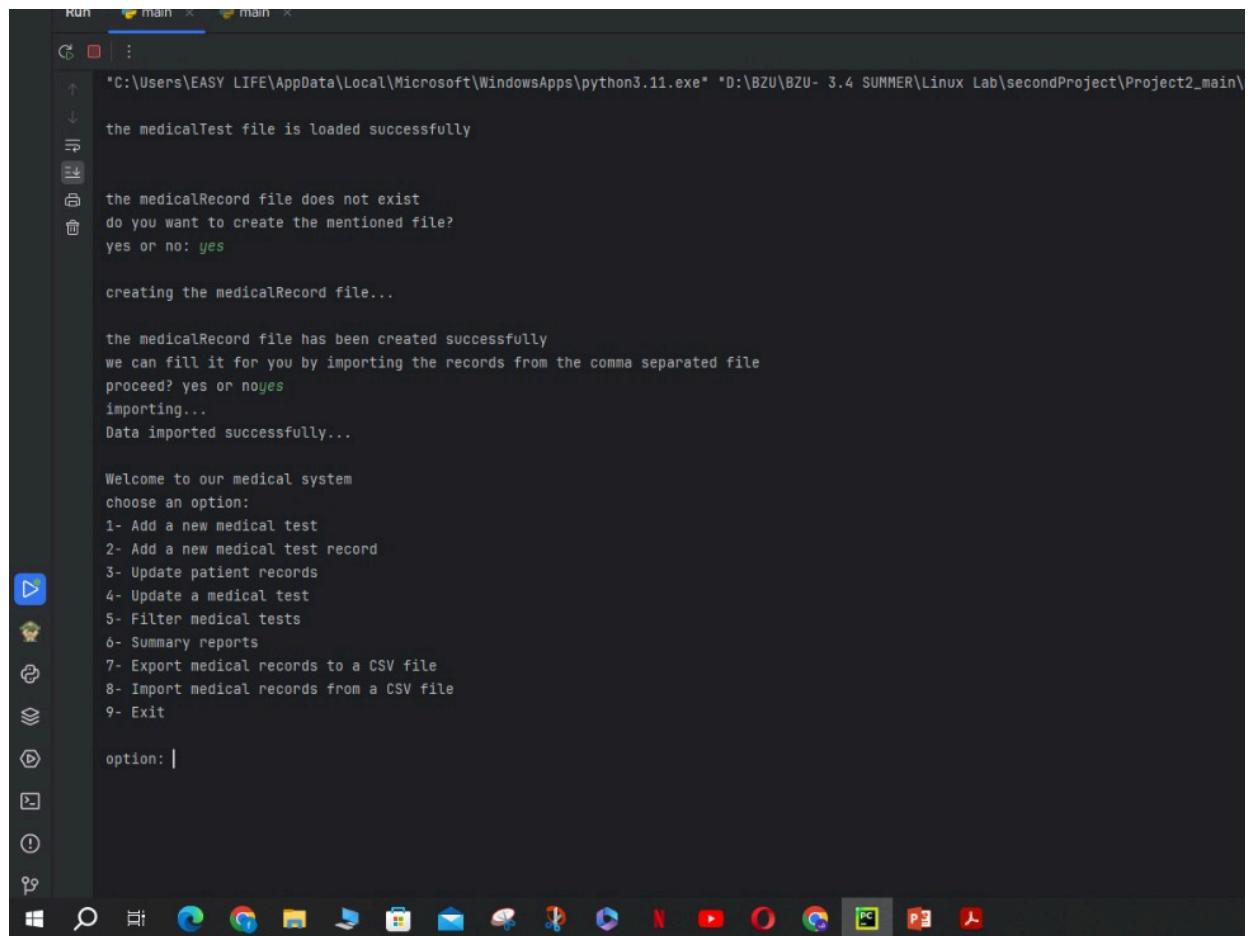
- Project Structure:** The project is named "secondProject". Inside "secondProject", there is a folder "Project2\_main" which contains a ".venv" folder and several Python files: Filter.py, main.py, medicalRecord, medicalTest, RecordInsertion.py, summary.py, UpdateRecord.py, UpdateTest.py, and Validations.py.
- Code Editor:** The main.py file is open. The code checks if "data.csv" exists. If it does not, it prints an error message and exits with code 1.
- Terminal:** The terminal window shows the command "python main.py" being run. The output indicates that "data.csv" does not exist, and the program exits with code 1.

```
i=0
flag=0
recordsFile="medicalRecord"
testsFile="medicalTest"
dataFile="data.csv"
emptyData=0
filteredFile="filtered"
path=os.path.abspath(__file__)
directory=os.path.dirname(path)
medicalRecord=os.path.join(directory,recordsFile)
if os.path.exists(dataFile):
    if os.path.getsize(dataFile)==0:
        emptyData=1
    else:
        print()
        print("the data.csv file does not exist, which will cause a problem in importing and exporting the data")
        print("create one and try again")
        exit("forced exiting")
Process finished with exit code 1
```

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 Existant 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 :**



The screenshot shows a terminal window titled "main" with the following text output:

```
*C:\Users\EASY LIFE\AppData\Local\Microsoft\WindowsApps\python3.11.exe* "D:\BZU\BZU- 3.4 SUMMER\Linux Lab\secondProject\Project2_main"
the medicalTest file is loaded successfully

the medicalRecord file does not exist
do you want to create the mentioned file?
yes or no: yes

creating the medicalRecord file...

the medicalRecord file has been created successfully
we can fill it for you by importing the records from the comma separated file
proceed? yes or noyes
importing...
Data imported successfully...

Welcome to our medical system
choose an option:
1- Add a new medical test
2- Add a new medical test record
3- Update patient records
4- Update a medical test
5- Filter medical tests
6- Summary reports
7- Export medical records to a CSV file
8- Import medical records from a CSV file
9- Exit

option: |
```

The terminal window is running on a Windows operating system, as indicated by the taskbar icons at the bottom.

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 :**

```
*C:\Users\EASY LIFE\AppData\Local\Microsoft\WindowsApps\python3.11.exe* "D:\BZU\BZU- 3.4 SUMMER\Linux Lab\secondProject\Project2_main\main.py"
the medicalTest file is loaded successfully

the medicalRecord file is empty, you can't use our program unless you fill it
we can fill it for you by importing the records from the comma separated file
proceed? yes or no: yes
importing...
Data imported successfully...

Welcome to our medical system
choose an option:
1- Add a new medical test
2- Add a new medical test record
3- Update patient records
4- Update a medical test
5- Filter medical tests
6- Summary reports
7- Export medical records to a CSV file
8- Import medical records from a CSV file
9- Exit

option: |
```

- **An empty medical Test file :**

```
*C:\Users\EASY LIFE\AppData\Local\Microsoft\WindowsApps\python3.11.exe* "D:\BZU\BZU- 3.4 SUMMER\Linux Lab\secondProject\Project2_main\main.py"
the file medicalTest is empty, fill it with proper data to use our program
Exiting the program

Process finished with exit code 1
```

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 :

The screenshot shows a terminal window with a dark theme. At the top, there's a toolbar with icons for Help, Run, and two tabs: "main" and "summary". The main content area displays a menu with the following options:

```
choose an option:  
1- Add a new medical test  
2- Add a new medical test record  
3- Update patient records  
4- Update a medical test  
5- Filter medical tests  
6- Summary reports  
7- Export medical records to a CSV file  
8- Import medical records from a CSV file  
9- Exit
```

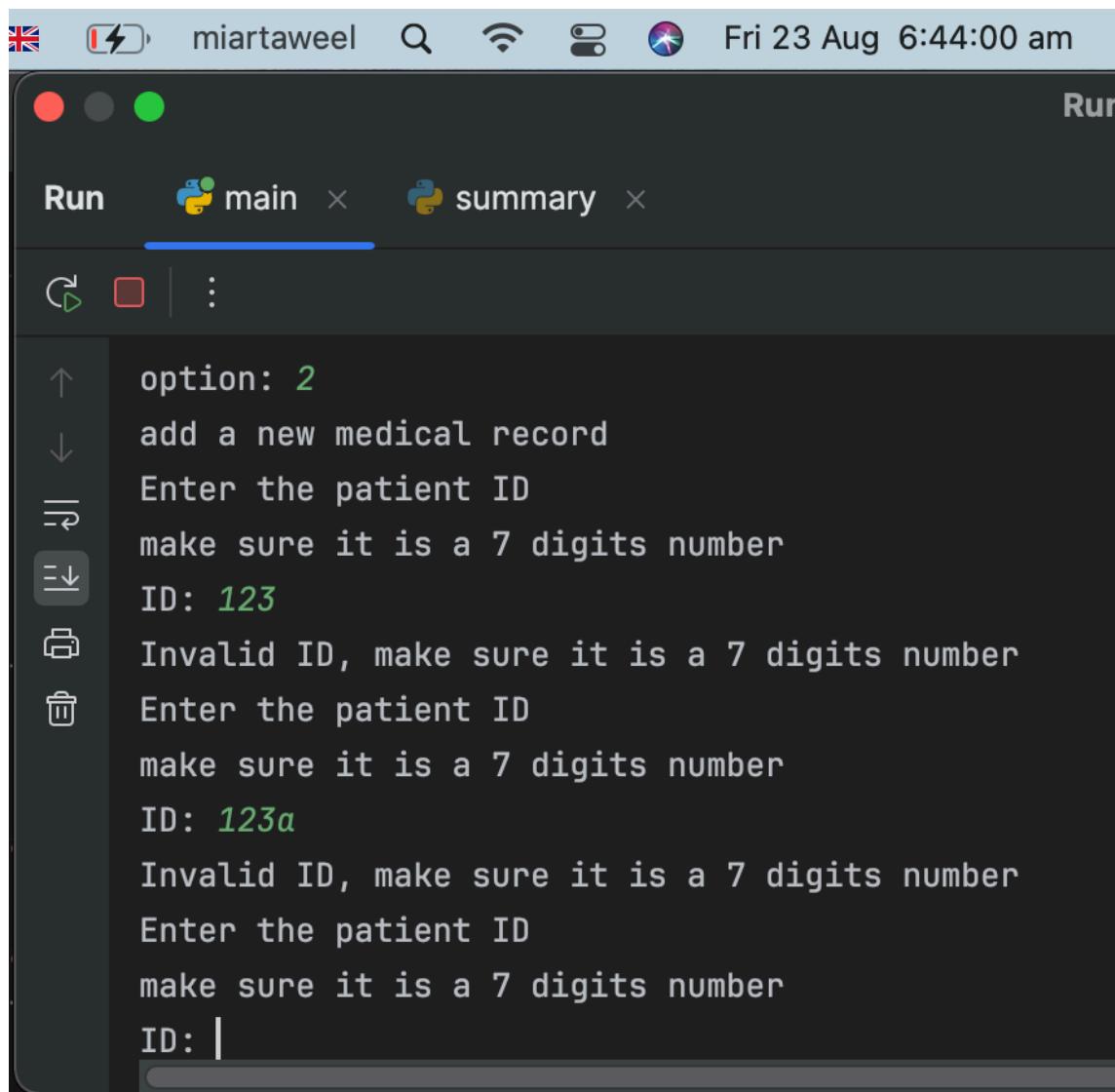
Below the menu, the user types "option: 6" and "summary". A question "Would you like to use the last filtered reports to create a summary?" is displayed, followed by "Yes or no" and "yes". The final message is "No filtered Records exist".

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 :



The screenshot shows a terminal window titled "Run" with two tabs: "main" and "summary". The "main" tab is active, indicated by a blue underline. The terminal output displays a script for adding a new medical record. It asks for a patient ID, specifies it must be a 7-digit number, and attempts to enter "123" and "123a", both of which are rejected as invalid. The user is then prompted to re-enter the ID.

```
option: 2
add a new medical record
Enter the patient ID
make sure it is a 7 digits number
ID: 123
Invalid ID, make sure it is a 7 digits number
Enter the patient ID
make sure it is a 7 digits number
ID: 123a
Invalid ID, make sure it is a 7 digits number
Enter the patient ID
make sure it is a 7 digits number
ID: |
```

## 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

```
7- White Blood cells (WBC)
option: 1
Hgb

use the following format while entering the date -> YYYY-MM-DD
Enter the date of the test: 2024-12-40
stick to the given format, invalid day

use the following format while entering the date -> YYYY-MM-DD
Enter the date of the test:
```

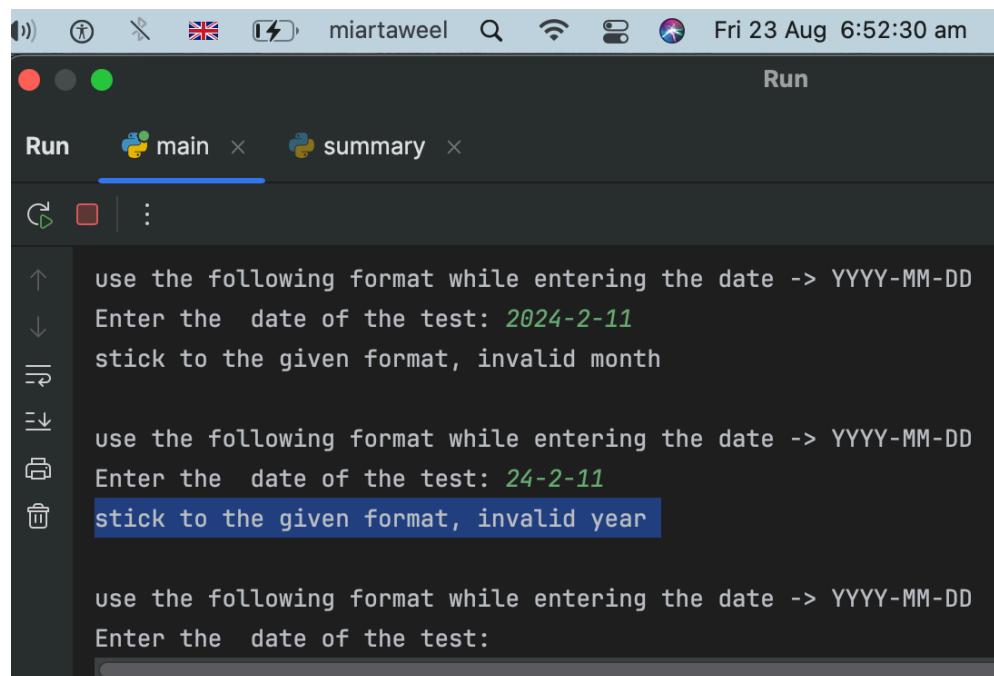
- **Month Validity** : An invalid month was inputted

```
use the following format while entering the date -> YYYY-MM-DD
Enter the date of the test: 2024-12-40
stick to the given format, invalid day

use the following format while entering the date -> YYYY-MM-DD
Enter the date of the test: 2024-13-29
stick to the given format, invalid month

use the following format while entering the date -> YYYY-MM-DD
Enter the date of the test:
```

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



```

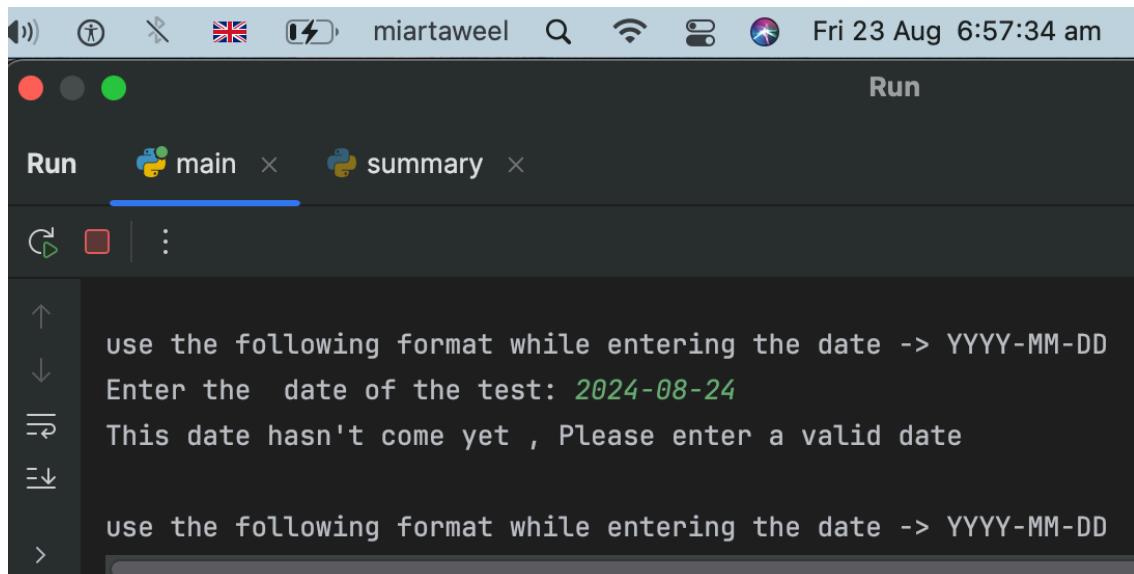
use the following format while entering the date -> YYYY-MM-DD
Enter the date of the test: 2024-2-11
stick to the given format, invalid month

use the following format while entering the date -> YYYY-MM-DD
Enter the date of the test: 24-2-11
stick to the given format, invalid year

use the following format while entering the date -> YYYY-MM-DD
Enter the date of the test:

```

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



```

use the following format while entering the date -> YYYY-MM-DD
Enter the date of the test: 2024-08-24
This date hasn't come yet , Please enter a valid date

use the following format while entering the date -> YYYY-MM-DD

```

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 it didn't come yet thus it was rejected and required the user to enter a valid value

The screenshot shows a macOS terminal window titled "Run". It has two tabs open: "main" and "summary". The "main" tab is active and contains the following Python code and its execution output:

```
use the following format while entering the date -> YYYY-MM-DD
Enter the date of the test: 2024-08-23

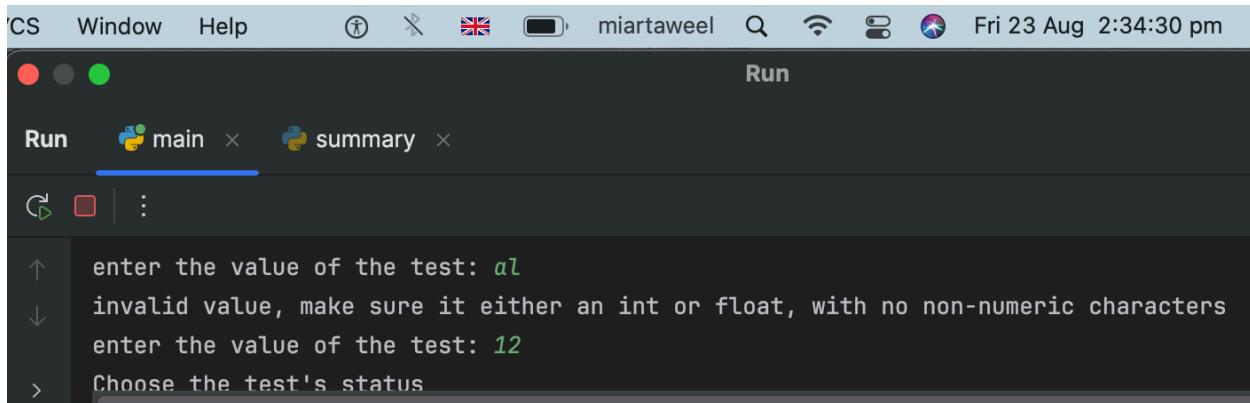
use the following format while entering the time -> HH:MM
Enter the time of the test: 07:00
This time hasn't come yet , Please enter a valid time

use the following format while entering the time -> HH:MM
Enter the time of the test:
```

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 :



A screenshot of a terminal window titled 'Run' showing a Python script execution. The window title bar includes icons for CS, Window, Help, and system status. The terminal shows the following interaction:

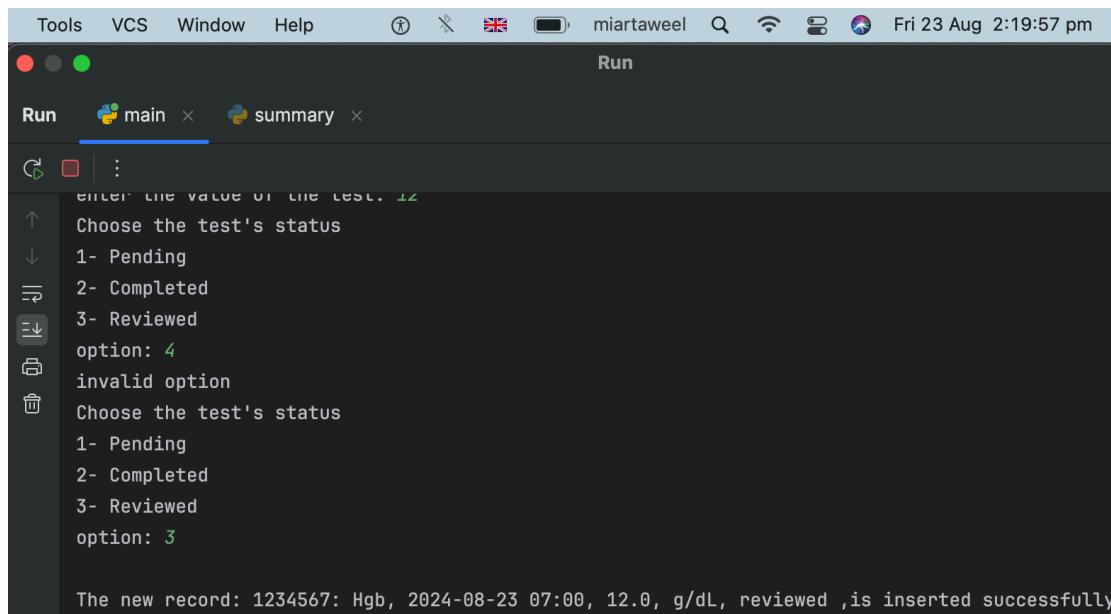
```
↑ enter the value of the test: al
↓ invalid value, make sure it either an int or float, with no non-numeric characters
↑ enter the value of the test: 12
> Choose the test's status
```

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 of 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 type and status a menu is printed to choose a valid existing value from it, as shown in the following images :

- **Test Status:**

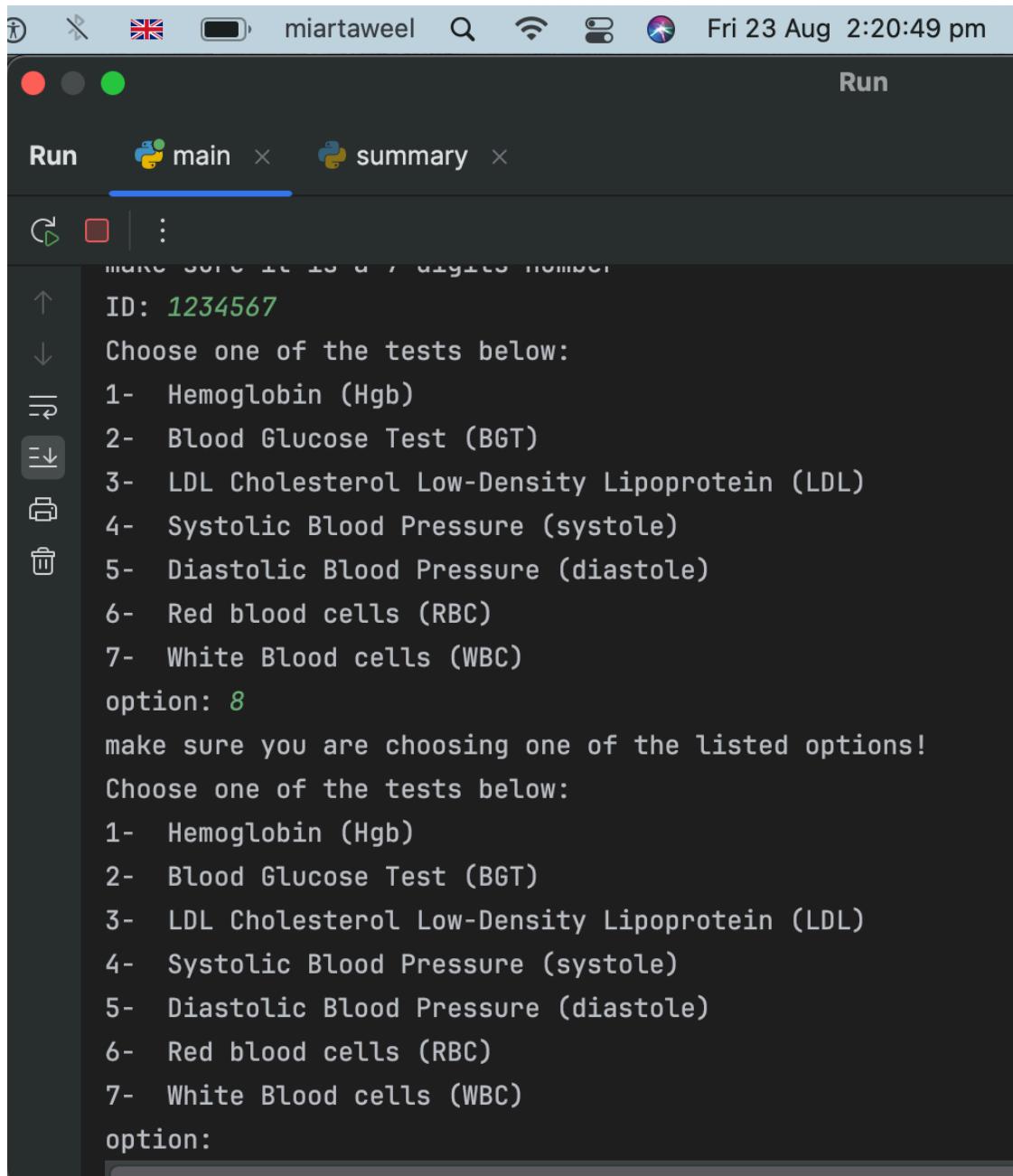


A screenshot of a terminal window titled 'Run' showing a Python script execution. The window title bar includes icons for Tools, VCS, Window, Help, and system status. The terminal shows the following interaction:

```
↑ Enter the value of the test: 12
↓ Choose the test's status
↓ 1- Pending
↓ 2- Completed
↓ 3- Reviewed
option: 4
invalid option
Choose the test's status
1- Pending
2- Completed
3- Reviewed
option: 3
```

The new record: 1234567: Hgb, 2024-08-23 07:00, 12.0, g/dL, reviewed ,is inserted successfully

- Test Type:



The screenshot shows a macOS terminal window titled "Run" with two tabs: "main" and "summary". The "main" tab is active. The terminal output is as follows:

```

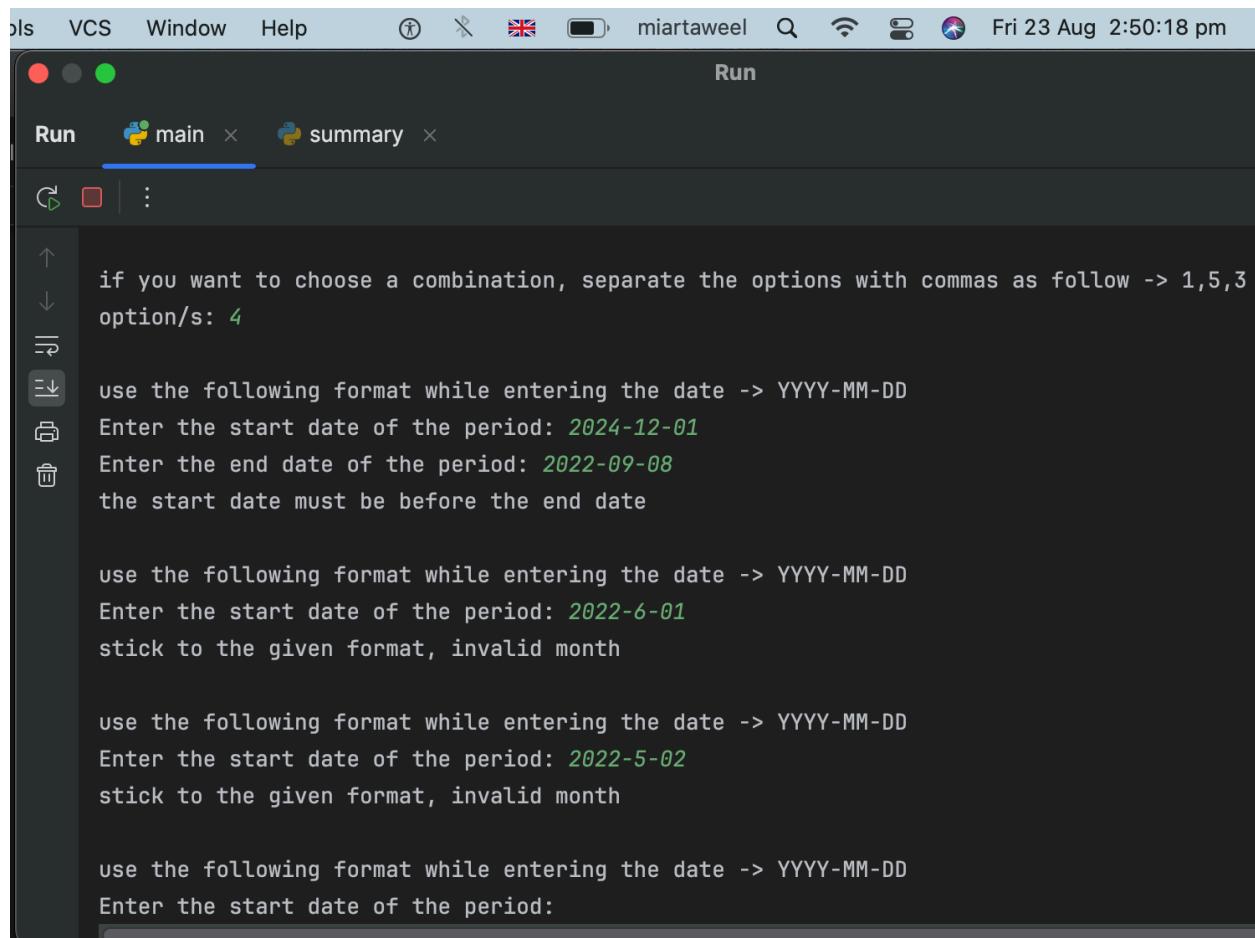
ID: 1234567
Choose one of the tests below:
1- Hemoglobin (Hgb)
2- Blood Glucose Test (BGT)
3- LDL Cholesterol Low-Density Lipoprotein (LDL)
4- Systolic Blood Pressure (systole)
5- Diastolic Blood Pressure (diastole)
6- Red blood cells (RBC)
7- White Blood cells (WBC)
option: 8
make sure you are choosing one of the listed options!
Choose one of the tests below:
1- Hemoglobin (Hgb)
2- Blood Glucose Test (BGT)
3- LDL Cholesterol Low-Density Lipoprotein (LDL)
4- Systolic Blood Pressure (systole)
5- Diastolic Blood Pressure (diastole)
6- Red blood cells (RBC)
7- White Blood cells (WBC)
option:

```

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:



The screenshot shows the PyCharm IDE's Run interface. The top bar includes tabs for 'Run', 'main' (selected), and 'summary'. The main window displays a terminal-like output of Python code. The code prompts the user to enter start and end dates for a period. It includes validation logic to ensure the start date is before the end date and follows the YYYY-MM-DD format.

```
if you want to choose a combination, separate the options with commas as follow -> 1,5,3
option/s: 4

use the following format while entering the date -> YYYY-MM-DD
Enter the start date of the period: 2024-12-01
Enter the end date of the period: 2022-09-08
the start date must be before the end date

use the following format while entering the date -> YYYY-MM-DD
Enter the start date of the period: 2022-6-01
stick to the given format, invalid month

use the following format while entering the date -> YYYY-MM-DD
Enter the start date of the period: 2022-5-02
stick to the given format, invalid month

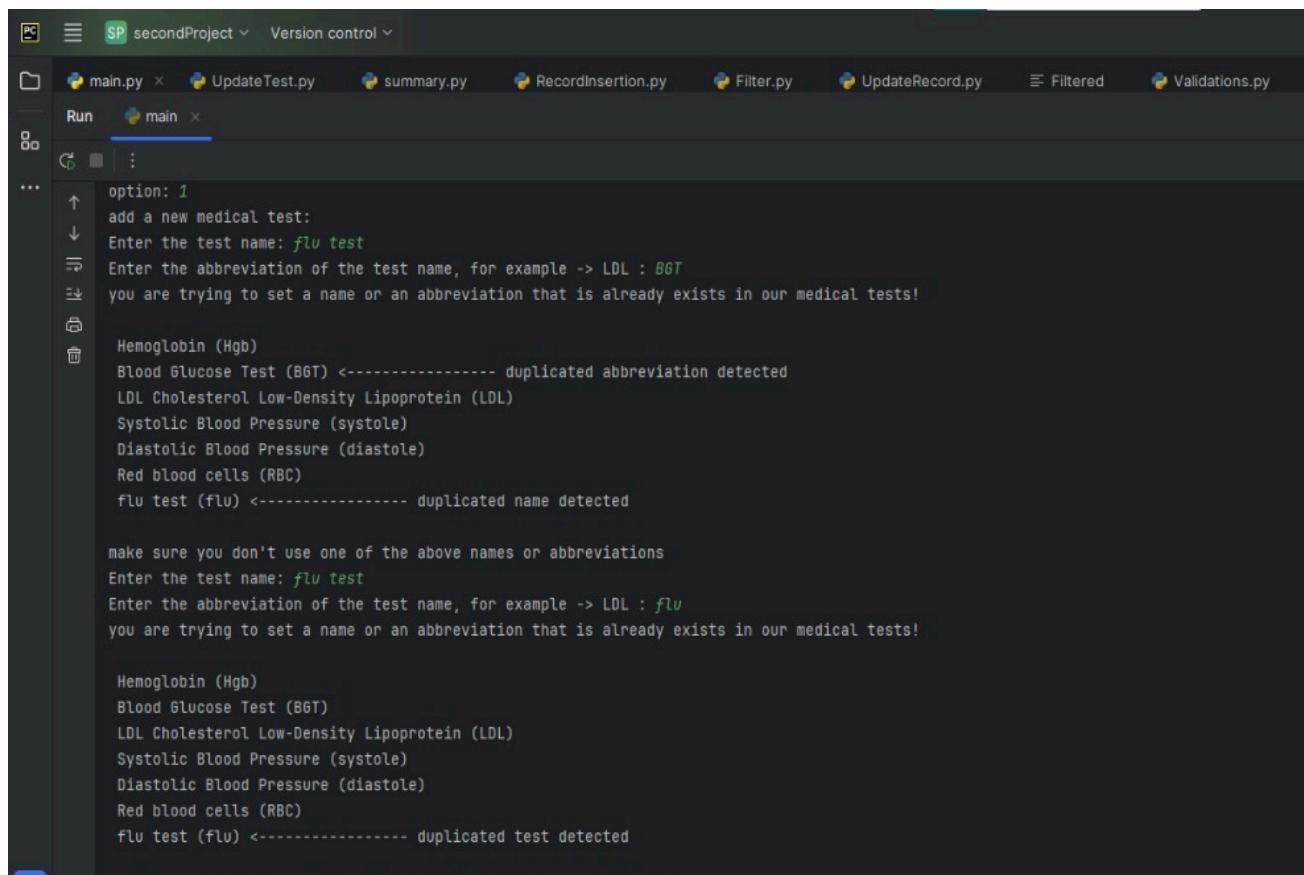
use the following format while entering the date -> YYYY-MM-DD
Enter the start date of the period:
```

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:**



```
SP secondProject Version control
main.py UpdateTest.py summary.py RecordInsertion.py Filter.py UpdateRecord.py Filtered Validations.py
Run main

option: 1
add a new medical test:
Enter the test name: flu test
Enter the abbreviation of the test name, for example -> LDL : BGT
you are trying to set a name or an abbreviation that is already exists in our medical tests!

Hemoglobin (Hgb)
Blood Glucose Test (BGT) <----- duplicated abbreviation detected
LDL Cholesterol Low-Density Lipoprotein (LDL)
Systolic Blood Pressure (systole)
Diastolic Blood Pressure (diastole)
Red blood cells (RBC)
flu test (flu) <----- duplicated name detected

make sure you don't use one of the above names or abbreviations
Enter the test name: flu test
Enter the abbreviation of the test name, for example -> LDL : flu
you are trying to set a name or an abbreviation that is already exists in our medical tests!

Hemoglobin (Hgb)
Blood Glucose Test (BGT)
LDL Cholesterol Low-Density Lipoprotein (LDL)
Systolic Blood Pressure (systole)
Diastolic Blood Pressure (diastole)
Red blood cells (RBC)
flu test (flu) <----- duplicated test detected
```

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:

```

C:\Users\EASY LIFE\AppData\Local\Microsoft\WindowsApps\python3.11.exe "D:\BZU\BZU- 3.4 SUMMER\Linux Lab\secondProject\Project2_main\main.py"
Welcome to our medical system
choose an option:
1- Add a new medical test
2- Add a new medical test record
3- Update patient records
4- Update a medical test
5- Filter medical tests
6- Summary reports
7- Export medical records to a CSV file
8- Import medical records from a CSV file
9- Exit

option: 1
add a new medical test:
Enter the test name: Red blood cells
Enter the abbreviation of the test name, for example -> LDL : rrr
you are trying to set a name or an abbreviation that is already exists in our medical tests!

Hemoglobin (Hgb)
Blood Glucose Test (BGT)
LDL Cholesterol Low-Density Lipoprotein (LDL)
Systolic Blood Pressure (systole)
Diastolic Blood Pressure (diastole)
Red blood cells (RBC) <----- duplicated name detected
flu test (flu)

make sure you don't use one of the above names or abbreviations
Enter the test name: newTest
Enter the abbreviation of the test name, for example -> LDL : newAbbr

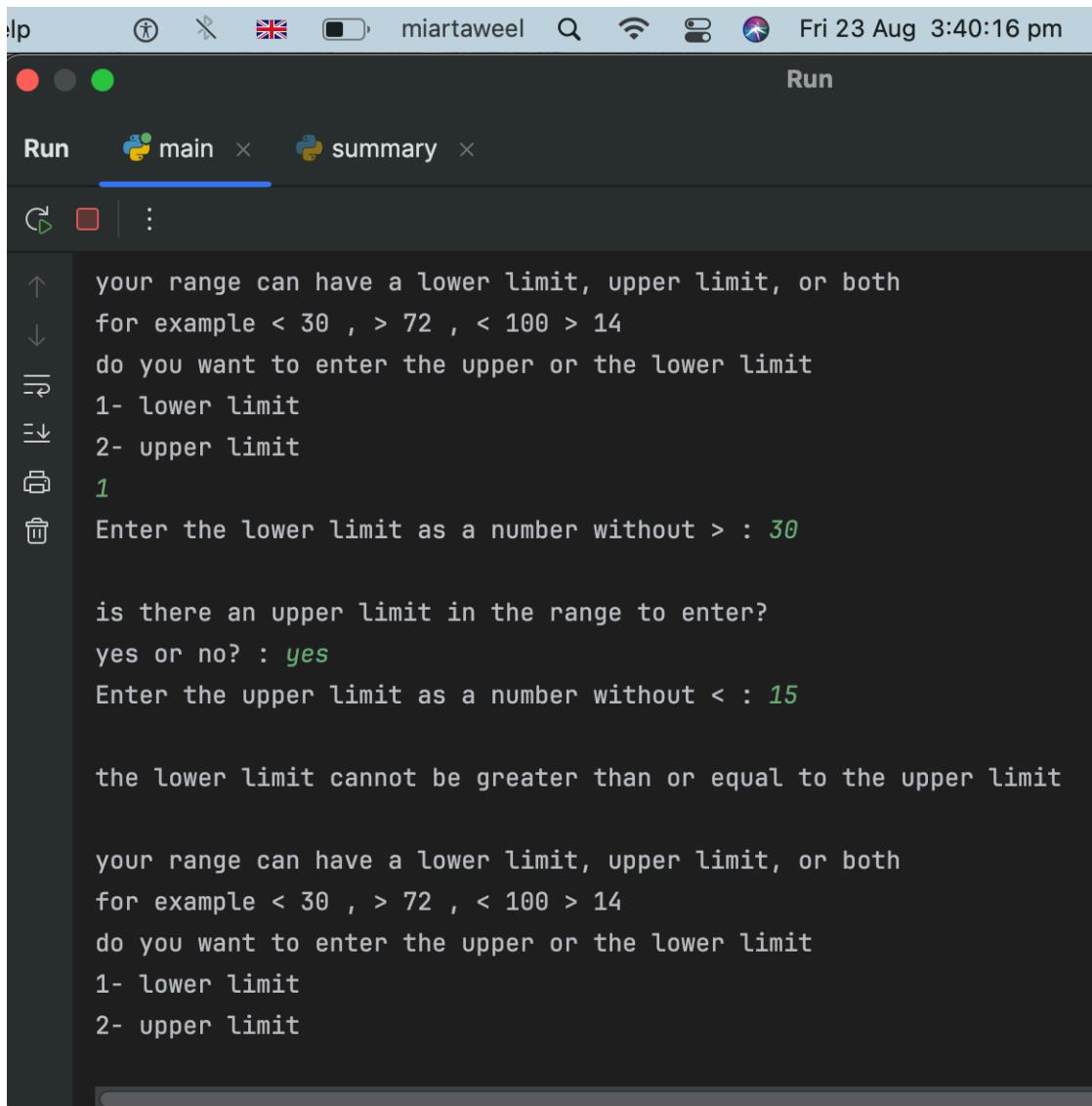
your range can have a lower limit, upper limit, or both
for example < 30 , > 72 , < 100 > 14
do you want to enter the upper or the lower limit
1- lower limit
2- upper limit
|

```

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:



The screenshot shows a terminal window with a dark theme. At the top, there are system icons for battery, signal, and time (Fri 23 Aug 3:40:16 pm). Below the title bar, there are tabs for "Run", "main" (selected), and "summary". The main area contains the following text:

```
your range can have a lower limit, upper limit, or both
for example < 30 , > 72 , < 100 > 14
do you want to enter the upper or the lower limit
1- lower limit
2- upper limit
1
Enter the lower limit as a number without > : 30

is there an upper limit in the range to enter?
yes or no? : yes
Enter the upper limit as a number without < : 15

the lower limit cannot be greater than or equal to the upper limit

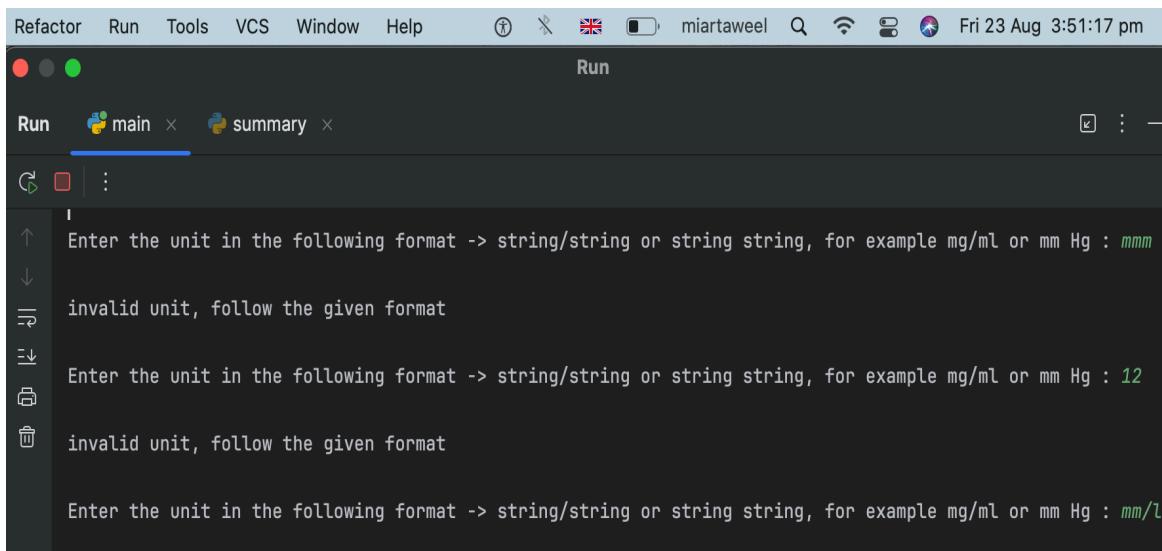
your range can have a lower limit, upper limit, or both
for example < 30 , > 72 , < 100 > 14
do you want to enter the upper or the lower limit
1- lower limit
2- upper limit
```

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:

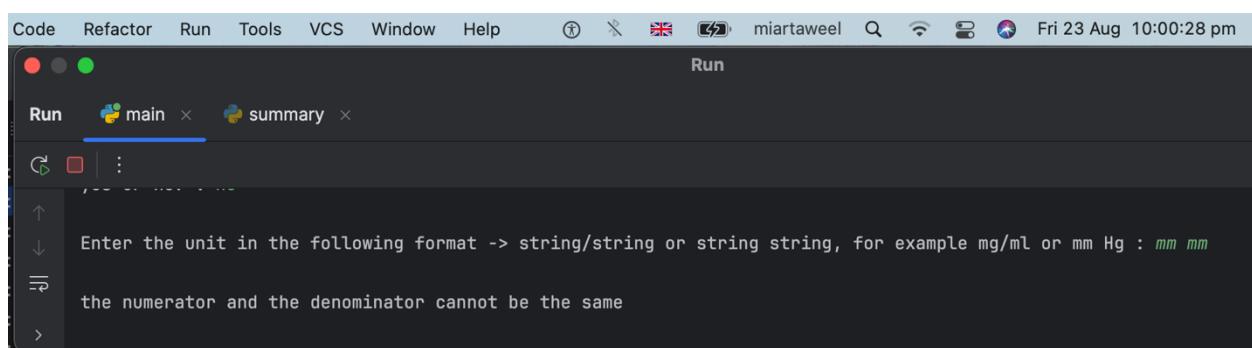
When inputting a unit to insert into a new medical test or when updating a test, the unit should be checked whether it's a valid unit written in the “char/char” or “char char” format, the following image represents the checking of its validity :

- **Invalid format:**



```
Refactor Run Tools VCS Window Help ⓘ ✎ miartaweeel Fri 23 Aug 3:51:17 pm
Run
Run main summary
Enter the unit in the following format -> string/string or string string, for example mg/ml or mm Hg : mm
invalid unit, follow the given format
Enter the unit in the following format -> string/string or string string, for example mg/ml or mm Hg : 12
invalid unit, follow the given format
Enter the unit in the following format -> string/string or string string, for example mg/ml or mm Hg : mm/L
```

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



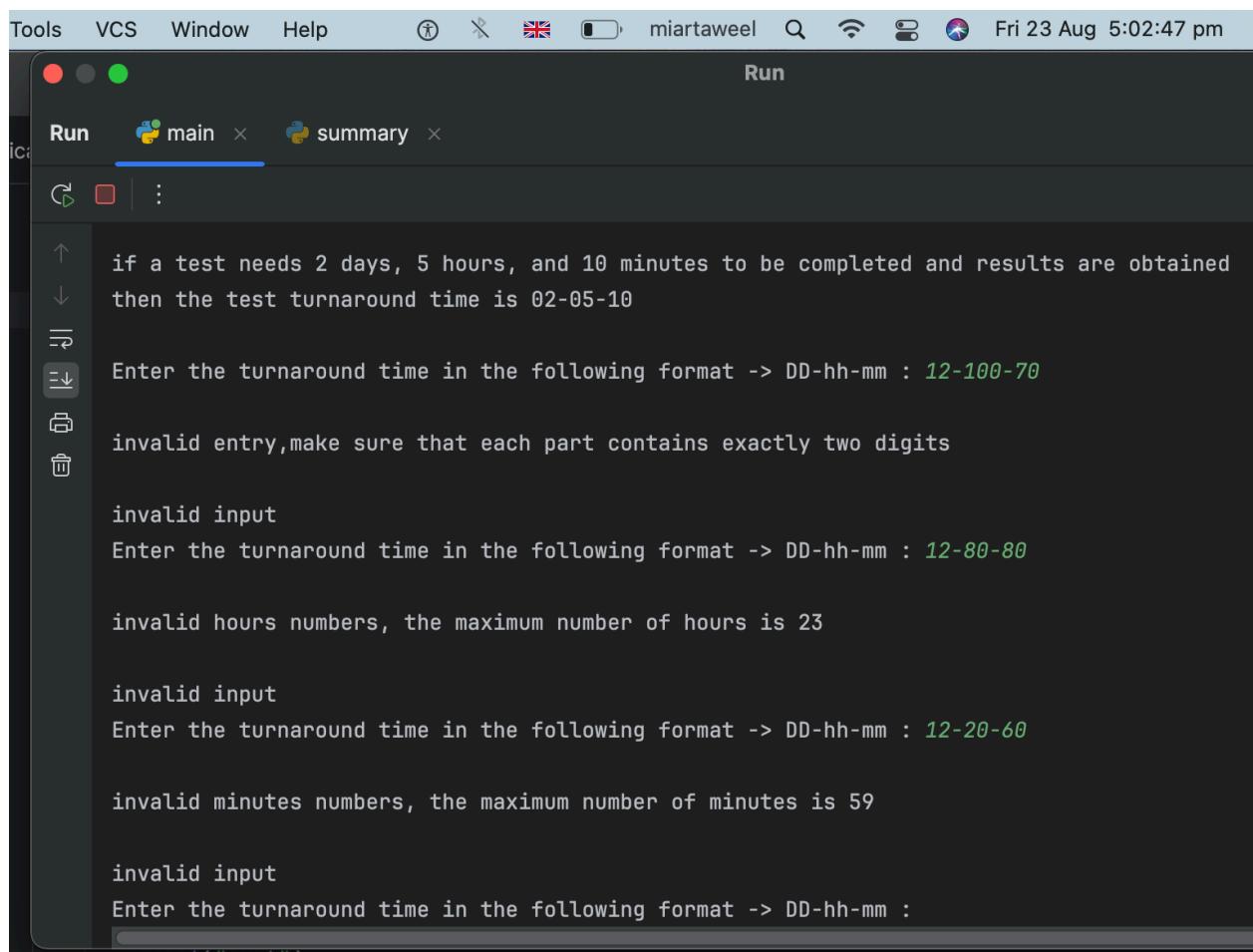
```
Code Refactor Run Tools VCS Window Help ⓘ ✎ miartaweeel Fri 23 Aug 10:00:28 pm
Run
Run main summary
Enter the unit in the following format -> string/string or string string, for example mg/ml or mm Hg : mm mm
the numerator and the denominator cannot be the same
```

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 it 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 :



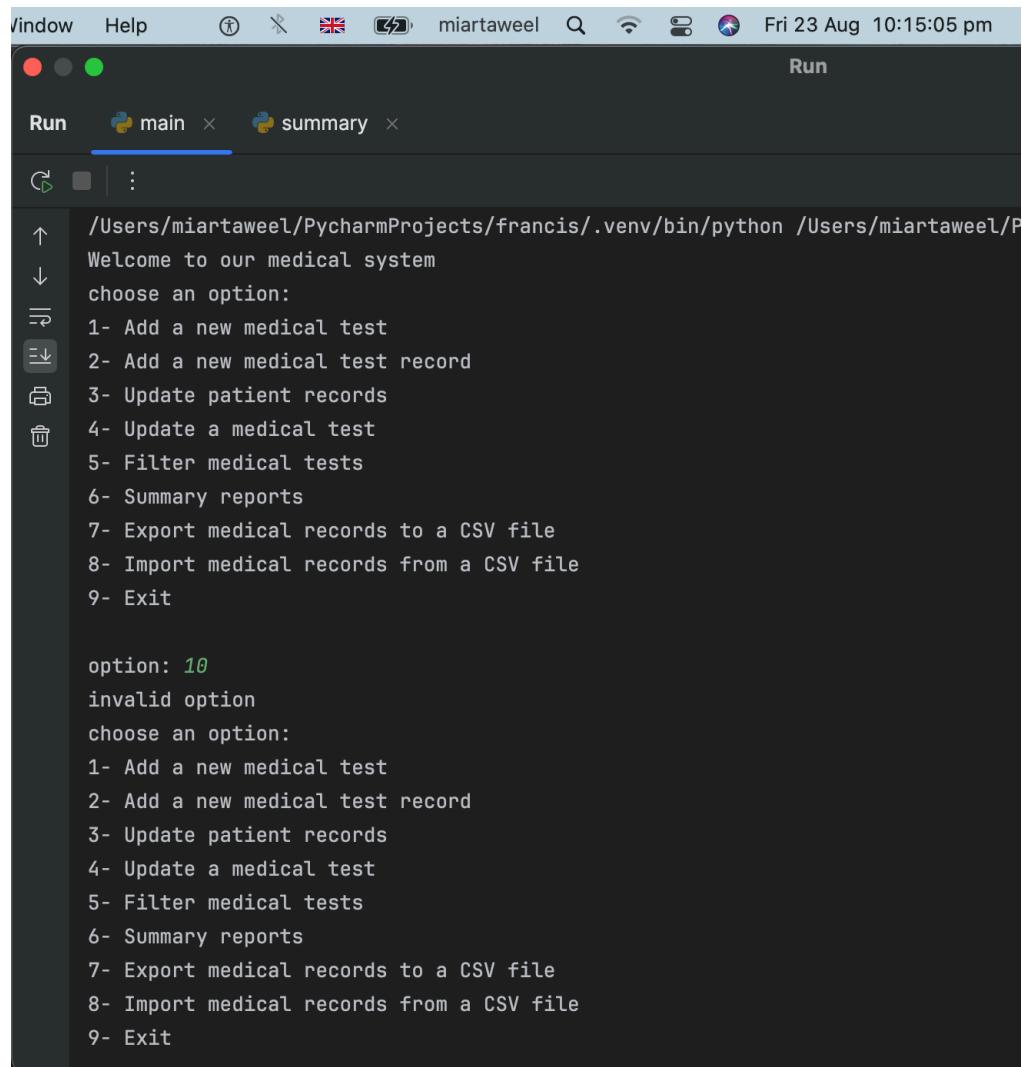
The screenshot shows a terminal window with a dark theme. At the top, there is a menu bar with 'Tools', 'VCS', 'Window', 'Help', and various system icons. The title bar says 'Run'. Below the title bar, there are two tabs: 'main' and 'summary', with 'main' being active. The main area of the terminal contains the following Python code and its execution output:

```
if a test needs 2 days, 5 hours, and 10 minutes to be completed and results are obtained  
then the test turnaround time is 02-05-10  
  
Enter the turnaround time in the following format -> DD-hh-mm : 12-100-70  
  
invalid entry,make sure that each part contains exactly two digits  
  
invalid input  
Enter the turnaround time in the following format -> DD-hh-mm : 12-80-80  
  
invalid hours numbers, the maximum number of hours is 23  
  
invalid input  
Enter the turnaround time in the following format -> DD-hh-mm : 12-20-60  
  
invalid minutes numbers, the maximum number of minutes is 59  
  
invalid input  
Enter the turnaround time in the following format -> DD-hh-mm :
```

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 it 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 :



The screenshot shows a terminal window titled 'Run' with two tabs: 'main' and 'summary'. The 'main' tab is active. The window title bar includes 'Window', 'Help', and other system icons. The date and time 'Fri 23 Aug 10:15:05 pm' are also visible. The terminal output is as follows:

```
/Users/miartaweel/PycharmProjects/francis/.venv/bin/python /Users/miartaweel/P
Welcome to our medical system
choose an option:
1- Add a new medical test
2- Add a new medical test record
3- Update patient records
4- Update a medical test
5- Filter medical tests
6- Summary reports
7- Export medical records to a CSV file
8- Import medical records from a CSV file
9- Exit

option: 10
invalid option
choose an option:
1- Add a new medical test
2- Add a new medical test record
3- Update patient records
4- Update a medical test
5- Filter medical tests
6- Summary reports
7- Export medical records to a CSV file
8- Import medical records from a CSV file
9- Exit
```

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:

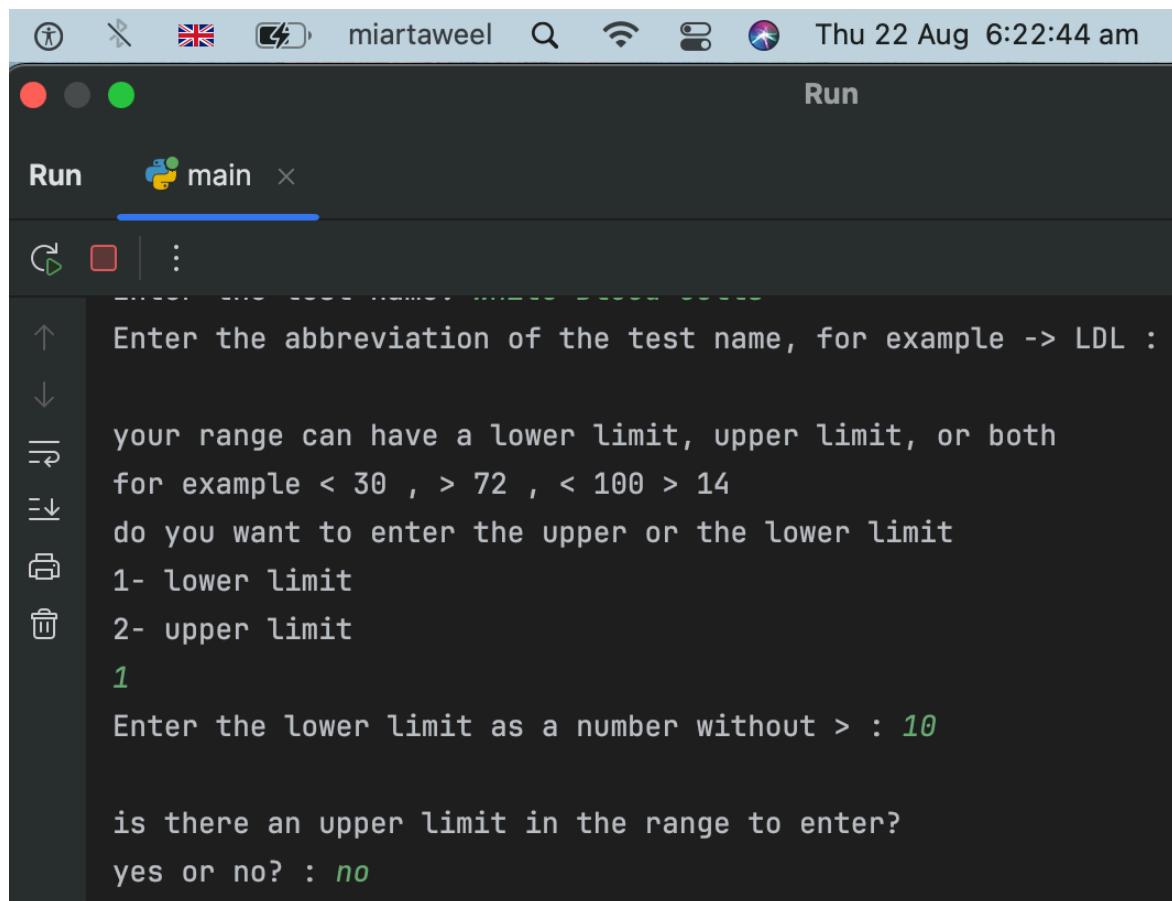
```
miartaweil Thu 22 Aug 6:17:33 am
Run main
>Welcome to our medical system
choose an option:
1- Add a new medical test
2- Add a new medical test record
3- Update patient records
4- Update a medical test
5- Filter medical tests
6- Summary reports
7- Export medical records to a CSV file
8- Import medical records from a CSV file
9- Exit

option: 1
add a new medical test:
```

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

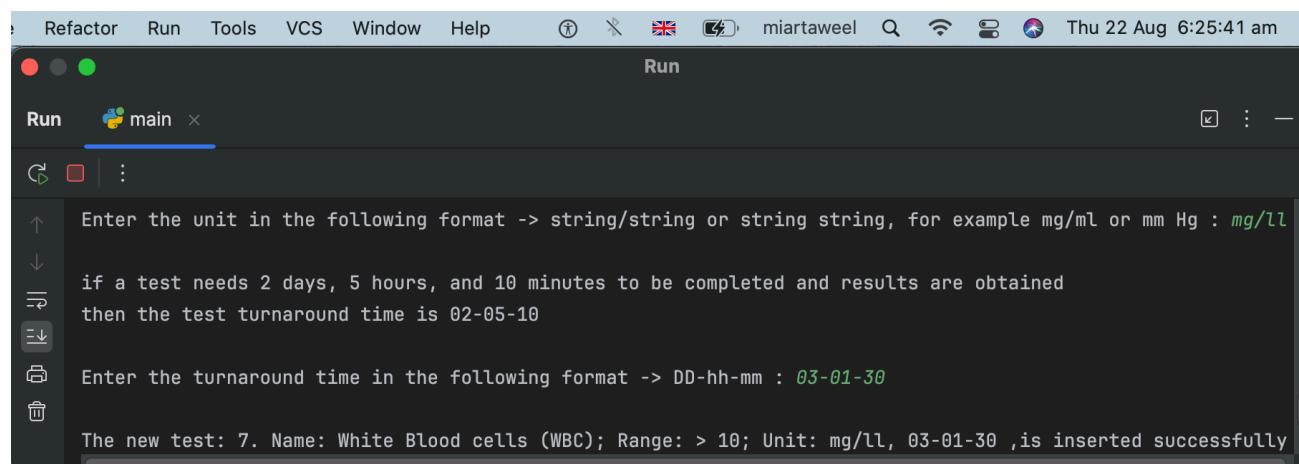
```
miartaweil Thu 22 Aug 6:19:29 am
Run main
option: 1
add a new medical test:
Enter the test name: White Blood cells
Enter the abbreviation of the test name, for example -> LDL : WBC
```

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



```
Run main ×
Run | : 
Enter the abbreviation of the test name, for example -> LDL : 
your range can have a lower limit, upper limit, or both
for example < 30 , > 72 , < 100 > 14
do you want to enter the upper or the lower limit
1- lower limit
2- upper limit
1
Enter the lower limit as a number without > : 10
is there an upper limit in the range to enter?
yes or no? : no
```

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



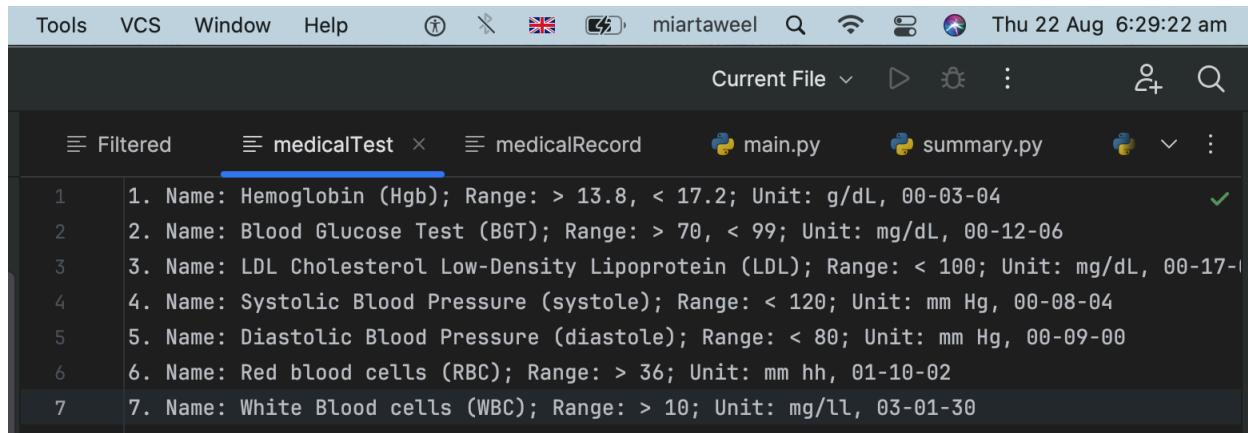
```
Refactor Run Tools VCS Window Help ⓘ ✎ miartaweil Q ⚡ Thu 22 Aug 6:25:41 am
Run main ×
Run | : 
Enter the unit in the following format -> string/string or string string, for example mg/ml or mm Hg : mg/l
if a test needs 2 days, 5 hours, and 10 minutes to be completed and results are obtained
then the test turnaround time is 02-05-10
Enter the turnaround time in the following format -> DD-hh-mm : 03-01-30
The new test: 7. Name: White Blood cells (WBC); Range: > 10; Unit: mg/l, 03-01-30 ,is inserted successfully
```

- **Output :**

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

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

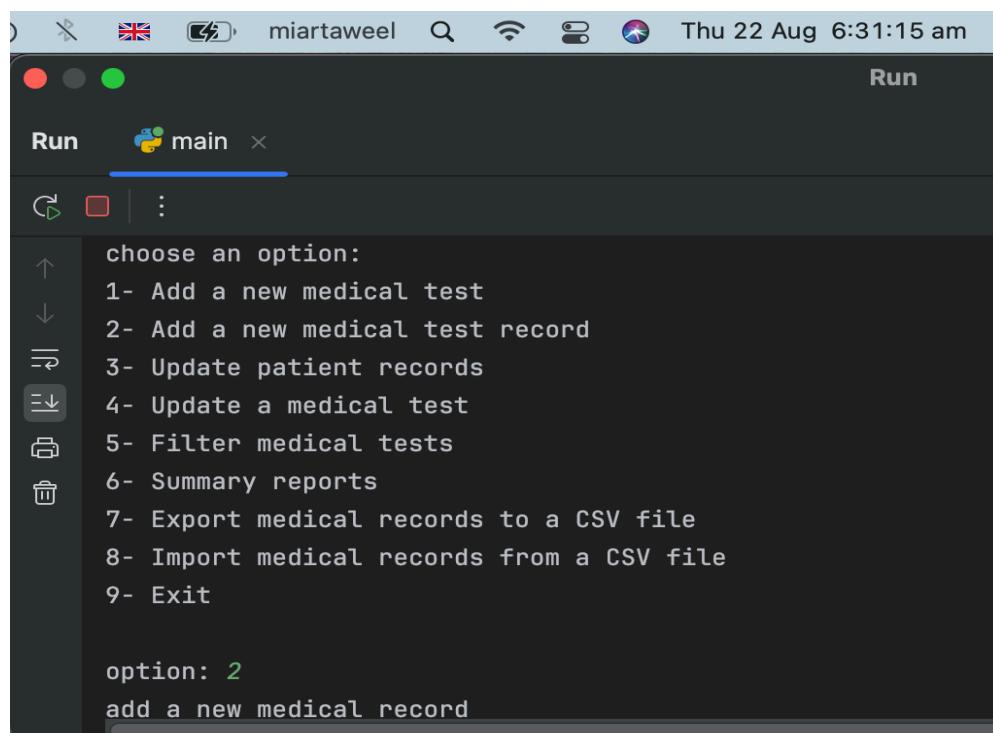
Which was in fact successfully inserted as viewed down below in the medical test file::



```
Tools VCS Window Help ⚡ 🇬🇧 miartaweeel Q WiFi Thu 22 Aug 6:29:22 am
Current File ⌂ ⌂ ⌂ main.py summary.py ⌂ ⌂ ⌂
Filtered medicalTest medicalRecord main.py summary.py ⌂ ⌂ ⌂
1. Name: Hemoglobin (Hgb); Range: > 13.8, < 17.2; Unit: g/dL, 00-03-04 ✓
2. Name: Blood Glucose Test (BGT); Range: > 70, < 99; Unit: mg/dL, 00-12-06
3. Name: LDL Cholesterol Low-Density Lipoprotein (LDL); Range: < 100; Unit: mg/dL, 00-17-01
4. Name: Systolic Blood Pressure (systole); Range: < 120; Unit: mm Hg, 00-08-04
5. Name: Diastolic Blood Pressure (diastole); Range: < 80; Unit: mm Hg, 00-09-00
6. Name: Red blood cells (RBC); Range: > 36; Unit: mm hh, 01-10-02
7. Name: White Blood cells (WBC); Range: > 10; Unit: mg/l, 03-01-30
```

### 3.2.Inserting a medical record :

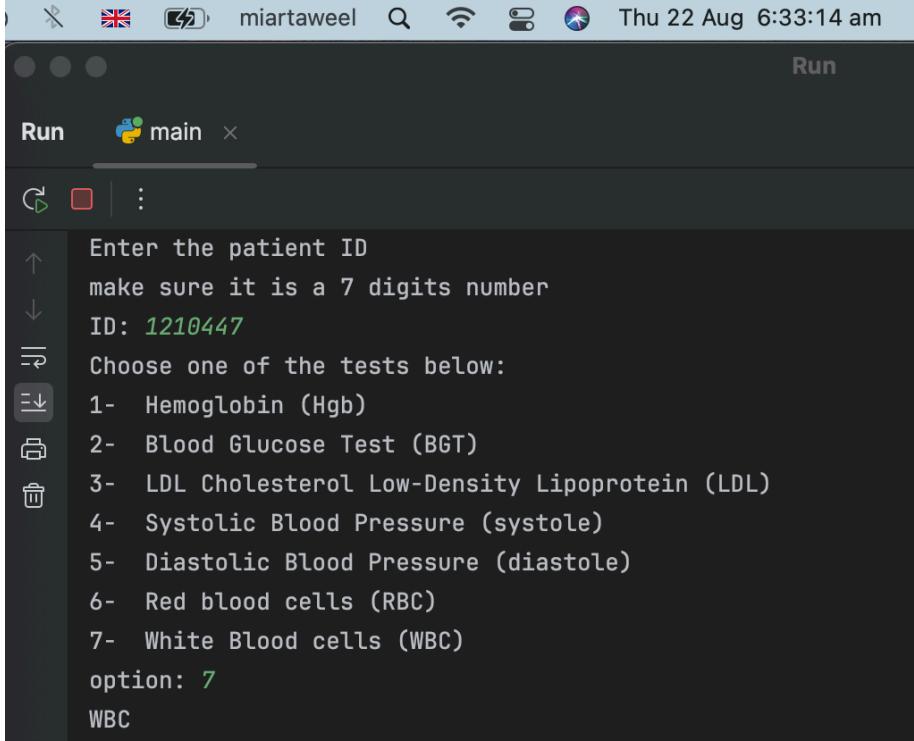
The following images represent inserting a new record into the medical record file with valid inputs , the output will be observed in the file after executing :



```
Run main
Run choose an option:
  1- Add a new medical test
  2- Add a new medical test record
  3- Update patient records
  4- Update a medical test
  5- Filter medical tests
  6- Summary reports
  7- Export medical records to a CSV file
  8- Import medical records from a CSV file
  9- Exit

option: 2
add a new medical record
```

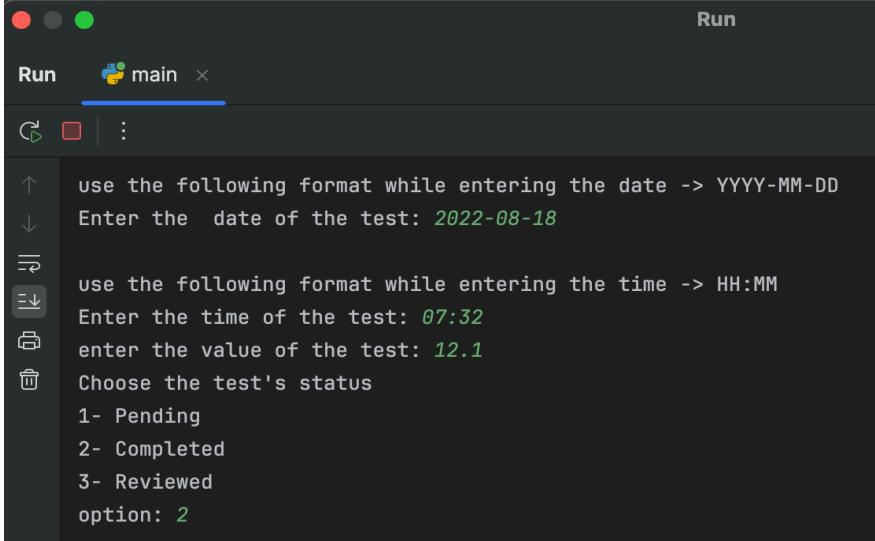
After Selecting the insert option from the option menu , a valid 7 integer number is inserted as the patient ID, in addition to selecting a valid test type from the test type menu.



```
Run main ×

Enter the patient ID
make sure it is a 7 digits number
ID: 1210447
Choose one of the tests below:
1- Hemoglobin (Hgb)
2- Blood Glucose Test (BGT)
3- LDL Cholesterol Low-Density Lipoprotein (LDL)
4- Systolic Blood Pressure (systole)
5- Diastolic Blood Pressure (diastole)
6- Red blood cells (RBC)
7- White Blood cells (WBC)
option: 7
WBC
```

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



```
Run main ×

use the following format while entering the date -> YYYY-MM-DD
Enter the date of the test: 2022-08-18

use the following format while entering the time -> HH:MM
Enter the time of the test: 07:32
enter the value of the test: 12.1
Choose the test's status
1- Pending
2- Completed
3- Reviewed
option: 2
```

**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 :

The screenshot shows a PyCharm interface with the following details:

- File Bar:** Un, Tools, VCS, Window, Help.
- User:** miartawee
- Version Control:** francis
- Current File:** medicalRecord
- Toolbars:** Current File, Find, Replace, Settings.
- Left Sidebar:** Folders, Files (main.py, summary.py, Validations.py, Recs), and a three-dot menu.
- Table View:** A list of 16 medical records. Each record includes a row number, a unique identifier, and a timestamp. The last column contains a green checkmark icon.

Index	Details	Status
1	1355555: Bgt, 2024-01-01 14:10, 44.2, mg/dL, completed, 2024-01-01 15:30	✓
2	1355555: LDL, 2024-01-01 14:10, 13.4, mg/dL, completed, 2024-01-01 15:30	
3	1300500: RBC, 2024-03-01 05:20, 13.5, mg/dL, completed, 2024-03-01 05:30	
4	1210447: BGT, 2020-04-07 13:43, 83.0, mg/dL, reviewed	
5	1212121: diastole, 2022-11-04 15:32, 12, mm Hg, completed, 2022-11-05 04:38	
6	1210447: BGT, 2020-04-07 13:43, 70.0, mg/dL, reviewed	
7	1210447: systole, 2023-07-18 18:14, 74.0, mm Hg, completed, 2023-07-19 02:18	
8	1210447: LDL, 2024-11-12 13:43, 52.5, mg/dL, pending	
9	1210447: BGT, 2012-03-08 12:12, 80.0, mg/dL, reviewed	
10	1212121: BGT, 2022-11-04 15:32, 12, mg/dL, completed, 2022-11-05 04:38	
11	1112223: diastole, 2020-02-04 14:02, 47, mg/dL, pending	
12	1210447: Hgb, 2020-03-27 21:27, 14, g/dL, completed, 2020-03-28 00:31	
13	1210447: systole, 2020-04-03 12:01, 14.2, g/dL, completed, 2020-04-03 15:05	
14	1210447: BGT, 2016-11-30 08:02, 41.2, mg/dL, completed, 2016-11-30 11:06	
15	1210447: LDL, 2016-11-30 08:02, 41.2, g/dL, completed, 2016-11-30 11:06	
16	1210447: WBC, 2022-08-18 07:32, 12.1, mg/ll, completed, 2022-08-21 09:03	

### 3.3. Update Record:

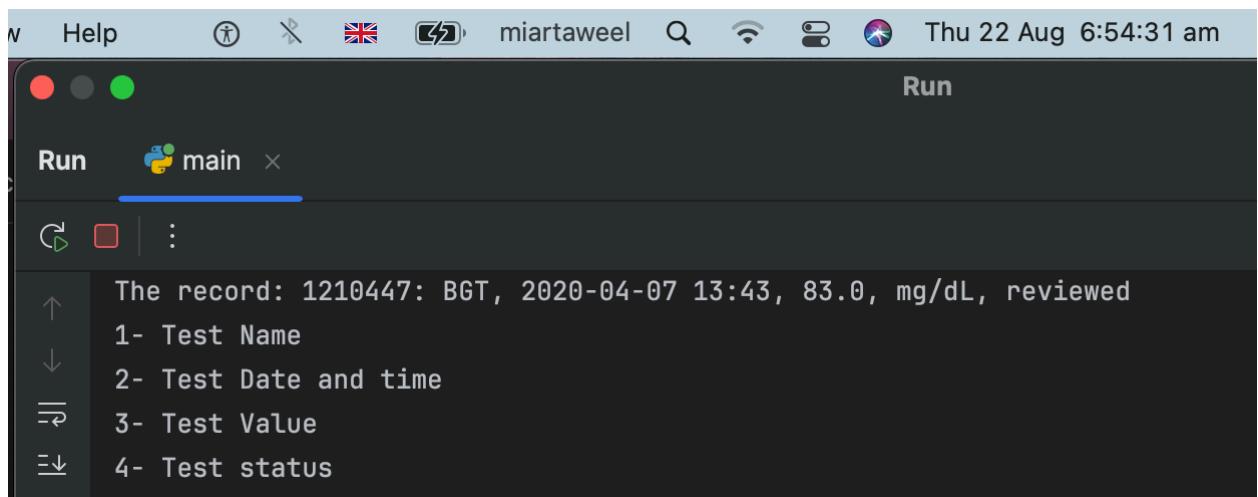
The following images represents updating all the fields of a specific test record based on choice as viewed down below :

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

```
choose an option:  
1- Add a new medical test  
2- Add a new medical test record  
3- Update patient records  
4- Update a medical test  
5- Filter medical tests |  
6- Summary reports  
7- Export medical records to a CSV file  
8- Import medical records from a CSV file  
9- Exit  
  
option: 3  
update patient records  
Enter the patient ID  
make sure it is a 7 digits number  
ID: 1210447  
1- 1210447: BGT, 2020-04-07 13:43, 83.0, mg/dL, reviewed  
2- 1210447: BGT, 2020-04-07 13:43, 70.0, mg/dL, reviewed  
3- 1210447: systole, 2023-07-18 18:14, 74.0, mm Hg, completed, 2023-07-19 02:18  
4- 1210447: LDL, 2024-11-12 13:43, 52.5, mg/dL, pending  
5- 1210447: BGT, 2012-03-08 12:12, 80.0, mg/dL, reviewed  
6- 1210447: Hgb, 2020-03-27 21:27, 14, g/dL, completed, 2020-03-28 00:31
```

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 .

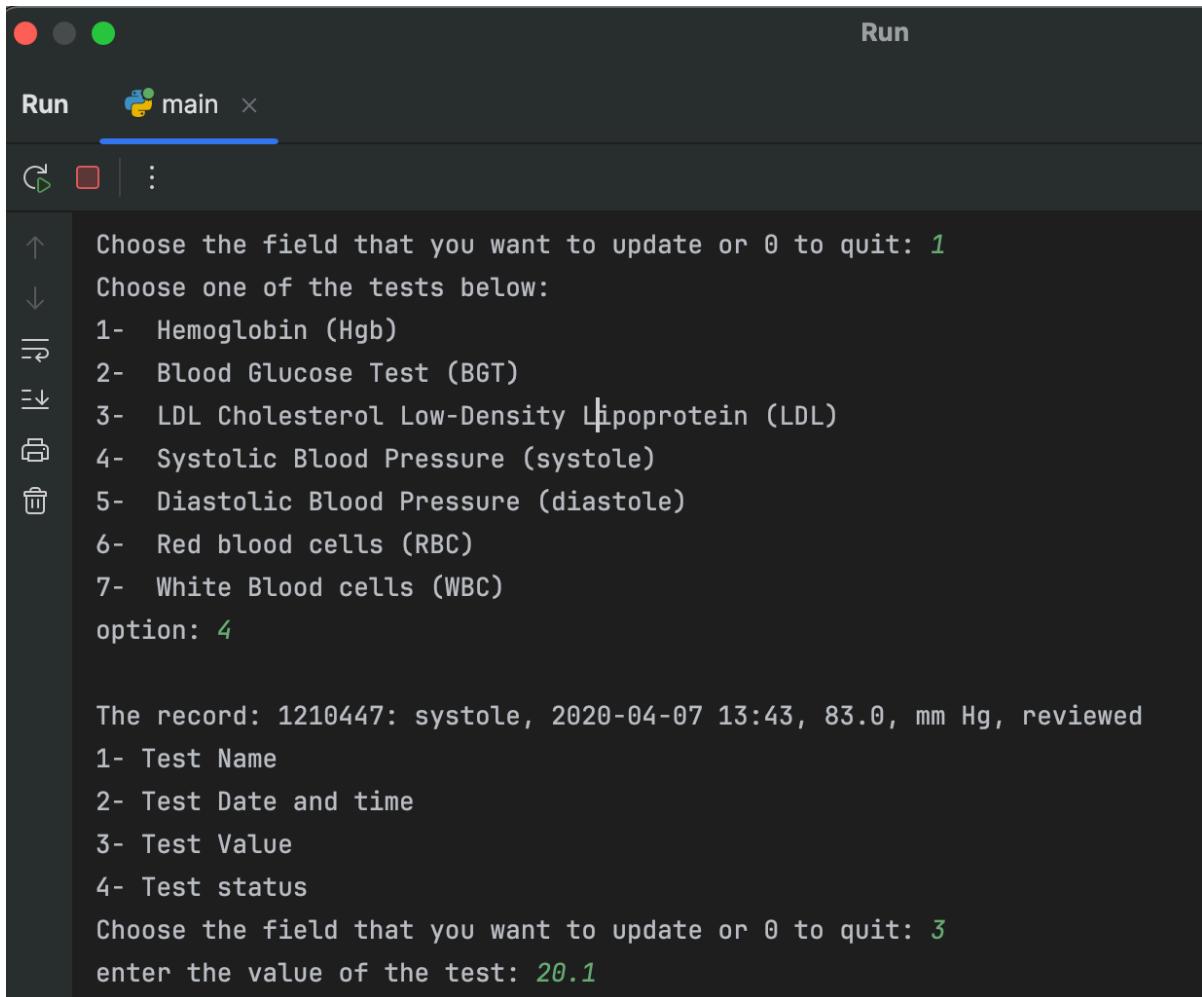
After a record is chosen ,a menu is shown to let the user choose which aspect they should update :



The terminal window shows the following menu:

```
The record: 1210447: BGT, 2020-04-07 13:43, 83.0, mg/dL, reviewed
1- Test Name
2- Test Date and time
3- Test Value
4- Test status
```

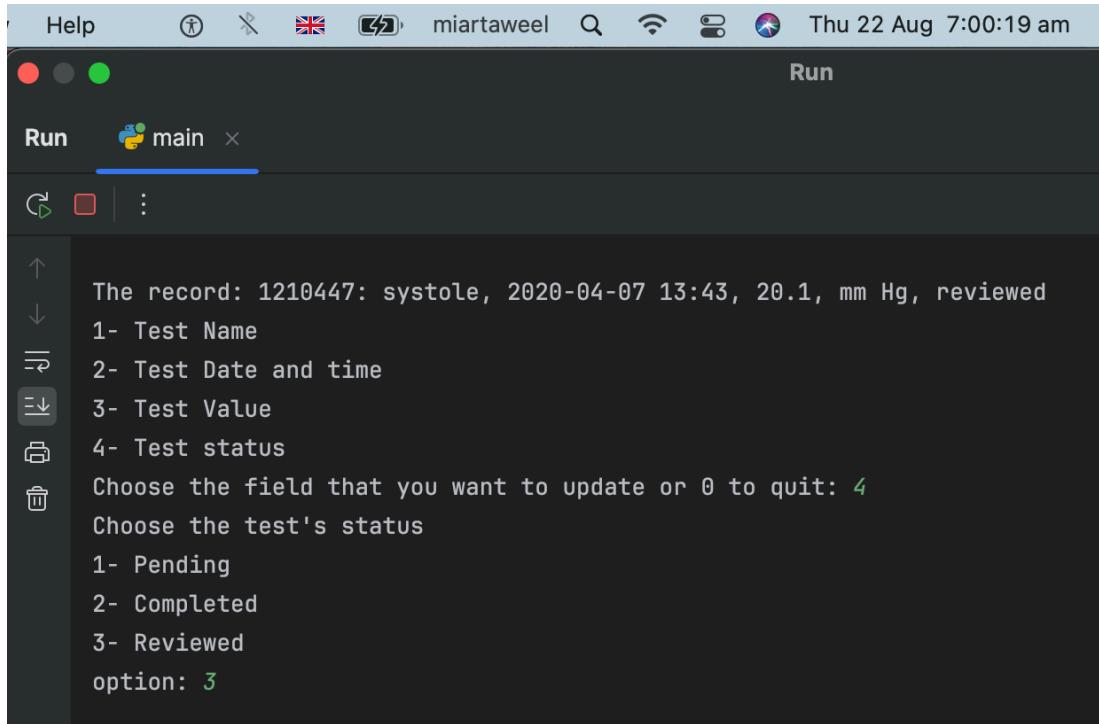
After choosing which aspect to update which in this case is the test type ,a new test type is chosen from its menu to update it , which then proceeds to replace the old value successfully as viewed down below:



```
Choose the field that you want to update or 0 to quit: 1
Choose one of the tests below:
1- Hemoglobin (Hgb)
2- Blood Glucose Test (BGT)
3- LDL Cholesterol Low-Density Lipoprotein (LDL)
4- Systolic Blood Pressure (systole)
5- Diastolic Blood Pressure (diastole)
6- Red blood cells (RBC)
7- White Blood cells (WBC)
option: 4

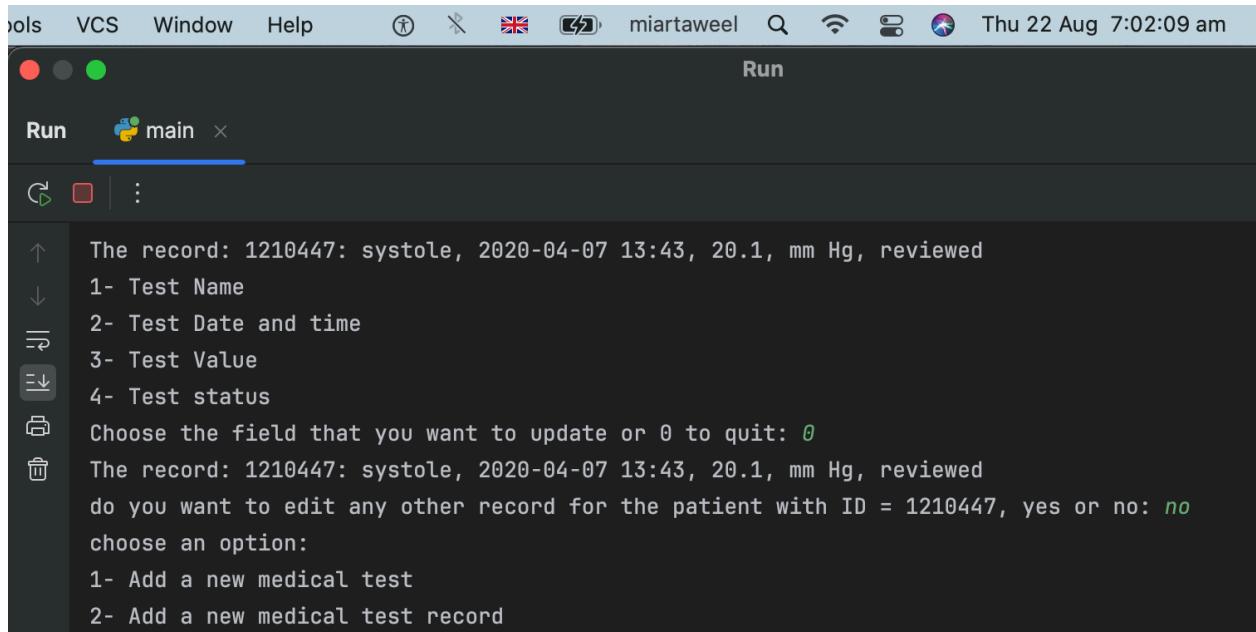
The record: 1210447: systole, 2020-04-07 13:43, 83.0, mm Hg, reviewed
1- Test Name
2- Test Date and time
3- Test Value
4- Test status
Choose the field that you want to update or 0 to quit: 3
enter the value of the test: 20.1
```

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



The record: 1210447: systole, 2020-04-07 13:43, 20.1, mm Hg, reviewed  
1- Test Name  
2- Test Date and time  
3- Test Value  
4- Test status  
Choose the field that you want to update or 0 to quit: 4  
Choose the test's status  
1- Pending  
2- Completed  
3- Reviewed  
option: 3

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 :



The record: 1210447: systole, 2020-04-07 13:43, 20.1, mm Hg, reviewed  
1- Test Name  
2- Test Date and time  
3- Test Value  
4- Test status  
Choose the field that you want to update or 0 to quit: 0  
The record: 1210447: systole, 2020-04-07 13:43, 20.1, mm Hg, reviewed  
do you want to edit any other record for the patient with ID = 1210447, yes or no: no  
choose an option:  
1- Add a new medical test  
2- Add a new medical test record

### 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 :

The screenshot shows a terminal window titled "secondProject" with several Python files listed in the sidebar: main.py, UpdateTest.py, summary.py, RecordInsertion.py, Filter.py, UpdateRecord.py, Filtered, Validations.py, medicalRecord, and medicalTest. The main area of the terminal displays the following text:

```
*C:\Users\EASY LIFE\AppData\Local\Microsoft\WindowsApps\python3.11.exe* D:\BZU\BZU- 3.4 SUMMER\Linux Lab\secondProject\Project2_main\main.py*
Welcome to our medical system
choose an option:
1- Add a new medical test
2- Add a new medical test record
3- Update patient records
4- Update a medical test
5- Filter medical tests
6- Summary reports
7- Export medical records to a CSV file
8- Import medical records from a CSV file
9- Exit

option: 4
update a medical test
Choose one of the tests below:
1- Hemoglobin (Hgb)
2- Blood Glucose Test (BGT)
3- LDL Cholesterol Low-Density Lipoprotein (LDL)
4- Systolic Blood Pressure (systole)
5- Diastolic Blood Pressure (diastole)
6- Red blood cells (RBC)
7- flu test (flu)
option: 7
1- Test Name
2- Test normal Range
3- Test Unit
4- Test turnaround
Choose the field that you want to update or 0 to quit: 1
the current test name is : flu test (flu);

Enter the test name: updated test
Enter the abbreviation of the test name, for example -> LDL : upT
hi

the new test name is : updated test (upT);

The updated test is now 7. Name: updated test (upT); Range: > 30, < 85; Unit: jj/jk, 02-08-10
```

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:

```

secondProject Version control
main.py UpdateTest.py summary.py RecordInsertion.py Filter.py UpdateRecord.py Filtered Validations.py medicalRecord medicalTest

Run main main

9- Exit

option: 4
update a medical test
Choose one of the tests below:
1- Hemoglobin (Hgb)
2- Blood Glucose Test (BGT)
3- LDL Cholesterol Low-Density Lipoprotein (LDL)
4- Systolic Blood Pressure (systole)
5- Diastolic Blood Pressure (diastole)
6- Red blood cells (RBC)
7- updated test (upT)
option: 7
1- Test Name
2- Test normal Range
3- Test Unit
4- Test turnaround
Choose the field that you want to update or 0 to quit: 2
the current test normal range is : > 30, < 85;
Enter the new Range

your range can have a lower limit, upper limit, or both
for example < 30 , > 72 , < 100 > 14
do you want to enter the upper or the lower limit
1- lower limit
2- upper limit
1
Enter the lower limit as a number without > : 42

is there an upper limit in the range to enter?
yes or not : yes
Enter the upper limit as a number without < : 117

the new test normal range is : > 42, < 117;

The updated test is now 7. Name: updated test (upT); Range: > 42, < 117; Unit: jj/jk, 02-08-10

```

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:

```

Run main main

9- Exit

option: 4
update a medical test
Choose one of the tests below:
1- Hemoglobin (Hgb)
2- Blood Glucose Test (BGT)
3- LDL Cholesterol Low-Density Lipoprotein (LDL)
4- Systolic Blood Pressure (systole)
5- Diastolic Blood Pressure (diastole)
6- Red blood cells (RBC)
7- updated test (upT)
option: 7
1- Test Name
2- Test normal Range
3- Test Unit
4- Test turnaround
Choose the field that you want to update or 0 to quit: 3
the current test unit is : jj/jk,

Enter the unit in the following format -> string/string or string string, for example mg/ml or mm Hg : mm po

the new test unit is : mm po,

The updated test is now 7. Name: updated test (upT); Range: > 42, < 117; Unit: mm po, 02-08-10

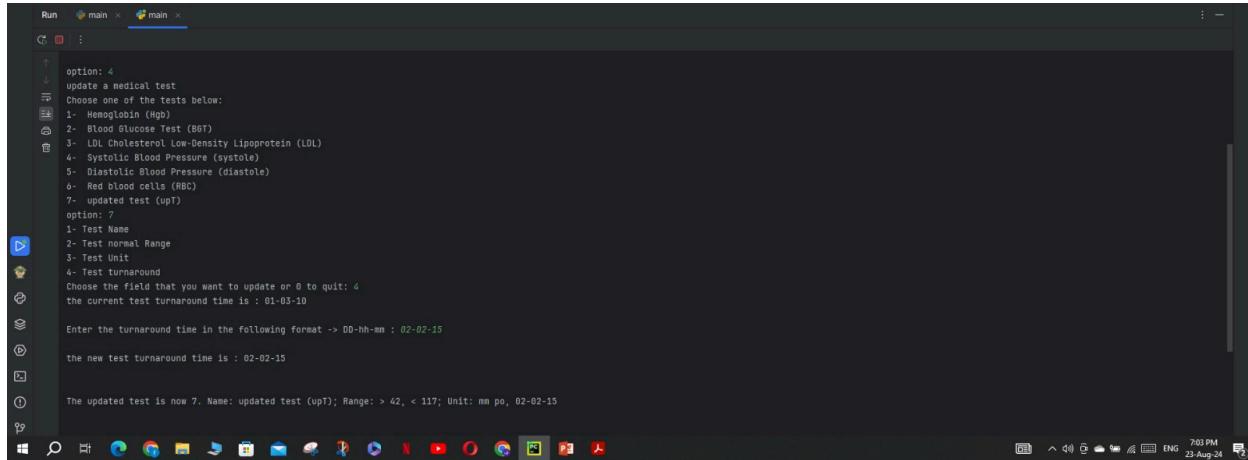
```

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:**



```
Run main x main x
option: 4
update a medical test
Choose one of the tests below:
1- Hemoglobin (Hgb)
2- Blood Glucose Test (BGT)
3- LDL Cholesterol Low-Density Lipoprotein (LDL)
4- Systolic Blood Pressure (systole)
5- Diastolic Blood Pressure (diastole)
6- Red blood cells (RBC)
7- updated test (upT)
option: 7
1- Test Name
2- Test normal Range
3- Test Unit
4- Test turnaround
Choose the field that you want to update or 0 to quit: 4
the current test turnaround time is : 01-03-10
Enter the turnaround time in the following format -> 00-hh-mm : 02-02-15
the new test turnaround time is : 02-02-15

The updated test is now 7. Name: updated test (upT); Range: > 42, < 117; Unit: mm po, 02-02-15
```

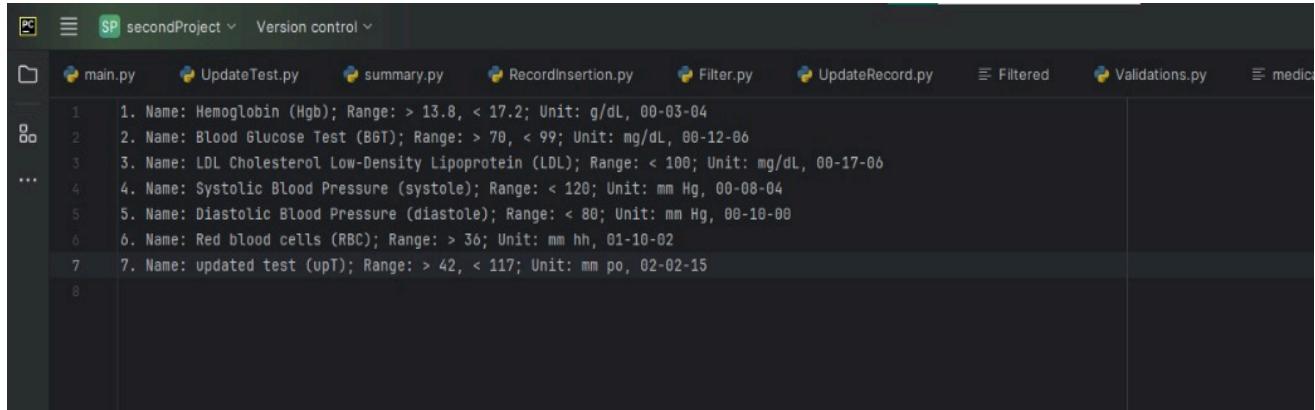
After updating the unit of that same test, based on the inputted data viewed in the previous picture , the test became :

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

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

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

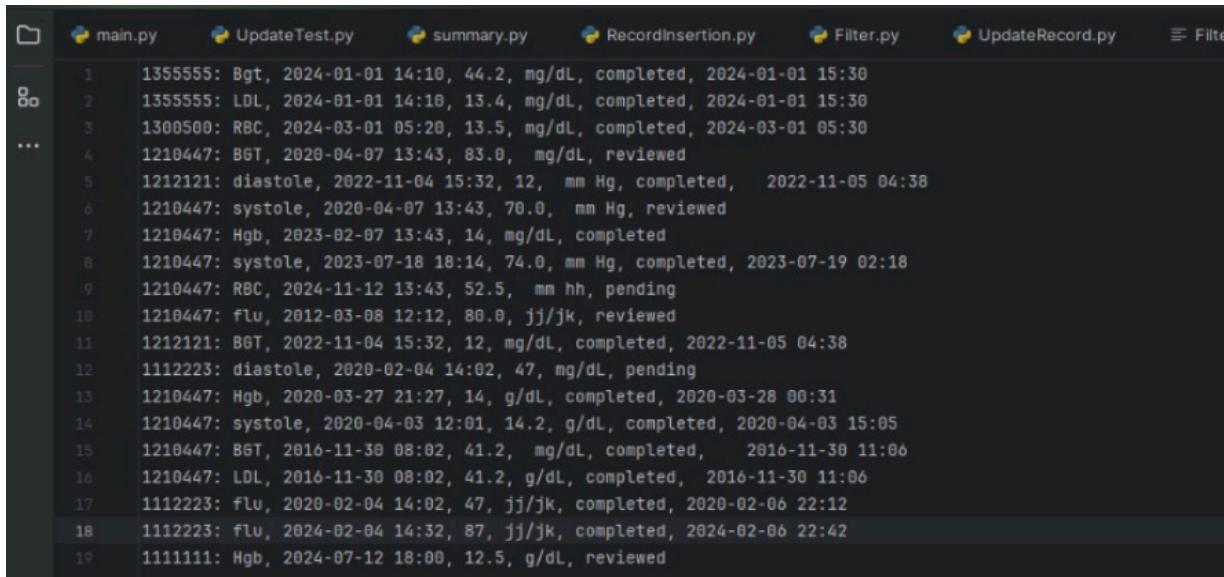
**Name: updated test (upT); Range: > 42, <117; Unit: mm po, 02-02-15**



```
SP secondProject Version control
main.py UpdateTest.py summary.py RecordInsertion.py Filter.py UpdateRecord.py Filtered Validations.py medical
1. Name: Hemoglobin (Hgb); Range: > 13.8, < 17.2; Unit: g/dL, 00-03-04
2. Name: Blood Glucose Test (BGT); Range: > 70, < 99; Unit: mg/dL, 00-12-06
3. Name: LDL Cholesterol Low-Density Lipoprotein (LDL); Range: < 100; Unit: mg/dL, 00-17-06
4. Name: Systolic Blood Pressure (systole); Range: < 120; Unit: mm Hg, 00-08-04
5. Name: Diastolic Blood Pressure (diastole); Range: < 80; Unit: mm Hg, 00-10-00
6. Name: Red blood cells (RBC); Range: > 36; Unit: mm hh, 01-10-02
7. Name: updated test (upT); Range: > 42, < 117; Unit: mm po, 02-02-15
```

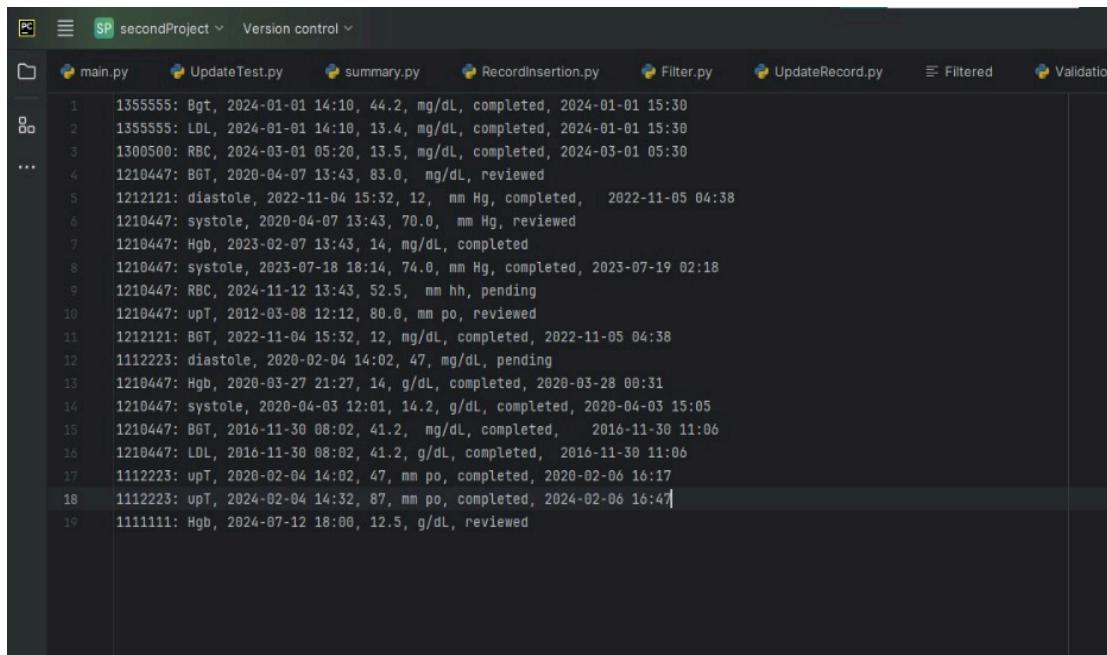
The changes were reflected onto the medical record file, and observed as viewed down below.

- **The old medical record Content:**



```
main.py UpdateTest.py summary.py RecordInsertion.py Filter.py UpdateRecord.py Filter
1 1355555: Bgt, 2024-01-01 14:10, 44.2, mg/dL, completed, 2024-01-01 15:30
2 1355555: LDL, 2024-01-01 14:10, 13.4, mg/dL, completed, 2024-01-01 15:30
3 1300500: RBC, 2024-03-01 05:20, 13.5, mg/dL, completed, 2024-03-01 05:30
...
4 1210447: BGT, 2020-04-07 13:43, 83.0, mg/dL, reviewed
5 1212121: diastole, 2022-11-04 15:32, 12, mm Hg, completed, 2022-11-05 04:38
6 1210447: systole, 2020-04-07 13:43, 70.0, mm Hg, reviewed
7 1210447: Hgb, 2023-02-07 13:43, 14, mg/dL, completed
8 1210447: systole, 2023-07-18 18:14, 74.0, mm Hg, completed, 2023-07-19 02:18
9 1210447: RBC, 2024-11-12 13:43, 52.5, mm hh, pending
10 1210447: flu, 2012-03-08 12:12, 80.0, jj/jk, reviewed
11 1212121: BGT, 2022-11-04 15:32, 12, mg/dL, completed, 2022-11-05 04:38
12 1112223: diastole, 2020-02-04 14:02, 47, mg/dL, pending
13 1210447: Hgb, 2020-03-27 21:27, 14, g/dL, completed, 2020-03-28 00:31
14 1210447: systole, 2020-04-03 12:01, 14.2, g/dL, completed, 2020-04-03 15:05
15 1210447: BGT, 2016-11-30 08:02, 41.2, mg/dL, completed, 2016-11-30 11:06
16 1210447: LDL, 2016-11-30 08:02, 41.2, g/dL, completed, 2016-11-30 11:06
17 1112223: flu, 2020-02-04 14:02, 47, jj/jk, completed, 2020-02-06 22:12
18 1112223: flu, 2024-02-04 14:32, 87, jj/jk, completed, 2024-02-06 22:42
19 1111111: Hgb, 2024-07-12 18:00, 12.5, g/dL, reviewed
```

- **The new medical record Content:**



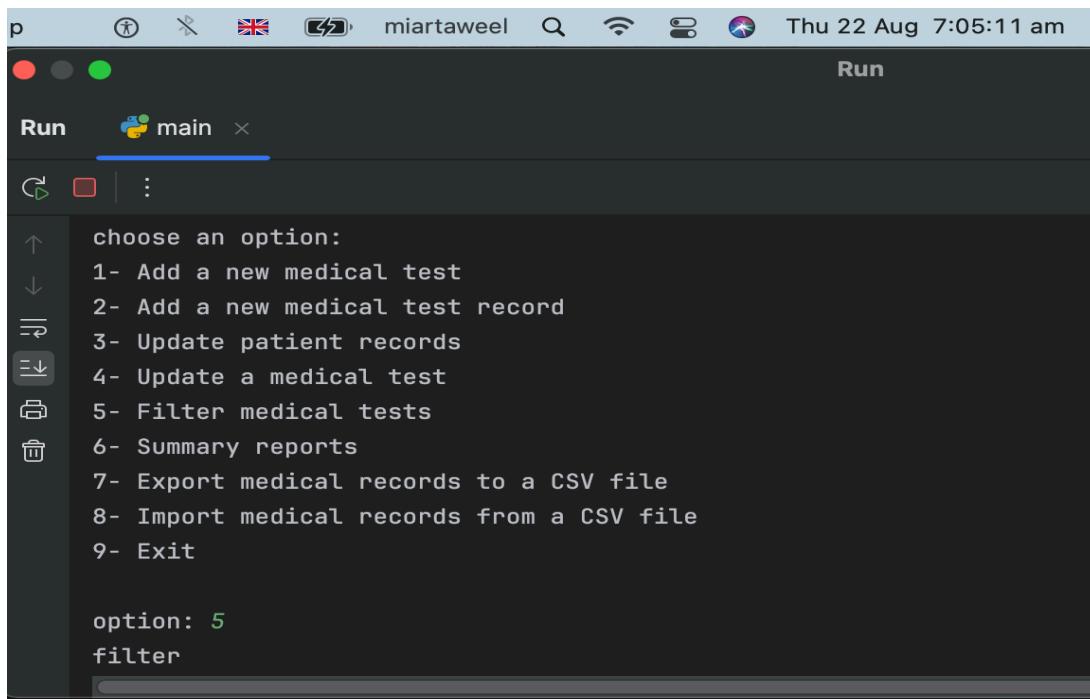
```
SP secondProject Version control
main.py UpdateTest.py summary.py RecordInsertion.py Filter.py UpdateRecord.py Filter Validation
1 1355555: Bgt, 2024-01-01 14:10, 44.2, mg/dL, completed, 2024-01-01 15:30
2 1355555: LDL, 2024-01-01 14:10, 13.4, mg/dL, completed, 2024-01-01 15:30
3 1300500: RBC, 2024-03-01 05:20, 13.5, mg/dL, completed, 2024-03-01 05:30
...
4 1210447: BGT, 2020-04-07 13:43, 83.0, mg/dL, reviewed
5 1212121: diastole, 2022-11-04 15:32, 12, mm Hg, completed, 2022-11-05 04:38
6 1210447: systole, 2020-04-07 13:43, 70.0, mm Hg, reviewed
7 1210447: Hgb, 2023-02-07 13:43, 14, mg/dL, completed
8 1210447: systole, 2023-07-18 18:14, 74.0, mm Hg, completed, 2023-07-19 02:18
9 1210447: RBC, 2024-11-12 13:43, 52.5, mm hh, pending
10 1210447: upT, 2012-03-08 12:12, 80.0, mm po, reviewed
11 1212121: BGT, 2022-11-04 15:32, 12, mg/dL, completed, 2022-11-05 04:38
12 1112223: diastole, 2020-02-04 14:02, 47, mg/dL, pending
13 1210447: Hgb, 2020-03-27 21:27, 14, g/dL, completed, 2020-03-28 00:31
14 1210447: systole, 2020-04-03 12:01, 14.2, g/dL, completed, 2020-04-03 15:05
15 1210447: BGT, 2016-11-30 08:02, 41.2, mg/dL, completed, 2016-11-30 11:06
16 1210447: LDL, 2016-11-30 08:02, 41.2, g/dL, completed, 2016-11-30 11:06
17 1112223: upT, 2020-02-04 14:02, 47, mm po, completed, 2020-02-06 16:17
18 1112223: upT, 2024-02-04 14:32, 87, mm po, completed, 2024-02-06 16:47
19 1111111: Hgb, 2024-07-12 18:00, 12.5, g/dL, reviewed
```

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:

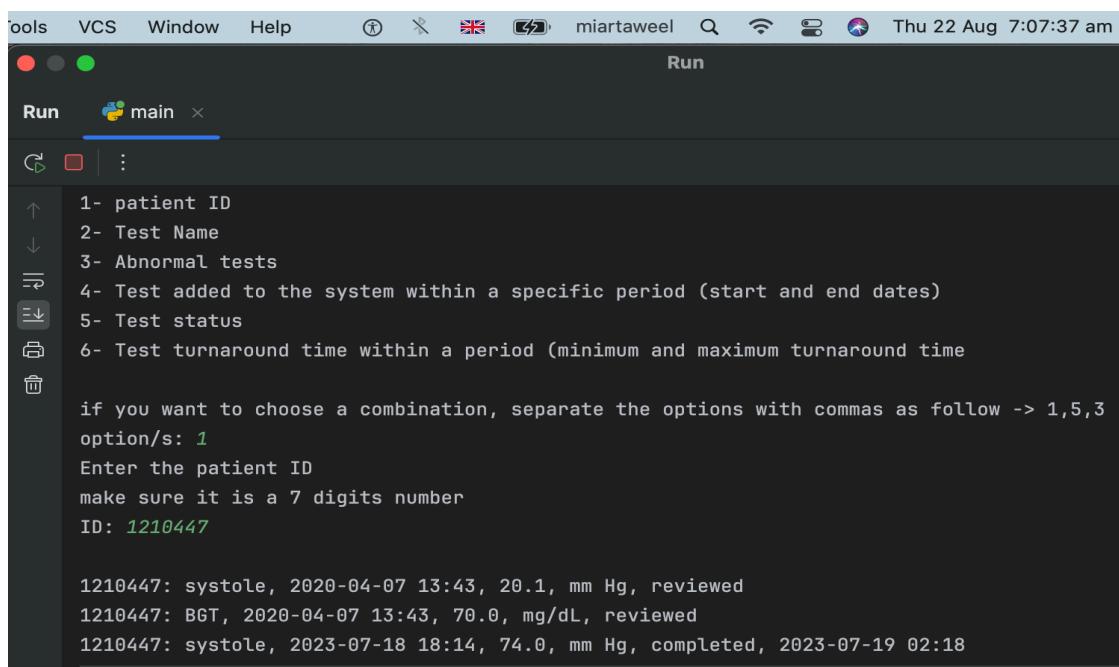
The operation starts by selecting the search by patient ID option from the option menu, then inserting an existing patient ID to search for :



```
choose an option:  
1- Add a new medical test  
2- Add a new medical test record  
3- Update patient records  
4- Update a medical test  
5- Filter medical tests  
6- Summary reports  
7- Export medical records to a CSV file  
8- Import medical records from a CSV file  
9- Exit  
  
option: 5  
filter
```

- **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 :



```
1- patient ID  
2- Test Name  
3- Abnormal tests  
4- Test added to the system within a specific period (start and end dates)  
5- Test status  
6- Test turnaround time within a period (minimum and maximum turnaround time)  
  
if you want to choose a combination, separate the options with commas as follow -> 1,5,3  
option/s: 1  
Enter the patient ID  
make sure it is a 7 digits number  
ID: 1210447  
  
1210447: systole, 2020-04-07 13:43, 20.1, mm Hg, reviewed  
1210447: BGT, 2020-04-07 13:43, 70.0, mg/dL, reviewed  
1210447: systole, 2023-07-18 18:14, 74.0, mm Hg, completed, 2023-07-19 02:18
```

- **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 :

```
actor Run Tools VCS Window Help ⓘ ✖ miartaweeel ⚡ Thu 22 Aug 7:26:45 am
Run main × summary × Run

From the following menu, choose an option or a combination of options to filter the data based on them

1- patient ID
2- Test Name
3- Abnormal tests
4- Test added to the system within a specific period (start and end dates)
5- Test status
6- Test turnaround time within a period (minimum and maximum turnaround time)

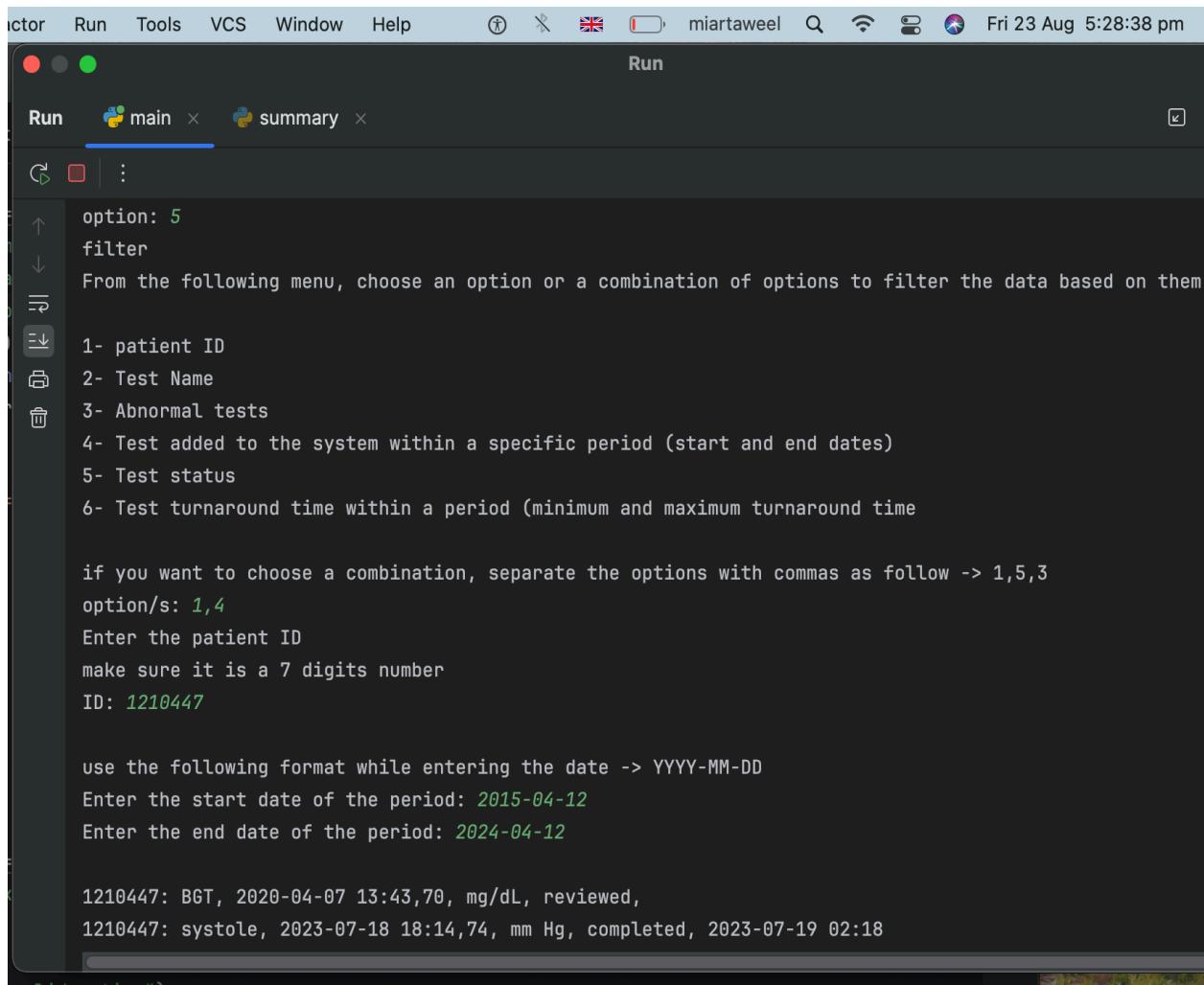
if you want to choose a combination, separate the options with commas as follow -> 1,5,3
option/s: 3,5
What is the status that you're searching for
1- Pending
2- Completed
3- Reviewed
option: 2

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
```

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 :



```
option: 5
filter
From the following menu, choose an option or a combination of options to filter the data based on them
1- patient ID
2- Test Name
3- Abnormal tests
4- Test added to the system within a specific period (start and end dates)
5- Test status
6- Test turnaround time within a period (minimum and maximum turnaround time)

if you want to choose a combination, separate the options with commas as follow -> 1,5,3
option/s: 1,4
Enter the patient ID
make sure it is a 7 digits number
ID: 1210447

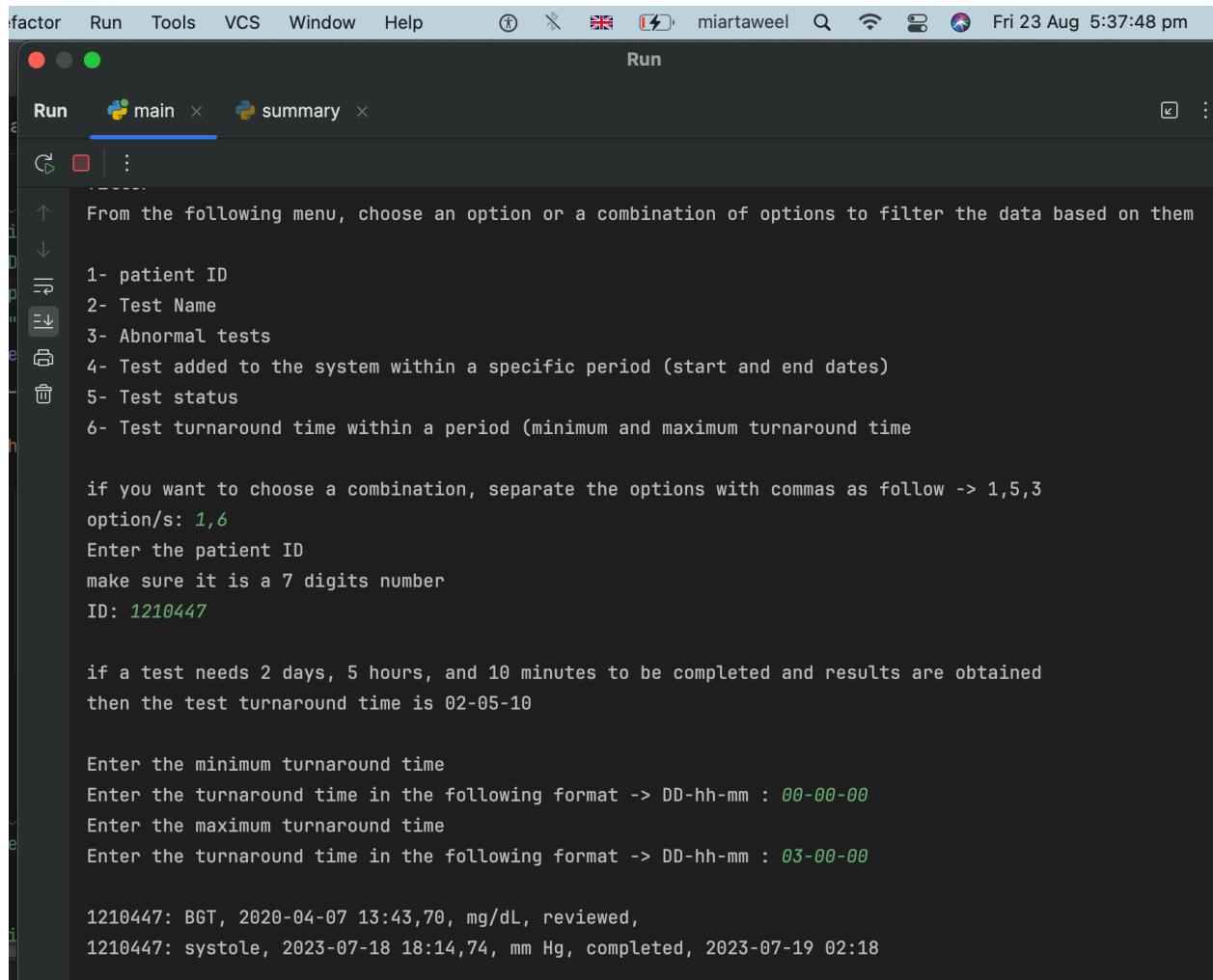
use the following format while entering the date -> YYYY-MM-DD
Enter the start date of the period: 2015-04-12
Enter the end date of the period: 2024-04-12

1210447: BGT, 2020-04-07 13:43,70, mg/dL, reviewed,
1210447: systole, 2023-07-18 18:14,74, mm Hg, completed, 2023-07-19 02:18
```

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 :



```
factor Run Tools VCS Window Help ⓘ ⚡ miartawee! Fri 23 Aug 5:37:48 pm
Run
Run main × summary ×
Run
From the following menu, choose an option or a combination of options to filter the data based on them
1- patient ID
2- Test Name
3- Abnormal tests
4- Test added to the system within a specific period (start and end dates)
5- Test status
6- Test turnaround time within a period (minimum and maximum turnaround time)

if you want to choose a combination, separate the options with commas as follow -> 1,5,3
option/s: 1,6
Enter the patient ID
make sure it is a 7 digits number
ID: 1210447

if a test needs 2 days, 5 hours, and 10 minutes to be completed and results are obtained
then the test turnaround time is 02-05-10

Enter the minimum turnaround time
Enter the turnaround time in the following format -> DD-hh-mm : 00-00-00
Enter the maximum turnaround time
Enter the turnaround time in the following format -> DD-hh-mm : 03-00-00

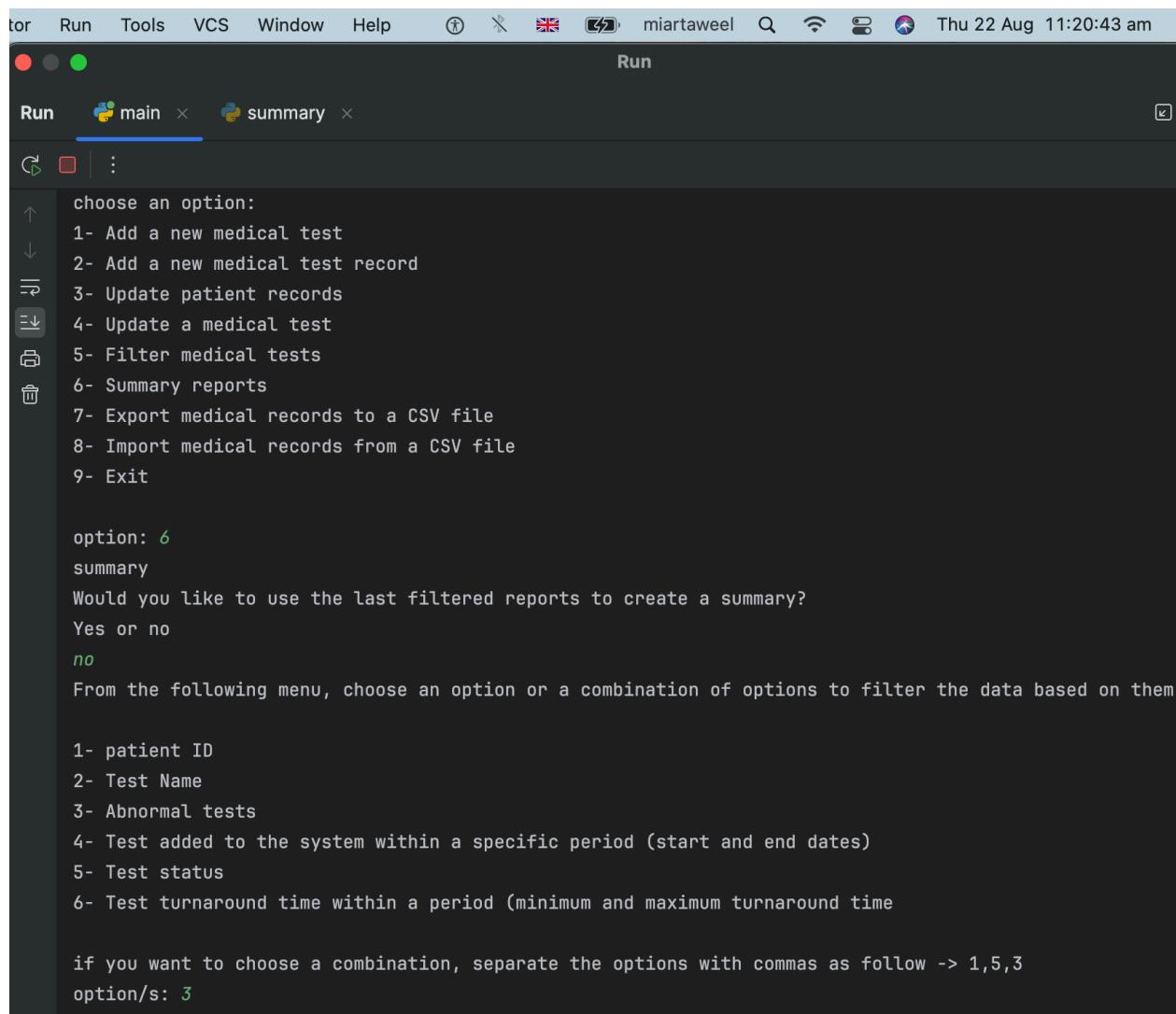
1210447: BGT, 2020-04-07 13:43,70, mg/dL, reviewed,
1210447: systole, 2023-07-18 18:14,74, mm Hg, completed, 2023-07-19 02:18
```

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 executed:



A screenshot of a terminal window titled "Run". The window shows a menu with "summary" selected. The terminal output displays a list of options, the selection of option 6 ("Summary reports"), a confirmation prompt, a filtering menu, and a final note about combining options.

```
choose an option:
1- Add a new medical test
2- Add a new medical test record
3- Update patient records
4- Update a medical test
5- Filter medical tests
6- Summary reports
7- Export medical records to a CSV file
8- Import medical records from a CSV file
9- Exit

option: 6
summary
Would you like to use the last filtered reports to create a summary?
Yes or no
no
From the following menu, choose an option or a combination of options to filter the data based on them

1- patient ID
2- Test Name
3- Abnormal tests
4- Test added to the system within a specific period (start and end dates)
5- Test status
6- Test turnaround time within a period (minimum and maximum turnaround time)

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

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

```

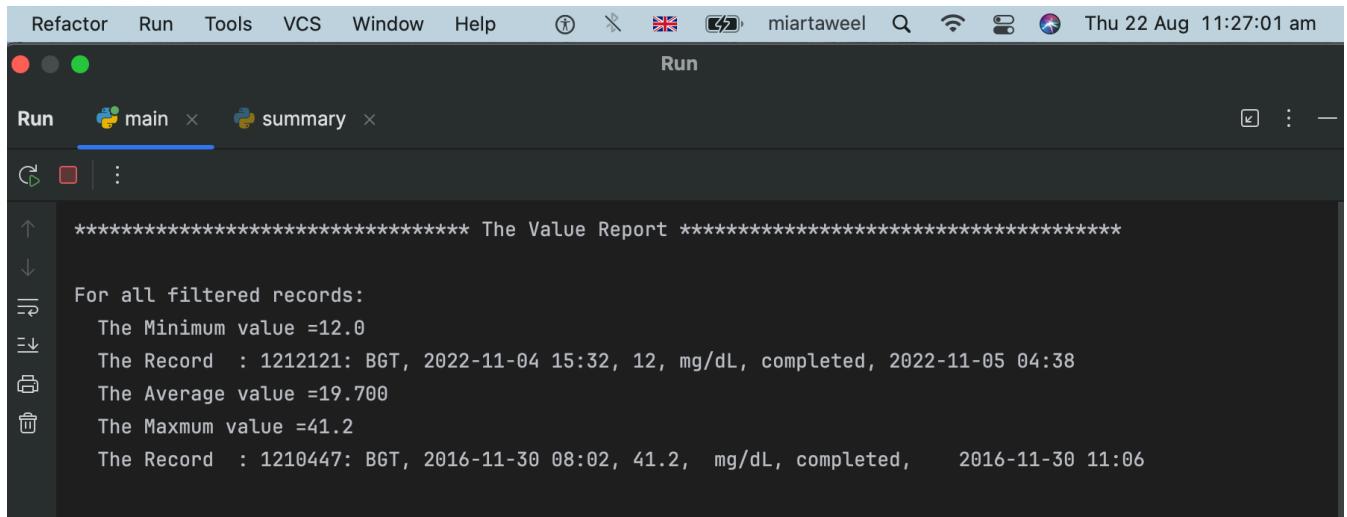
if you want to choose a combination, separate the options with commas as follow -> 1,5,3
option/s: 3

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 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 screenshot shows the PyCharm IDE interface with the 'summary' file open. The code output window displays a summary report for all filtered records:

```

*****
The Value Report *****
For all filtered records:
The Minimum value =12.0
The Record : 1212121: BGT, 2022-11-04 15:32, 12, mg/dL, completed, 2022-11-05 04:38
The Average value =19.700
The Maximum value =41.2
The Record : 1210447: BGT, 2016-11-30 08:02, 41.2, mg/dL, completed, 2016-11-30 11:06

```

- **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 exist

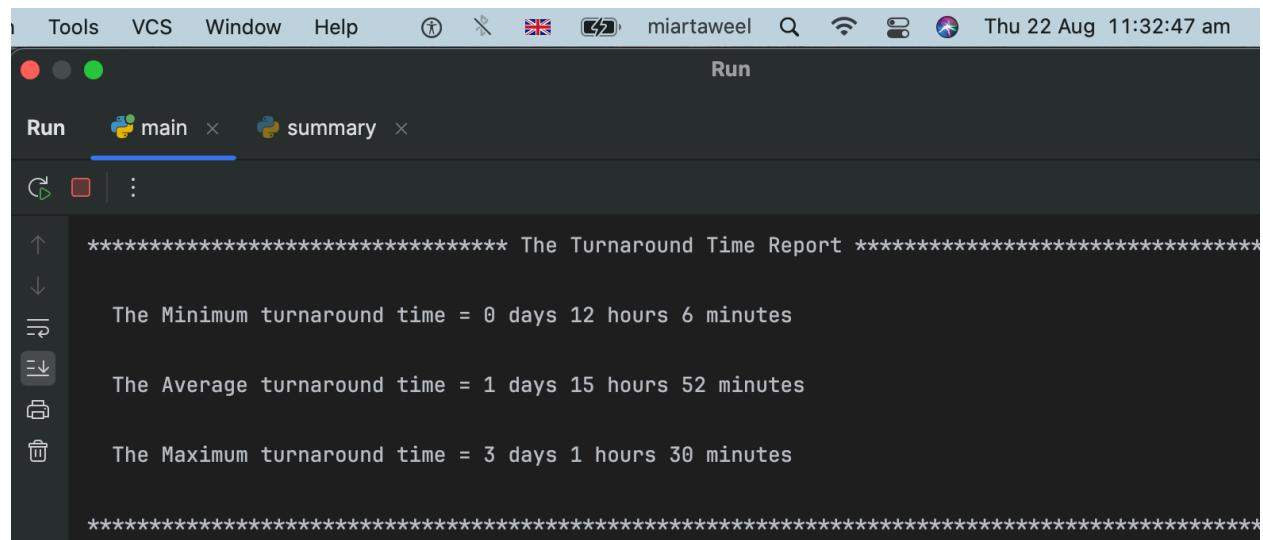
-Red blood cells (RBC):
  The Minimum value =13.5
  The Minimum Record : 1300500: RBC, 2024-03-01 05:20, 13.5, mg/dL, completed, 2024-03-01 05:30
  The Maximum value =13.5
  The Maximum Record =1300500: RBC, 2024-03-01 05:20, 13.5, mg/dL, completed, 2024-03-01 05:30
  The average =13.500

-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**



```

*****
The Turnaround Time Report *****

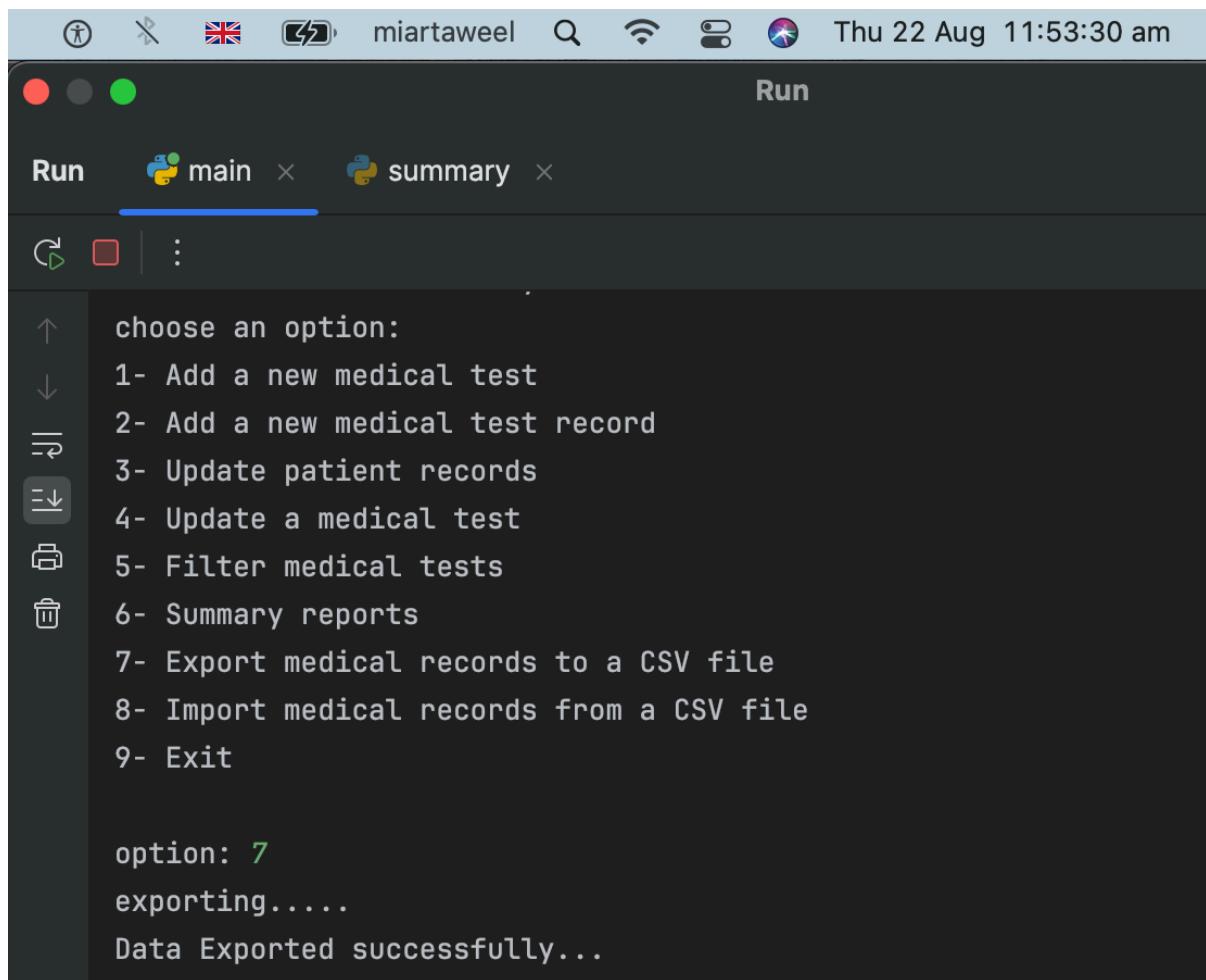
The Minimum turnaround time = 0 days 12 hours 6 minutes
The Average turnaround time = 1 days 15 hours 52 minutes
The Maximum turnaround time = 3 days 1 hours 30 minutes
*****
```

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:



A screenshot of a terminal window titled "Run". The window shows a Python menu with the following options:

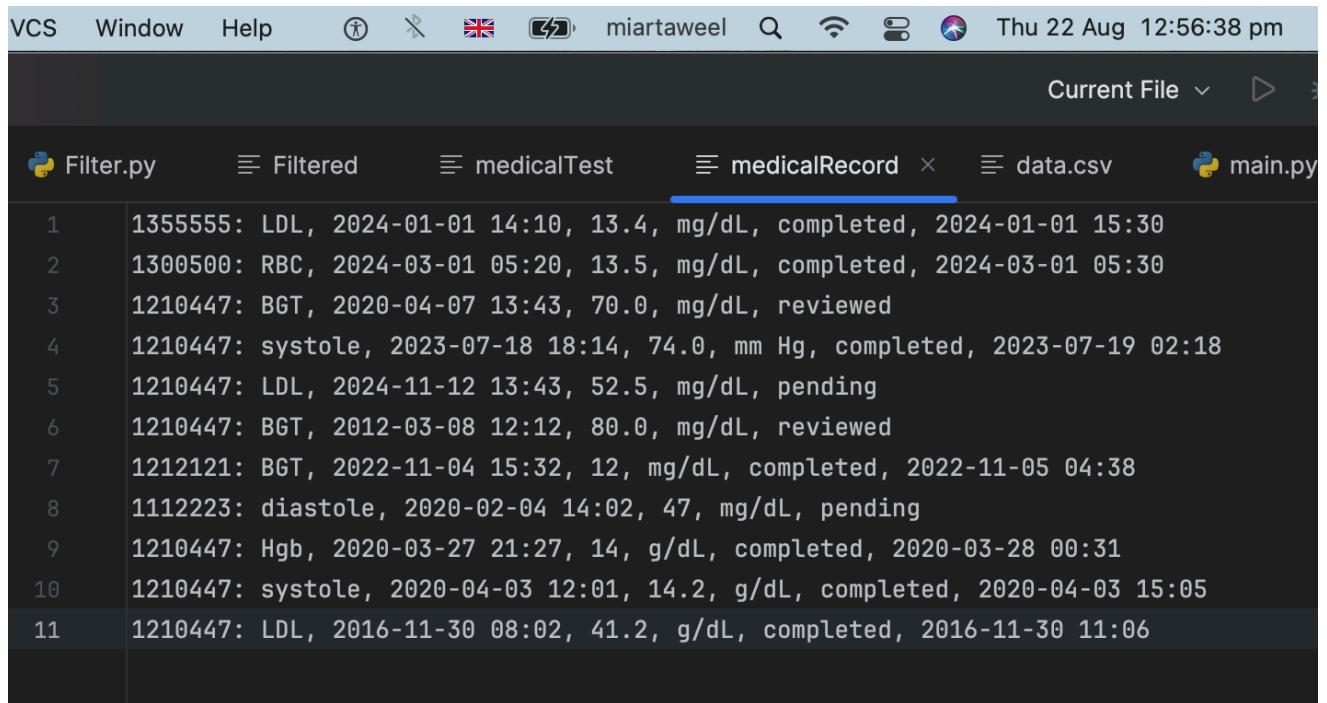
```
choose an option:  
1- Add a new medical test  
2- Add a new medical test record  
3- Update patient records  
4- Update a medical test  
5- Filter medical tests  
6- Summary reports  
7- Export medical records to a CSV file  
8- Import medical records from a CSV file  
9- Exit
```

After selecting option 7, the terminal displays:

```
option: 7  
exporting.....  
Data Exported successfully...
```

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::**



The screenshot shows a terminal window with the following details:

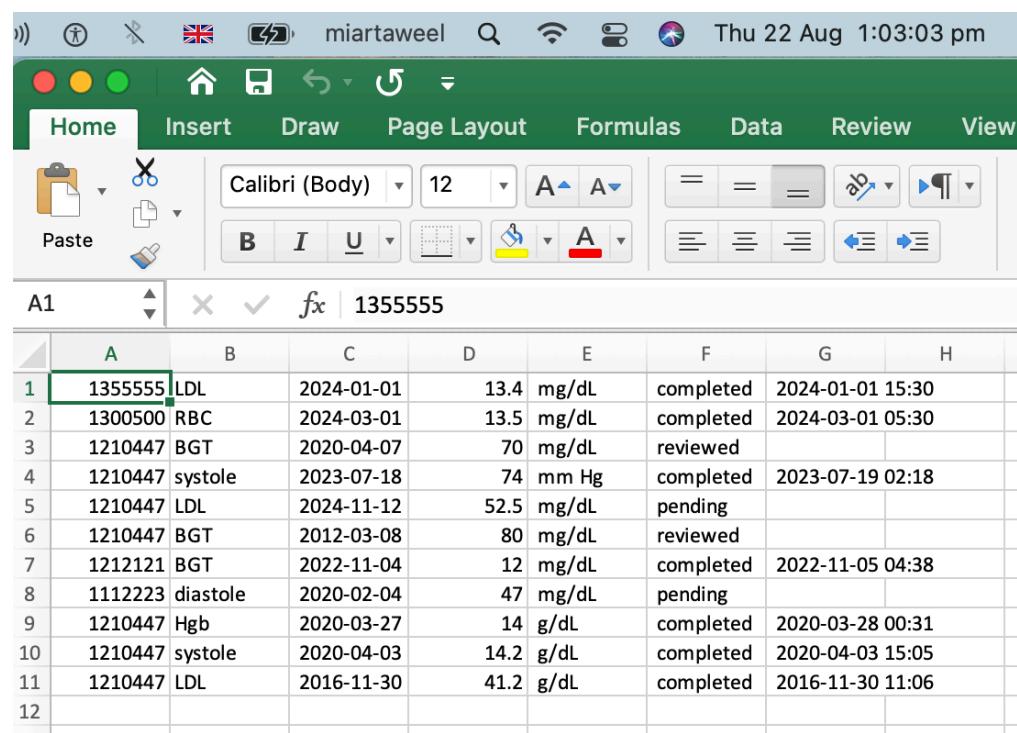
- Top bar: VCS, Window, Help, icons for user, UK flag, battery, signal, and network.
- Current File dropdown: Current File ▾
- File tabs: Filter.py, Filtered, medicalTest, **medicalRecord**, data.csv, main.py
- Content area (stdout):
 

```

1 1355555: LDL, 2024-01-01 14:10, 13.4, mg/dL, completed, 2024-01-01 15:30
2 1300500: RBC, 2024-03-01 05:20, 13.5, mg/dL, completed, 2024-03-01 05:30
3 1210447: BGT, 2020-04-07 13:43, 70.0, mg/dL, reviewed
4 1210447: systole, 2023-07-18 18:14, 74.0, mm Hg, completed, 2023-07-19 02:18
5 1210447: LDL, 2024-11-12 13:43, 52.5, mg/dL, pending
6 1210447: BGT, 2012-03-08 12:12, 80.0, mg/dL, reviewed
7 1212121: BGT, 2022-11-04 15:32, 12, mg/dL, completed, 2022-11-05 04:38
8 1112223: diastole, 2020-02-04 14:02, 47, mg/dL, pending
9 1210447: Hgb, 2020-03-27 21:27, 14, g/dL, completed, 2020-03-28 00:31
10 1210447: systole, 2020-04-03 12:01, 14.2, g/dL, completed, 2020-04-03 15:05
11 1210447: LDL, 2016-11-30 08:02, 41.2, g/dL, completed, 2016-11-30 11:06
      
```

- **Output viewed in CSV file :**

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



The screenshot shows an Excel spreadsheet with the following details:

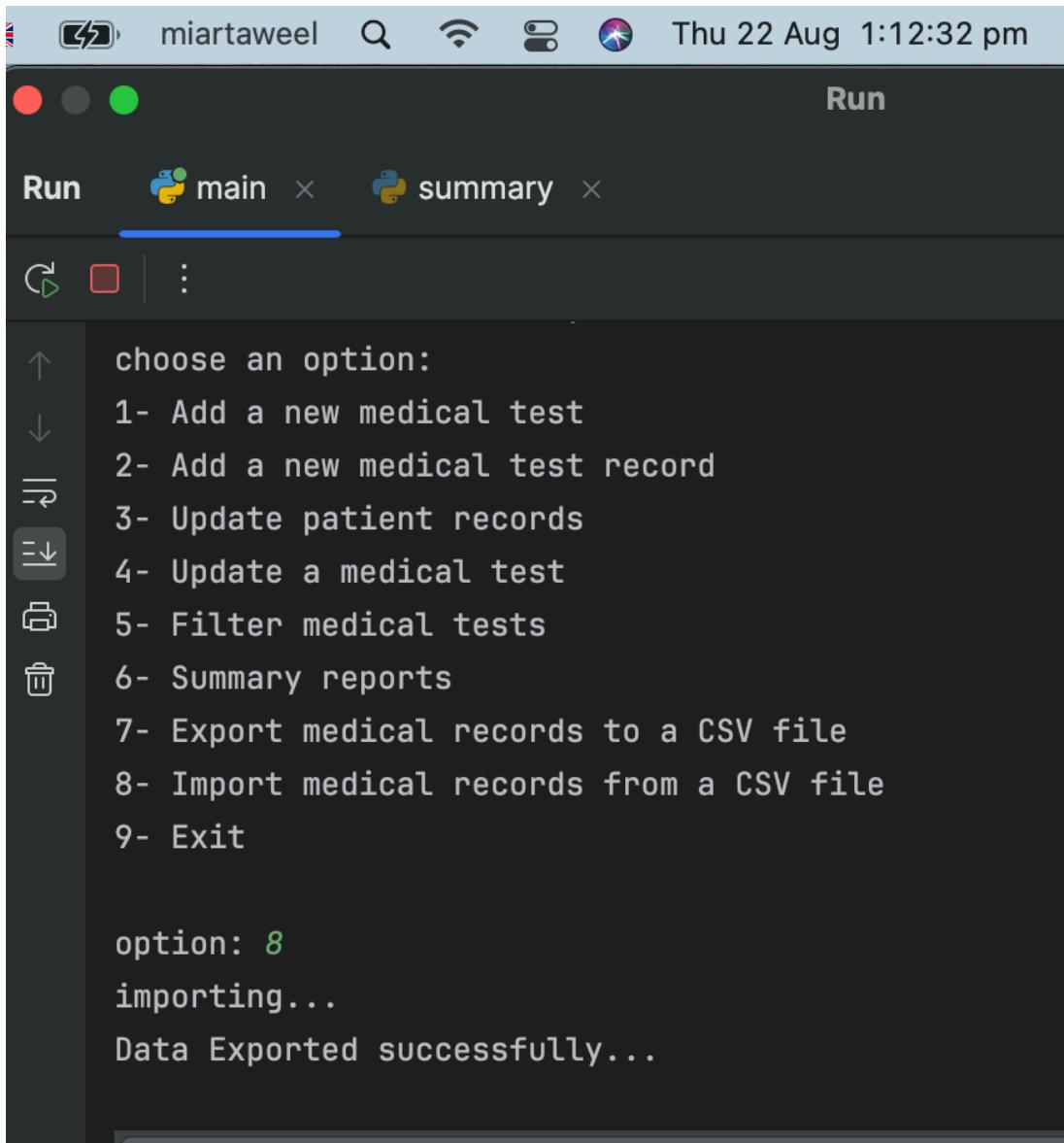
- Top menu: Home, Insert, Draw, Page Layout, Formulas, Data, Review, View.
- Toolbar icons: Paste, Calibri (Body), 12pt, Bold, Italic, Underline, Alignment, Font Color, Alignment, Font Size.
- Cell A1: 1355555
- Table structure:
 

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

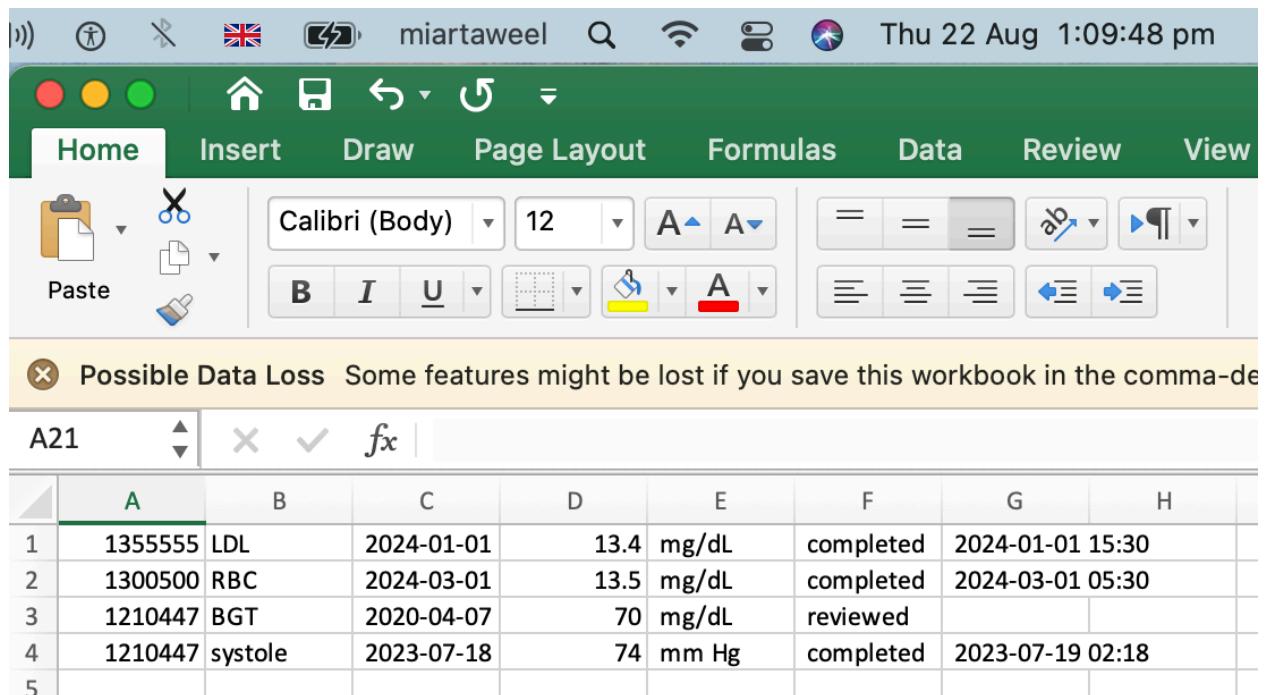
The operation starts by selecting the “Import medical records from a CSV file “ option from the drop down menu:



```
choose an option:  
1- Add a new medical test  
2- Add a new medical test record  
3- Update patient records  
4- Update a medical test  
5- Filter medical tests  
6- Summary reports  
7- Export medical records to a CSV file  
8- Import medical records from a CSV file  
9- Exit  
  
option: 8  
importing...  
Data Exported successfully...
```

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

- CSV file Data::

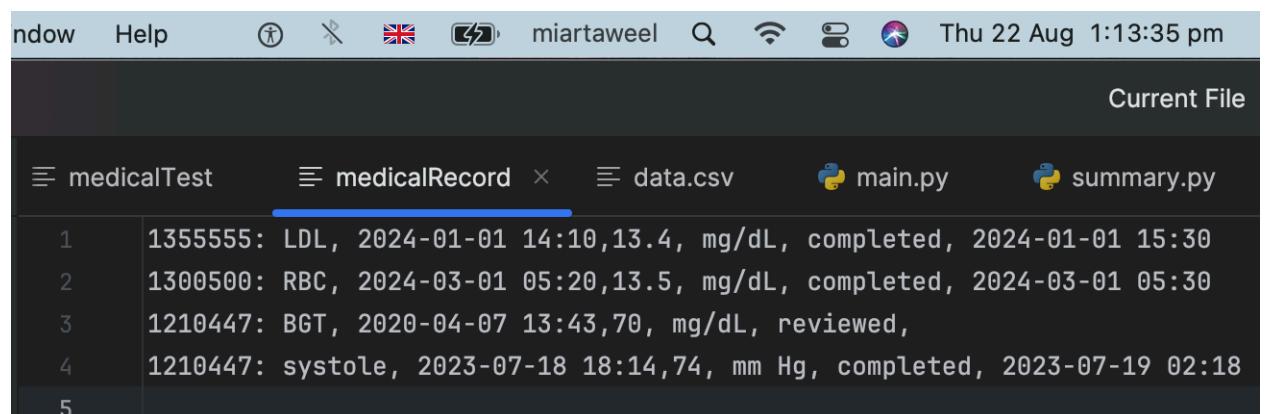


A screenshot of Microsoft Excel showing a CSV file named "miartaweeel". The Excel ribbon is visible at the top, with the "Home" tab selected. The data is displayed in a table with columns A through H. The rows contain the following data:

	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 :



A screenshot of a terminal window titled "Current File". The window shows several tabs: "medicalTest", "medicalRecord", "data.csv", "main.py", and "summary.py". The "medicalRecord" tab is active, displaying the following text:

```
1 1355555: LDL, 2024-01-01 14:10,13.4, mg/dL, completed, 2024-01-01 15:30
2 1300500: RBC, 2024-03-01 05:20,13.5, mg/dL, completed, 2024-03-01 05:30
3 1210447: BGT, 2020-04-07 13:43,70, mg/dL, reviewed,
4 1210447: systole, 2023-07-18 18:14,74, mm Hg, completed, 2023-07-19 02:18
5
```