**CSU33D03 Group 8**

**Project 2 Report**



|  |  |
| --- | --- |
| Name | Student ID |
| Liam Fenelon | 17324889 |
| Seán Gallagher | 17324883 |
| Shubham Goel |  |
| Aditya Shrivastava | 19323354 |

Date of Submission: 10/04/2020

# **Problem we are tackling:**

As the number of diagnosed COVID-19 cases upsurge daily there is an increasing pressure on the Health system.

There is a high possibility that the trajectory of the increase in cases could be worse than predicted and this could result in the crippling of the Health System due to the growing stress and pressure.

Overtime it is going to get challenging to keep track of, pay attention to and offer personalized advice to every citizen in the country as there will be an unhealthy divide in attention between diagnosed cases and the ones that are symptomatic/asymptomatic but still need help.

Our proposed solution is a network that aims to: reduce the stress on the HSE and help them optimize their system, keep track of the infected and uninfected population, and provide adequate attention and advice to each citizen.

# **Solution we propose:**

Our solution is to develop a Web application that will collect, store and update almost every citizens information and status regarding the Coronavirus.

The app will allow each citizen to input their:

1. Personal Details: Name, PPS Number, Age, Gender, Location (County, Address) etc.
2. Their status regarding COVID- 19: Positive, Negative or Have Symptoms
3. Suggest Action and feedback by the Health Services on the next step that the user should take

The information collected will help the HSE organize and keep track of the population. Having vital information and being in touch with almost every citizen in an organized fashion will help the health services to strategize well and reduce pressure on them.

From a citizen’s perspective, one would feel more secure as he/she is in touch with the health services in a more personalized way and can get feedback regarding their situation as and when needed.

# **Summary of collective Project 1 Knowledge and Experience**

## Strengths

The group had strong individual members and after working on Project 1 each member had a strong understanding of:

* How to design a network and implement it on Python
* A strong understanding of the TCP/IP model and the protocols such as TCP and UDP
* How to implement client-server and p2p networks on Python and C
* Developing a GUI on Python.

## Beneficial features

From each members Project 1’s we as a group were able to pick out the beneficial features for our Project 2, they were:

* Using python’s socket library for implementing our network
* A GUI, and
* Seán’s project 1 code which was used to implement our solution for Project 2

## Lesson learned and expert knowledge discussed and agreed

Each of us learned quite a few things from project 1 such as how to properly design and implement a network on a system.

After several group meetings we agreed on what we should proceed with and what should we pick up from our previous projects.

* We decided that we would proceed with a Client-Server implementation for our proposed network
* Python was selected as the language for implementing the Project since it was convenient to use and it was simpler to design a GUI over it

# **Needs and requirements that we plan to target and how.**

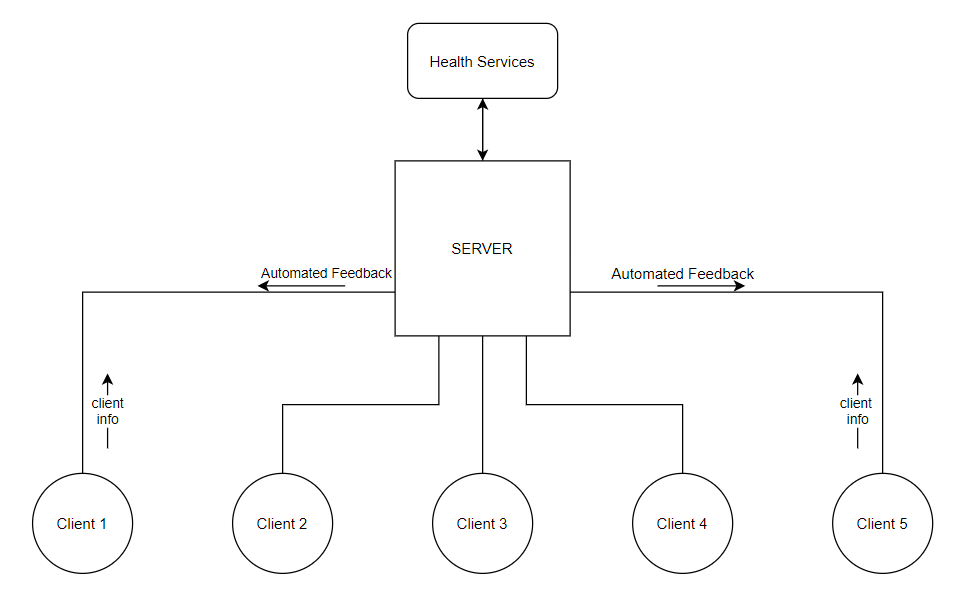
As our solution is a web application that will collect, store and update almost every citizens information and status regarding the Coronavirus, we plan to target the following:

1. The collection of user information, general as well as specific information w.r.t the virus
   * We do so by implementing a simple *GUI* which simply asks for the user’s details such as their PPS Number, Name, Age, Location, Gender Phone Number etc. *and specifics such as*: *If you have, had or have not contracted the virus or not and If the user has symptoms*
2. Forming statistics with the user data collected
   * With the collected user information which is stored in the server, we created variables in Python for the data collected and incremented/ decremented it as and when user data was collected and updated.
   * For example, if a user’s gender is male, the counter for male is incremented by 1
3. Providing the user with **automatic feedback** based on their status regarding the virus, symptoms and their age
   * Based on the user information collected in the server, we compare their information in a few *if else blocks* in our code and then provide them general feedback which would be provided by the Health Services
   * For example, if a user is (a.) over the age of 60 and (b.) has symptoms but (c.) has never contracted the virus then the feedback would be: Please self-isolate, you will be tested at your doorstep soon.
4. Delivering the data collected on the server to the Health Services.
   * The Health Services will have access to all the data collected from the application and it would be provided to them.

# **Networking/communication aspects of your planned approach and solution**

The networking aspect of our planned approach were:

1. A Client-Server Network: The users act as clients and their information is stored on a server
2. TCP was used as the Transport Layer Protocol as it is more reliable and has features such as error checking which is essential for storing the correct client information. IPv4 was used as the internet layer protocol
3. To communicate with the user, a GUI was created based on the tkinter library on Python



# **Other requisite knowledge and skills required**

The other requisite knowledge we required were:

1. A good knowledge of programming languages such as C and Python which helped us in implementing the GUI
2. A good understanding of the current situation of the virus. This helped us to model our web app to make it more friendly and convenient
3. A good understanding of computer networks and the protocols required to implement our solution.

# **Pseudo Code**

## Server

This program acts as the server for the network, accepting new connections from clients wishing to join. Clients can then send their information and in return the server can send back an automated feedback response, based off that information previously sent.

Set-up { Declare variables used in various functions Ask for the Servers name, create a header from it and encode for sending Create a socket at the local host IP and on a non-privileged port number Create a list for all the potential client sockets, a dictionary for all clients user name details and create a dictionary to store all users’ details }

def received messages() { Try’s to receive the header, the length of the header and then receive the of the incoming message from a client of that length. Then returns the header and the message as an object

}

def handle\_message(sock) { Decodes the various messages received prior to the calling of this function if the user hasn’t left the network. If they have, the client is removed from the socket list and their details deleted. Sends the user an immediate response informing the message was received. Stores all of the information received from the client in a dictionary. A risk rating is then calculated based off the information contained in the messages previously decoded. }

def automated\_response(sock) { Based off the risk rating calculated in the previous function, one of the possible responses is sent to the client. In each case, a header for the response message is created and both the response and the response header are encoded. These two along with the server’s name header and the server’s name are sent first to allow the client to know who the message came from. }

def stat\_calc() {

Bases off the information taken in prior to the calling of this function, various statistics are calculated and printed out to the server’s console. }

Main { Run through and check all the sockets in the socket list to see if any are active. If the notified socket is the server socket, it means someone is trying to join the network { accept the new connection, an decode the user name append the client to the list of clients and reply with conformation of joining }

Else: it’s a client already known on the network Receive all the different messages via the receive\_messgae function and deal with them appropriately using the handle\_message function. Then print out the dictionary containing all the client’s details and call the stat\_calc function }

## Client

The following program allows for a client to join the network, interacting with the server through a GUI. Initially they send their PPS number and then the connection is established. They now send their information and get an automated feedback response.

Set up { Set up variables and create object for GUI: “top = tk()” }

class Checkbar(Frame) { used to create the checkboxes and obtain which text box is checked off. }

def receive\_message(sock) {

try: Gets incoming server’s name header and server’s name. Then decode them. Then does the same for the message. Once receives, place in textbox

except IOError as e: If error, closes the console }

*\*\*This a button on the GUI that send the user\_id to the server when pressed*

def enter\_user() {

gets the user\_id from the text box and if it is not blank then {Sends the user\_id to the server and then wait and receive the returned message }

*else* Prints an error message – “Enter PPS Number First” in the feedback text box }

*\*\*This a button on the GUI that send all the information to the server when pressed*

def commit\_info() {

if all the fields are not blank { Gets all the information from the fields entered on the GUI by the user and set the various variables accordingly. Sends the information and clear the text on the GUI Then waits and receives response messages from the server else prints out error message directing to complete all fields }

def quit\_gui() { msg = ‘quit’ send to server, notifying closing of connection console close }

GUI Construction { This section of code is what gives the GUI, “top” all of its features. This is done through the use of the tkinter library. \*\*Note: I could go through all this but this would take up the entire 6 pages of this report allocated. It is pretty self-explanatory}

The client’s socket is then set up like in the server’s program

Main() {

The client is constantly prompt for an input if there is an input message { call receive\_message(client\_socket) }

# **Test cases and Validation**

We created the Server and Client program on Python. First, we run the Server program and name the server so that it is up and running. We can run multiple client programs which would pass the client information through the GUI to the server and receive feedback in return.

For our test cases:

1. We set up the server following which we run our client program and we see the GUI

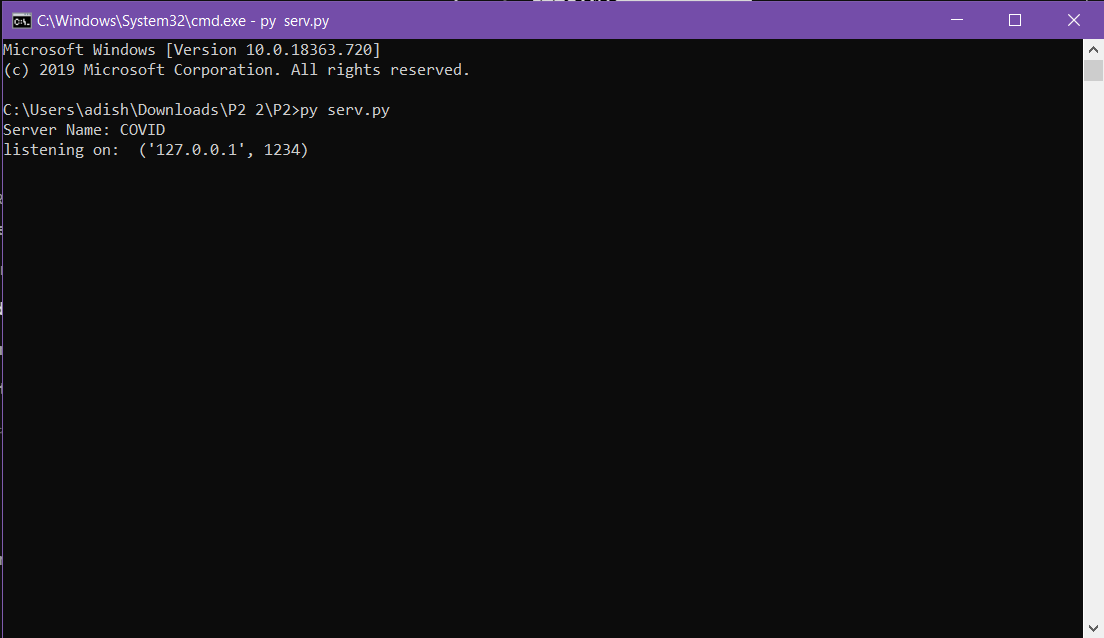
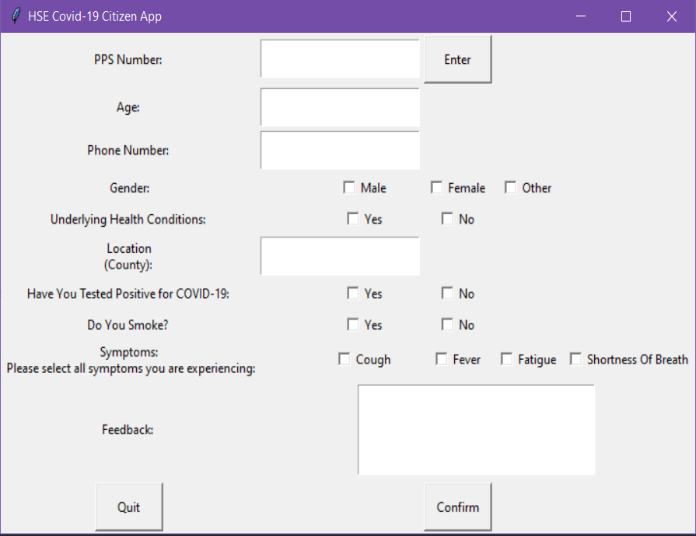
 

Figure 1

1. To check whether our application was working accurately and to validate it we ran a few test cases

* First, we entered one client’s details with the gender as Male, Age:71 and several symptoms to indicate that the client was at risk. We see these stats on the server in figure 2 and we see that the precise feedback has been received by the Client

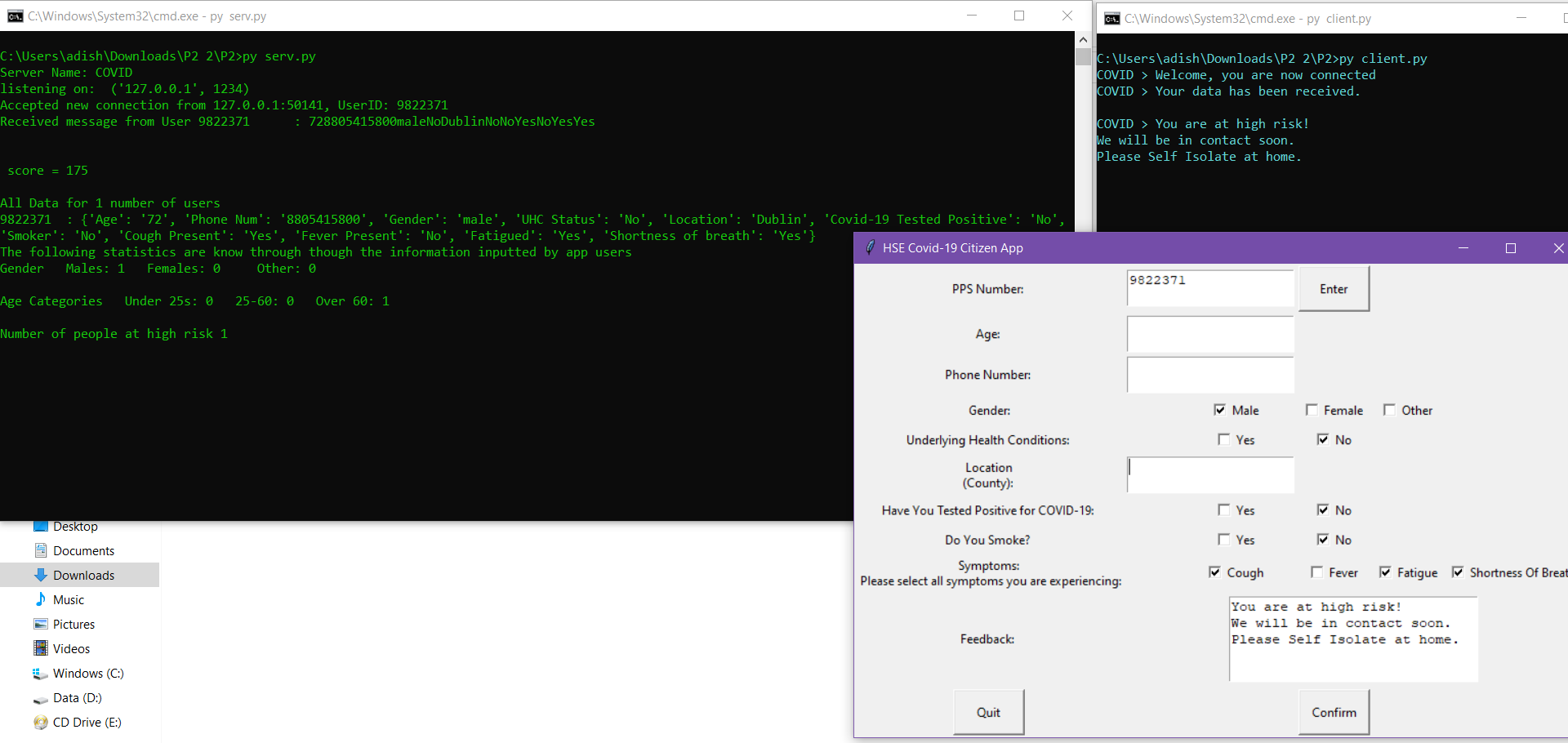


Figure 2

* Following this, we run another client program to check whether the server can handle several client at once. The second client runs successfully and then we enter similar information as the first client to receive automated feedback. As we see we receive the precise feedback that we should in figure 3
* In figure 3 we can also see the statistics on the server such as Age categories and Number of people at high risk

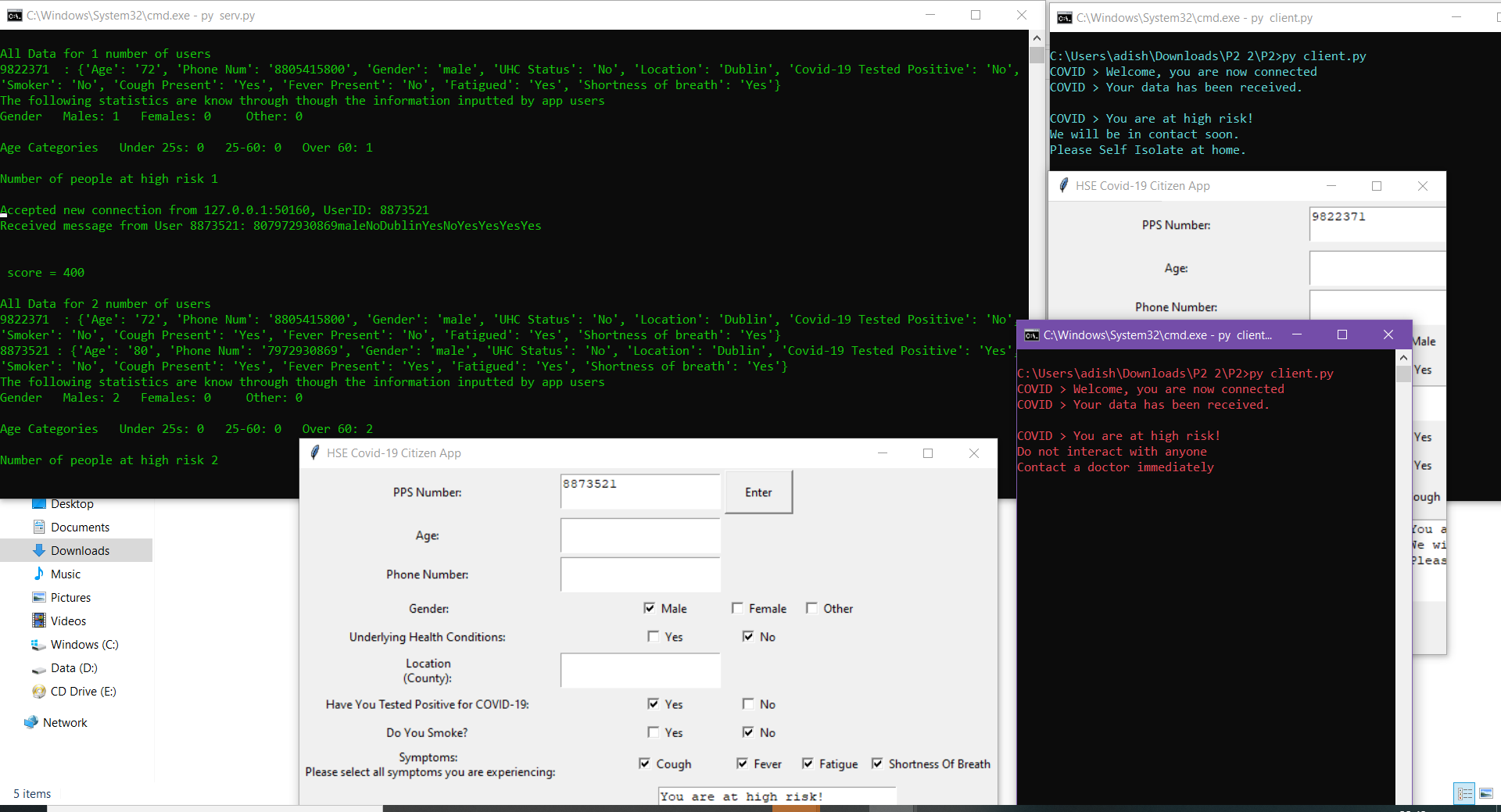


Figure 3

# **Responsibilities taken**

|  |  |
| --- | --- |
| Name: | Responsibility: |
| Liam Fenelon | Developing and implementing the GUI and the program for the client. Lead demonstration and created README file. |
| Seán Gallagher | Developed and implemented the entire code for the client and the server for our solution. Also, produced the pseudo code for the report. |
| Shubham Goel | Nothing – Didn’t even reply. |
| Aditya Shrivastava | Implementation a dictionary in Python to store users’ information at the server. Created the report and slides for the presentation for Project 2. |

# **The value and potential importance of your solution and implementation**

* Our solution is important as it would help the health services get in touch with almost every citizen with an internet connection in the country. This way the Health Services can have an update on every citizens status regarding the virus. Our application also allows the country to have each citizens information organized.
* Our application has the potential to know each citizens location which would help the HSE spot clusters and this will also make contact tracing simpler for them. This reduces the risk of spreading the virus and it can be contained with ease.
* As the HSE will have the required information it can also help them realize which members of the country need Cocooning.
* Our implementation provides the client with a convenient interface where they can store and update their information and status regarding the virus.
* Out application is important from a client’s perspective as it helps them stay in touch with the HSE in a more personalized manner and they can contact them through the application as and when needed.

From the points stated above we see that our application is highly valuable from both the health services and the citizens perspective. It also has a very high potential and this can be reached with a few more touches and refining performed on the application.

# **Individual Feedback**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Liam Fenelon | Seán Gallagher | Shubham Goel | Aditya Shrivastava |
| Name initials |  | LS | SGa | SG | AS |
| Effort |  | High | High | Zero | High |
| Effectiveness |  | High | High | Zero | High |
|  |  |  |  |  |  |

***Aditya Shrivastava***: Project 2 was a good learning experience for me. I realized how my understanding in Computer Networks and socket programming could actually help me develop an application that would provide an essential service during the pandemic. I really enjoyed working on this project with my teammates who were so helpful, coordinated flawlessly and put in their maximum effort.

***Seán Gallagher:*** Project two gave me a great chance to expand on my project 1 implementation and build upon what I had learned already. Collaborating with my group members gave me an insight into others ideas and methods than I had not been exposed to. We were able to implement a high-level interface, much better than the on I had implemented in project 1, and implement more means to ensure reliable data transfer, test cases I had not coded for in project 1. Overall, project 2 definitely benefitted my learning and by the end of it I had earned the basics if not more, of a new programming language; Python.

***Liam Fenelon:*** Project two was interesting challenge that I enjoyed thoroughly. It was interesting to apply socket programming to a real-world scenario. It was also interesting to work with others. The collaboration between the group members expanded my scope and helped develop ideas. We began by examining our individual projects 1 to see what we were able to build on. We discovered that I was the only one who did not implement project 1 in python. I began familiarizing myself with python while Sean began building the client-server model. Once I was confident with python, I coordinated with Sean to see what I needed to work on with the code. We turned our attention to the GUI and ensuring the user would give us reliable and appropriate data that was easily interpreted by the server. Overall, project 2 was very beneficial to my knowledge of computer networks. I gained new skills in python which I had never used before and broadened my scope of possibilities by exposure to other group members and their ideas.

***Shubham Goel:***