Contents

1.0 Introduction

This is a documentation regarding the Vaccine Inventory Management System that needs developing for an authorized pharmaceutical company. It has come to my attention that the pharmaceutical company has a list of requirements that must be instilled within the system. The system's performance is expected to at least satisfy the basic requirements stated by the pharmaceutical company.

2.0 Assumptions

1. There are only two text files, named "vaccine.txt" and "dist.txt".
2. File vaccine.txt will contain the current information of vaccines.
3. File dist.txt will record distributed vaccines.
4. User can add the quantity of a particular vaccine.
5. User can decrease the quantity of a particular vaccine.
6. System must be menu-based.
7. File will update when vaccine decreases.
8. File will update when vaccine increases.
9. User can search for a particular vaccine.
10. System can produce a list of all vaccines and their distributed quantities.
11. Produced list of all vaccines and their distributed quantities must be in descending order based on quantity distributed.
12. System can display the details of current vaccines.
13. System can handle wrong or incorrect inputs.
14. System allows user to double check their inputs.
15. System validates certain non-duplicable variables.

3.0 Pseudo Code Design

3.1 Display Vaccine

```
Begin

    Declare line

    open file "vaccine.txt" in read mode

    loop line from file

        print(line)

    End loop

    close file

End
```

3.2 Receive and Update Vaccine

```
Start
    Declare line,string_subject,line_count,vaccine_content,
    Declare VAC_name,VAC_code,VAC_country,VAC_dosage,
    Declare VAC_pop_covered,VAC_quantity,index,x,code,rec_amount
    Declare match

    print("Write down the vaccine code and the quantity of
        vaccine received.")
    read code
    read rec_amount
    index = 0
    line_count = 0
    x = 0
    match = 0
    open file "vaccine.txt" in read mode
    loop line from file
        if (line_count == 0)
            string_subject = line
        else
            vaccine_content = split (",") from line.
            VAC_name[index] = vaccine_content[0]
            VAC_code[index] = vaccine_content[1]
            VAC_country[index] = vaccine_content[2]
            VAC_dosage[index] = vaccine_content[3]
            VAC_pop_covered[index] = vaccine_content[4]
            VAC_quantity[index] = vaccine_content[5]
            index += 1
            line_count += 1
        End if
    End loop
    close file

    while (x < index)
        if (VAC_code[x] == code)
            match = 1
            break
        End if
    End while

    if (match == 0)
        print("Wrong code, please try again.")
    Else
        open file "vaccine.txt" in write mode
        write string_subject into file;
        index = 0
        while(x < index)
            write VAC_name[x] into file
            write VAC_code[x] into file
            write VAC_country[x] into file
            write VAC_dosage[x] into file
            write VAC_pop_covered[x] into file
            write VAC_quantity[x] into file
            x += 1
        End while
        close file
    End if
End
```

3.3 Distribute and Update Vaccine

```
Start
        Declare code,dist_amount,permission_id,dist_string_subject,line,dist_content,
        Declare DIST_initial_quantity,DIST_dist_amount,DIST_final_quantity,index
        Declare line_count,vaccine_content,VAC_name,VAC_code,VAC_country
        Declare num,highest,k,j,vac_string_subject,DIST_permissionDIST_code
        Declare line_count,match,x,y,VAC_dosage,VAC_pop_covered,VAC_quantity
        index = 0
        line_count = 0
        x = 0
        match = 0
        vac_index = 0
        y = 0
        num = 0
        k = 0
        j = 0

        print("Write down the vaccine code: ")
        read code
        print("The distributed amount is: ")
        read dist_amount
        print("The permission id is: ")
        read permission_id

        open file "dist.txt" in read mode
        loop line from file
                if (line_count == 0)
                        dist_string_subject = line
                else
                        dist_content = split (",") from line.
                        DIST_permission[index] = vaccine_content[0]
                        DIST_code[index] = vaccine_content[1]
                        DIST_initial_quantity[index] = vaccine_content[2]
                        DIST_dist_amount[index] = vaccine_content[3]
                        DIST_final_quantity[index] = vaccine_content[4]
                        index += 1
                        line_count += 1
                End if
        End loop
        close file

        while (x < index)
                if (DIST[x] == permission_id)
                        print("Permission id:",permission_id,"has
                            already been registered.)
                        match = 1
                        break
                End if
                x += 1
        End while

        if (match == 0)
                line_count = 0
                open file "vaccine.txt" in read mode
                loop line from file
                        if (line_count == 0)
                                vac_string_subject = line
                        else
```

```
            vaccine_content = split (",") from line.
            VAC_name[vac_index] = vaccine_content[0]
            VAC_code[vac_index] = vaccine_content[1]
            VAC_country[vac_index] = vaccine_content[2]
            VAC_dosage[vac_index] = vaccine_content[3]
            VAC_pop_covered[vac_index] = vaccine_content[4]
            VAC_quantity[vac_index] = vaccine_content[5]
            vac_index += 1
            line_count += 1
        End if
    End loop
    close file

    while (y < vac_index)
        if (VAC_code[y] == code)
            initial_quantity = VAC_quantity[y]
            final_quantity = initial_quantity - dist_amount
            VAC_quantity[y] = final_quantity
            break
        End if
        y += 1
    End while

    open file "vaccine.txt" in write mode
    write vac_string_subject into file
    x = 0
    while (k < vac_index)
        while (j < vac_index)
            if (VAC_quantity[j] > num)
                num = VAC_quantity[j]
                highest = j
            End if
            j += 1
        End while

        write VAC_name[highest] into file
        write VAC_code[highest] into file
        write VAC_country[highest] into file
        write VAC_dosage[highest] into file
        write VAC_covered[highest] into file
        write VAC_quantity[highest] into file
        VAC_quantity[highest] = 0
        num = 0
        highest = 0
        k += 1
    End while
    close file

    DIST_permission[index] = permission_id
    DIST_code[index] = code
    DIST_initial_quantity[index] = initial_quantity
    DIST_dist_amount[index] = dist_amount
    DIST_final_quantity[index] = final_quantity

    open "dist.txt" in write mode
    write dist_string_subject into file
    k = 0
    j = 0
```

```
                while (k < index+1)
                        while (j < index+1)
                                if (DIST_dist_amount[j] > num)
                                        num = DIST_dist_amount[j]
                                        highest = j
                                End if
                                j += 1
                        End while

                        write DIST_permission[highest] into file
                        write DIST_code[highest] into file
                        write DIST_initial_quantity[highest] into file
                        write DIST_dist_amount[highest] into file
                        write DIST_final_quantity[highest] into file
                        DIST_dist_amount[highest] = 0
                        num = 0
                        highest = 0
                        k += 1
                End while
                close file
        End if
End
```

3.4 Search by Code

```
Start

    Declare line,vaccine_content,VAC_name,VAC_code

    Declare VAC_country,VAC_dosage,VAC_pop_covered

    Declare VAC_quantity,index,line_count,x,string_subject


    index = 0

    line_count = 0

    x = 0


    print("Please write down the vaccine code: ")

    read code

    open file "vaccine.txt" in read mode

    loop line from file

        if (line_count == 0)

            string_subject = line

        else

            vaccine_content = split (",") from line.

            VAC_name[index] = vaccine_content[0]

            VAC_code[index] = vaccine_content[1]

            VAC_country[index] = vaccine_content[2]

            VAC_dosage[index] = vaccine_content[3]

            VAC_pop_covered[index] = vaccine_content[4]

            VAC_quantity[index] = vaccine_content[5]

            index += 1

            line_count += 1

        End if

    End loop

    close file
    while (x < index)
        if (VAC_code[x] == code)
            print(line)
            print (VAC_name[x])
            print (VAC_code[x])
            print (VAC_country[x])
            print (VAC_dosage[x])
            print (VAC_pop_covered[x])
            print (VAC_quantity[x])
        End if
        x += 1
    End while
End
```

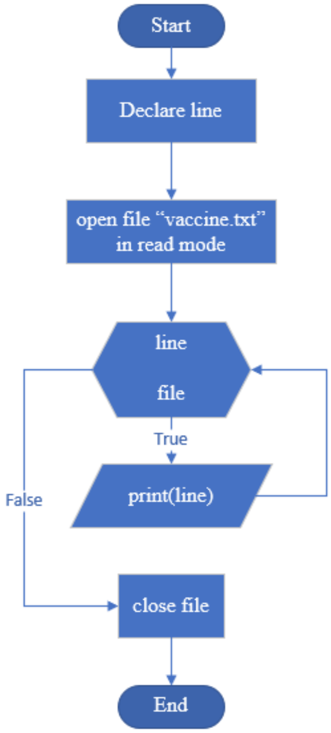
3.5 Display Distributed Vaccine

```
Start
    Declare line_count,line,dist_content,DIST_permission,DIST_code
    Declare DIST_initial_quantity,DIST_dist_amount,DIST_final_quantity
    Declare vaccine_content,i,j,index,code,command,TOT_code
    Declare TOT_distributed_quantity,num,total,highest

    index = 0
    line_count = 0
    i = 0
    j = 0
    num = 0
    highest = 0

    print("Type 1 to display all distributed vaccines' quantity.");
print("Type 2 to display chosen distributed vaccines' quantity.");
read command


open file "dist.txt" in read mode
loop line from file
    if (line_count == 0)
        dist_string_subject = line
    else
        dist_content = split (",") from line.
        DIST_permission[index] = vaccine_content[0]
        DIST_code[index] = vaccine_content[1]
        DIST_initial_quantity[index] = vaccine_content[2]
        DIST_dist_amount[index] = vaccine_content[3]
        DIST_final_quantity[index] = vaccine_content[4]
        index += 1
        line_count += 1
    End if
End loop
close file


if (command == 1)
    while (i < 5)
        while (j < index)
            if (DIST_code[j] != 0)
                code = DIST_code[j]
                TOT_code[i] = DIST_code[j]
                break
            End if
            j += 1
        End while

        j = 0
        while (j < index)
            if (DIST_code[j] == code)
                total += DIST_dist_amount[j]
                DIST_code[j] = 0
            End if
            j += 1
        End while

        TOT_distributed_quantity = total
        total = 0
        i += 1
    End while

    print("Code","Total Distributed")
    i = 0
    j = 0
    while (i < 5)
        while (j < 5)
            if (TOT_distributed_quantity[j] > num)
                num = TOT_distributed_quantity[j]
                highest = j
            End if
            j += 1
        End while

        print(TOT_code[highest])
```

```
            print(TOT_distributed_quantity[highest])
            TOT_distributed_quantity[highest] = 0
            num = 0
            highest = 0
            i += 1
        End while

    Else if (command == 2)
        print("write down the vaccine code: ")
        read code
        print(dist_string_subject)
        while (i < index)
            if DIST_code[i] == code
                print(DIST_permission[i])
                print(DIST_code[i])
                print(DIST_initial_quantity[i])
                print(DIST_dist_amount[i])
                print(DIST_final_quantity[i])
            End if
            i += 1
        End while
    End if
End
```

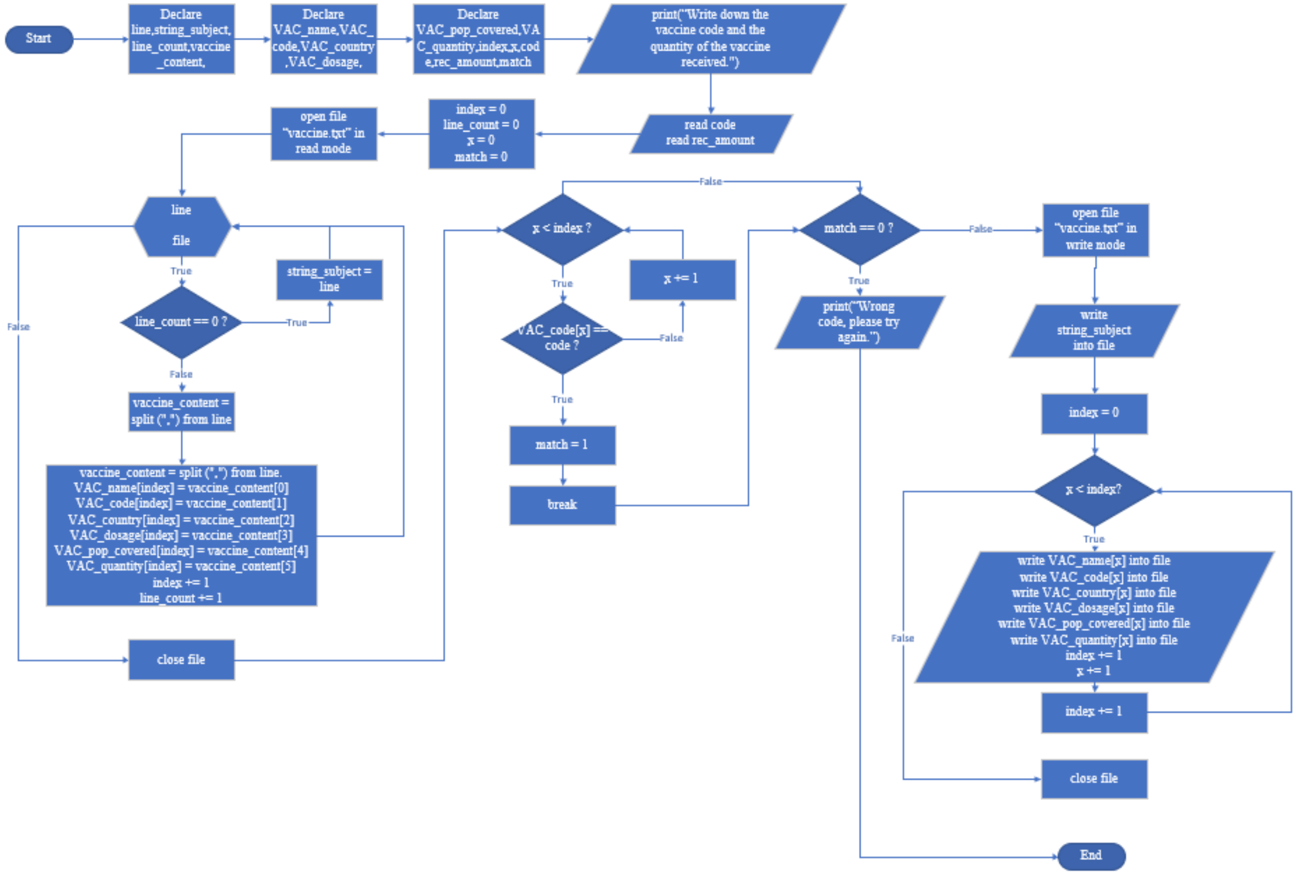
4.0 Flowchart Design

4.1 Display Vaccine



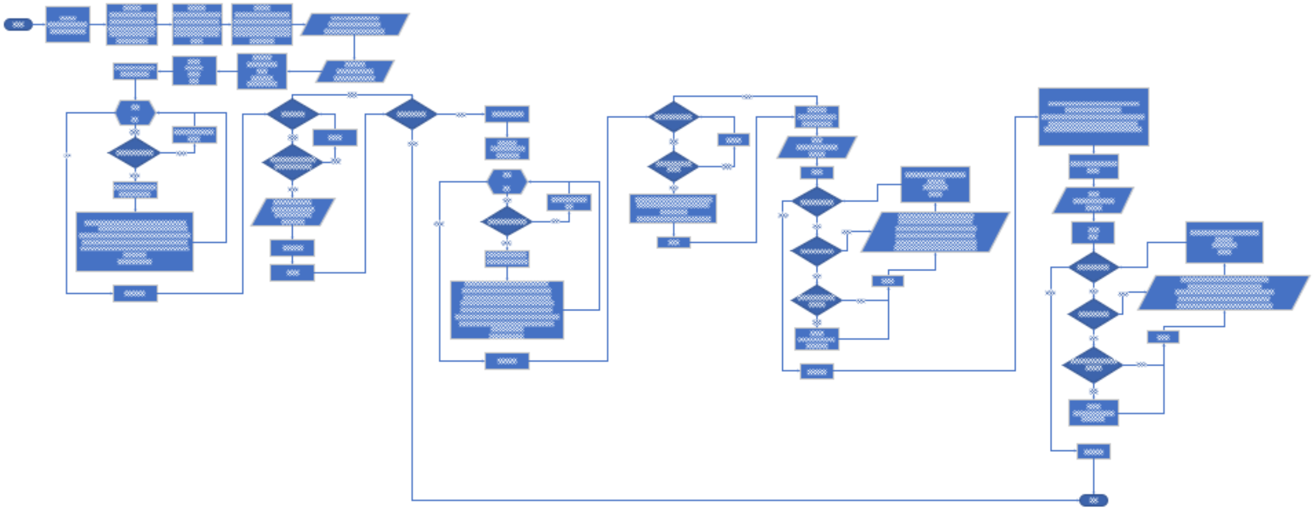
(Figure 4.1: *Display Vaccine Flowchart*)

4.2 Receive and Update Vaccine


Start
Declare line,string_subject, line_count,vaccine _content,
Declare VAC_name,VAC_ code,VAC_country ,VAC_dosage,
Declare VAC_pop_covered,VA C_quantity,index,x,cod e,rec_amount,match
print("Write down the vaccine code and the quantity of the vaccine received.")
read code read rec_amount
index = 0 line_count = 0 x = 0 match = 0
open file "vaccine.txt" in read mode
line file
True
False
line_count == 0 ?
True
string_subject = line
False
vaccine_content = split (",") from line
vaccine_content = split (",") from line. VAC_name[index] = vaccine_content[0] VAC_code[index] = vaccine_content[1] VAC_country[index] = vaccine_content[2] VAC_dosage[index] = vaccine_content[3] VAC_pop_covered[index] = vaccine_content[4] VAC_quantity[index] = vaccine_content[5] index += 1 line_count += 1
close file
x < index ?
False
True
VAC_code[x] == code ?
False
x += 1
True
match = 1
break
match == 0 ?
True
print("Wrong code, please try again.")
False
open file "vaccine.txt" in write mode
write string_subject into file
index = 0
x < index?
True
False
write VAC_name[x] into file write VAC_code[x] into file write VAC_country[x] into file write VAC_dosage[x] into file write VAC_pop_covered[x] into file write VAC_quantity[x] into file index += 1 x += 1
index += 1
close file
End

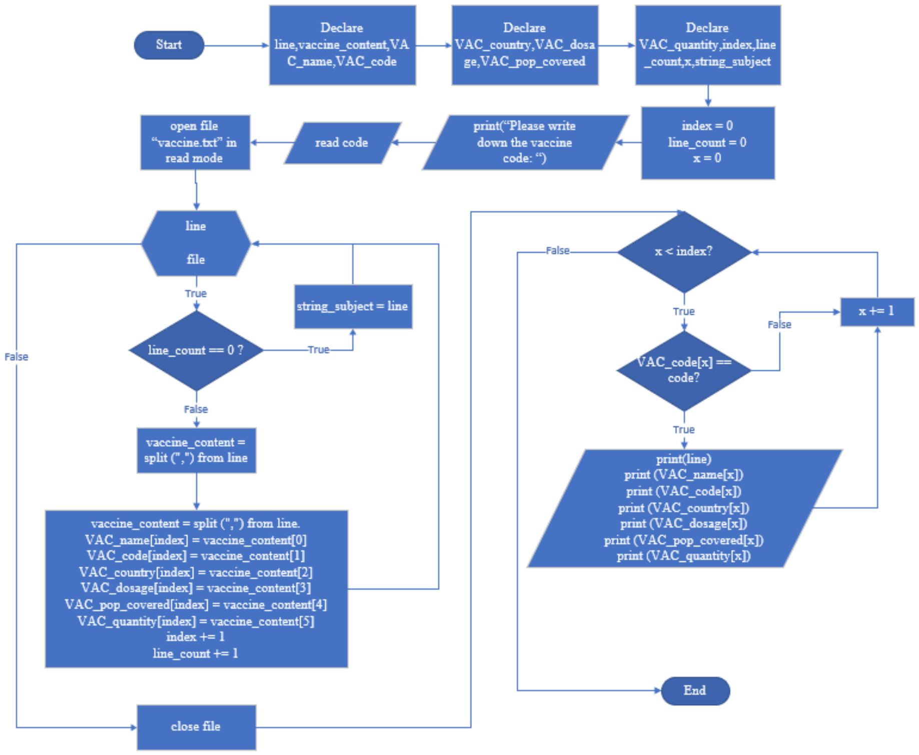
(Figure 4.2: *Receive and Update Vaccine Flowchart*)

4.3 Distribute and Update Vaccine



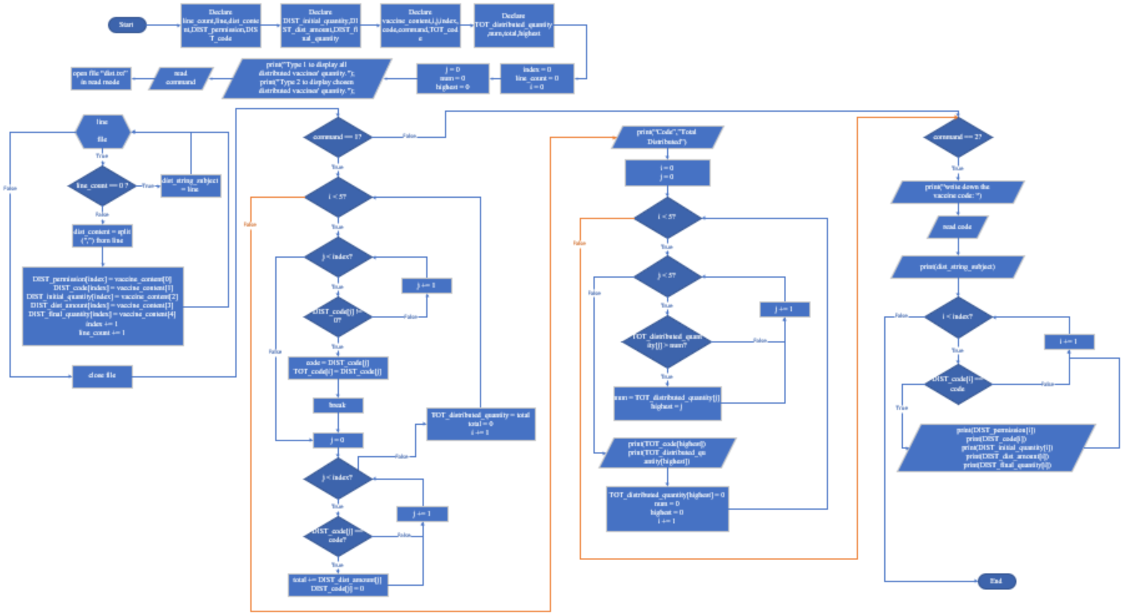
(Figure 4.3: *Distribute and Update Vaccine Flowchart*)

4.4 Search by Code


Start
Declare line,vaccine_content,VAC_name,VAC_code
Declare VAC_country,VAC_dosage,VAC_pop_covered
Declare VAC_quantity,index,line_count,x,string_subject
index = 0
line_count = 0
x = 0
print("Please write down the vaccine code: ")
read code
open file "vaccine.txt" in read mode
line
file
True
line_count == 0 ?
True
string_subject = line
False
vaccine_content = split (",") from line
vaccine_content = split (",") from line.
VAC_name[index] = vaccine_content[0]
VAC_code[index] = vaccine_content[1]
VAC_country[index] = vaccine_content[2]
VAC_dosage[index] = vaccine_content[3]
VAC_pop_covered[index] = vaccine_content[4]
VAC_quantity[index] = vaccine_content[5]
index += 1
line_count += 1
False
close file
x < index?
False
True
VAC_code[x] == code?
False
x += 1
True
print(line)
print (VAC_name[x])
print (VAC_code[x])
print (VAC_country[x])
print (VAC_dosage[x])
print (VAC_pop_covered[x])
print (VAC_quantity[x])
End

(Figure 4.4: *Search by Code Flowchart*)

4.5 Display Distributed Vaccine

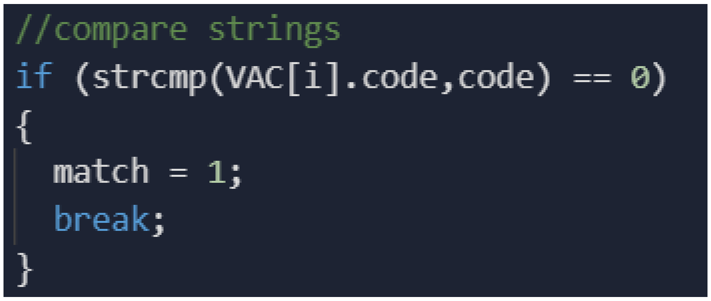

Start
Declare line_count,line,dist_content,DIST_permision,DIST_code
Declare DIST_initial_quantity,DIST_dist_amount,DIST_final_quantity
Declare vaccine_content,i,j,index,code,command,TOT_code
Declare TOT_distributed_quantity,num,total,highest
index = 0
line_count = 0
i = 0
j = 0
num = 0
highest = 0
print("Type 1 to display all distributed vaccines' quantity.");
print("Type 2 to display chosen distributed vaccines' quantity.");
read command
open file "dist.txt" in read mode
line
file
True
line_count == 0 ?
True
dist_string_subject = line
False
dist_content = split (",") from line
DIST_permision[index] = vaccine_content[0]
DIST_code[index] = vaccine_content[1]
DIST_initial_quantity[index] = vaccine_content[2]
DIST_dist_amount[index] = vaccine_content[3]
DIST_final_quantity[index] = vaccine_content[4]
index += 1
line_count += 1
False
close file
command == 1?
False
True
i < 5?
False
True
j < index?
True
DIST_code[j] != 0?
False
j += 1
True
False
code = DIST_code[j]
TOT_code[i] = DIST_code[j]
break
j = 0
False
TOT_distributed_quantity = total
total = 0
i += 1
j < index?
True
DIST_code[j] == code?
False
j += 1
True
total += DIST_dist_amount[j]
DIST_code[j] = 0
print("Code","Total Distributed")
i = 0
j = 0
i < 5?
False
True
j < 5?
False
True
TOT_distributed_quantity[j] > num?
False
j += 1
True
num = TOT_distributed_quantity[j]
highest = j
print(TOT_code[highest])
print(TOT_distributed_quantity[highest])
TOT_distributed_quantity[highest] = 0
num = 0
highest = 0
i += 1
command == 2?
True
print("write down the vaccine code: ")
read code
print(dist_string_subject)
i < index?
False
True
DIST_code[i] == code
False
i += 1
True
print(DIST_permission[i])
print(DIST_code[i])
print(DIST_initial_quantity[i])
print(DIST_dist_amount[i])
print(DIST_final_quantity[i])
End

(Figure 4.5: *Display Distributed Vaccine Flowchart*)

5.0 Additional Features

5.1 Header <stdlib.h>

5.1.1 strcmp function



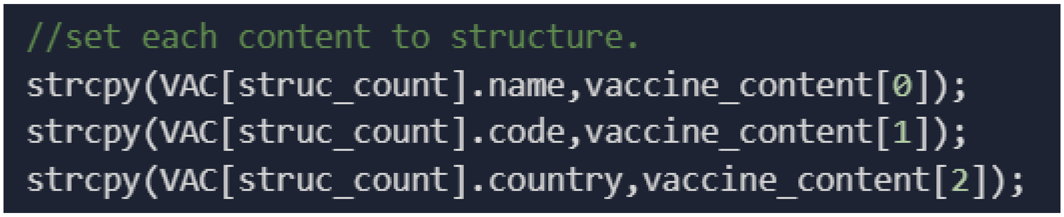
```
//compare strings
if (strcmp(VAC[i].code,code) == 0)
{
  match = 1;
  break;
}
```

(Figure 5.1.1)

As the comment specifies, strcmp is used to compare between two strings. In this case, string VAC[i].code is comparing with string code. This specific code verify if the strings are identical.

1. If identical, returns 0.
2. if VAC[i].code < code, returns value less than 0.
3. if VAC[i].code > code, returns value higher than 0.

5.1.2 strcpy function

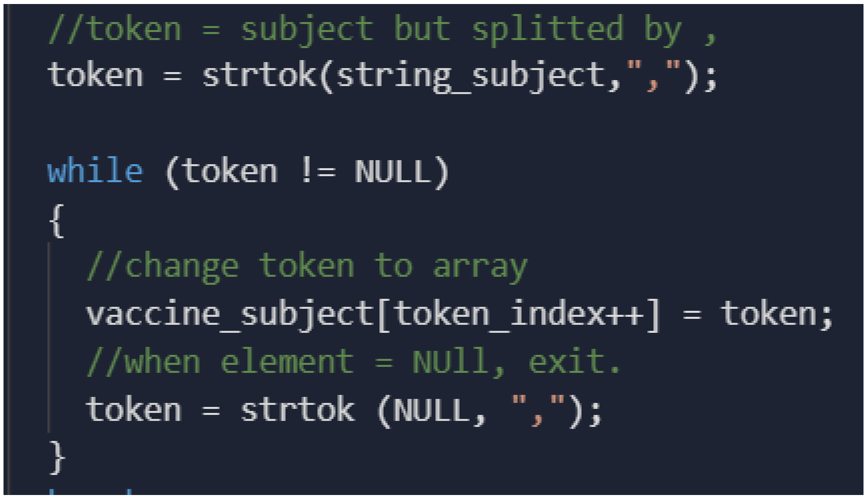


```
//set each content to structure.
strcpy(VAC[struc_count].name,vaccine_content[0]);
strcpy(VAC[struc_count].code,vaccine_content[1]);
strcpy(VAC[struc_count].country,vaccine_content[2]);
```

(Figure 5.1.2)

strcpy or also known as string copy is to copy string. strcpy accepts two arguments, for example if its strcpy(str1,str2), this function replaces str1 with str 2.

5.1.3 strtok function



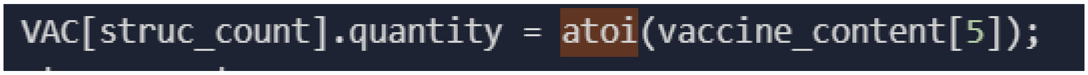
```
//token = subject but splitted by ,
token = strtok(string_subject,",");

while (token != NULL)
{
  //change token to array
  vaccine_subject[token_index++] = token;
  //when element = NUll, exit.
  token = strtok (NULL, ",");
}
```

(Figure 5.1.3)

Referring to Figure 5.1.3, strtok takes the string in string_subject and breaks it down into smaller strings using the delimiter ",", these small strings are called tokens. This function returns a pointer to the first token found in the string. (tutorialspoint, n.d.) strtok is called the second time with NULL in its argument because subsequent calls after the first call of token must have NULL as its first argument to notify the function to continue tokenizing or forwarding to the next token of the previous string.

5.2 Header <string.h>

5.2.1 atoi function

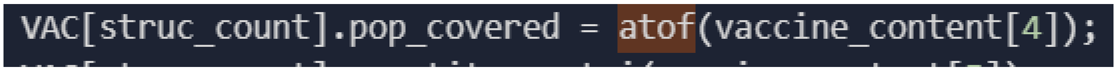


```
VAC[struc_count].quantity = atoi(vaccine_content[5]);
```

(Figure 5.2.1)

atoi converts string to integer. If no valid conversion is possible, it returns 0.

5.2.2 atof function

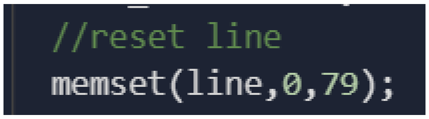


```
VAC[struc_count].pop_covered = atof(vaccine_content[4]);
```

(Figure 5.2.2)

atof converts string into float. Only numbers can be converted through atof. If no valid conversion is possible, such as the argument being characters, it will return (0.0).

5.2.3 memset function

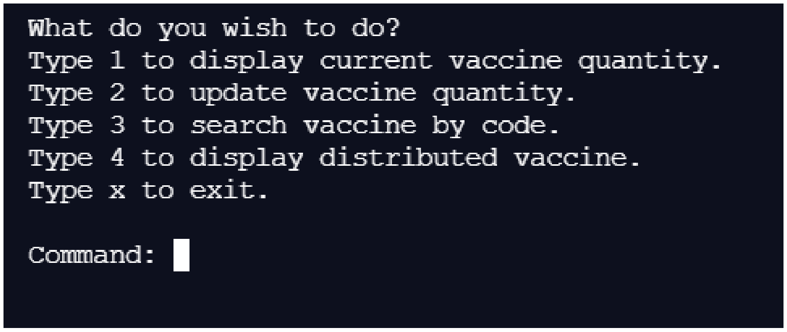


```
//reset line
memset(line,0,79);
```

(Figure 5.2.3)

memset accepts 3 arguments, which we will refer it to (X, Y, Z). X represents an array, in Figure 5.2.3, its a character array. Y represents value to be set. While Z represents size that will be replaced by Y. In normal circumstances, Y would be a character, but in Figure 5.2.3 it is 0 which is to reset the line or clearing the array.

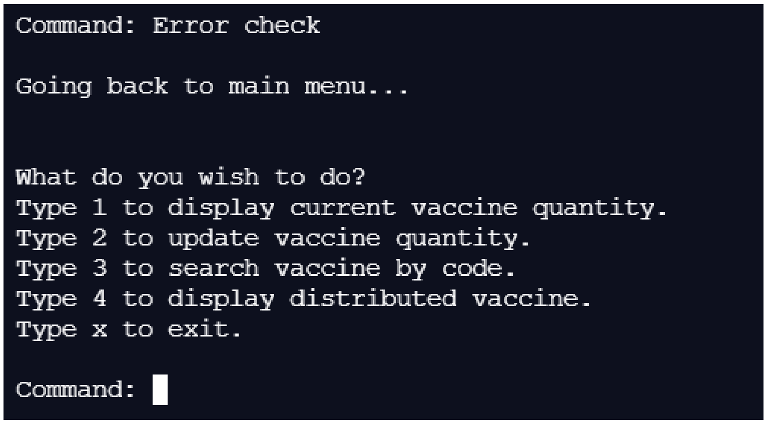
6.0 Sample Input and Output



```
What do you wish to do?
Type 1 to display current vaccine quantity.
Type 2 to update vaccine quantity.
Type 3 to search vaccine by code.
Type 4 to display distributed vaccine.
Type x to exit.

Command: █
```

(Figure 6.1)

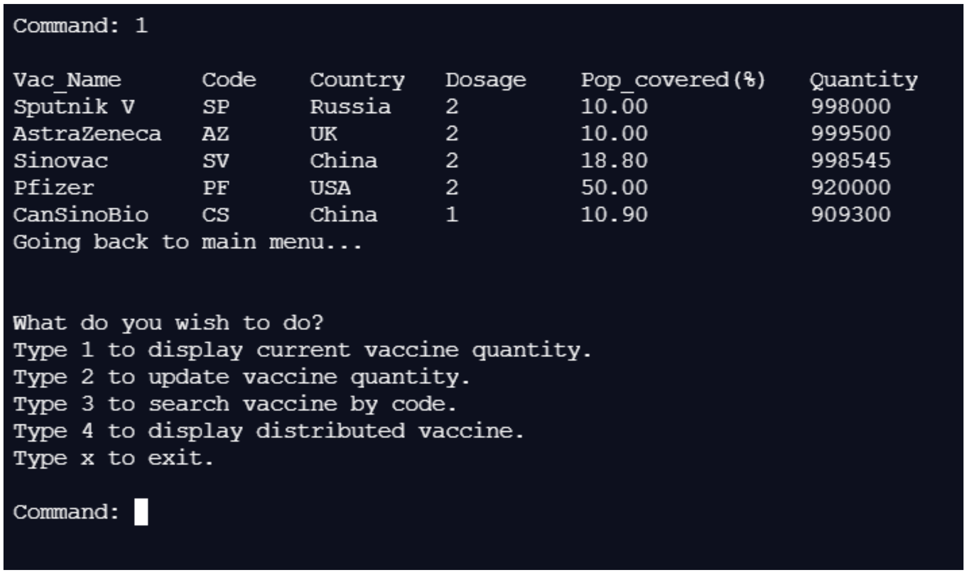
This is the starting output as soon as the program starts. This is the main menu.



```
Command: Error check

Going back to main menu...


What do you wish to do?
Type 1 to display current vaccine quantity.
Type 2 to update vaccine quantity.
Type 3 to search vaccine by code.
Type 4 to display distributed vaccine.
Type x to exit.

Command: █
```

(Figure 6.2)

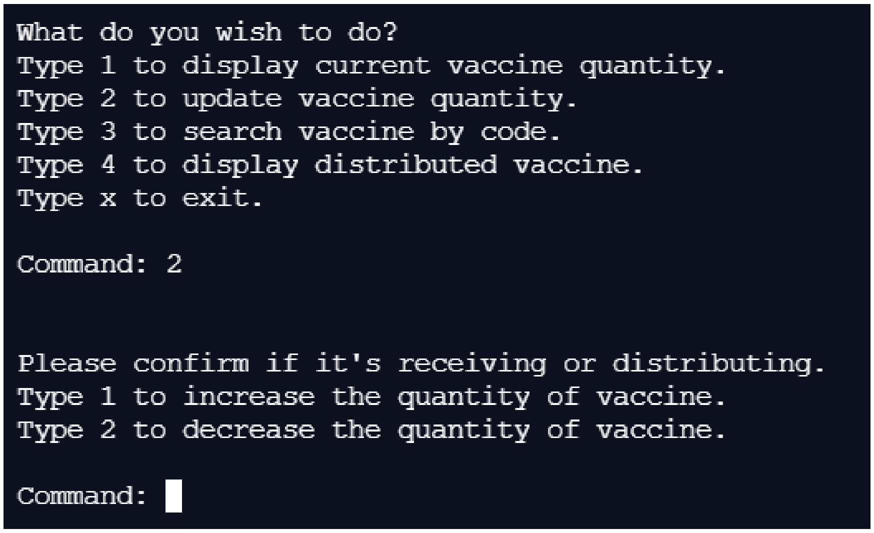
Error Check.



```
Command: 1

Vac_Name      Code    Country   Dosage    Pop_covered(%)   Quantity
Sputnik V     SP      Russia    2         10.00            998000
AstraZeneca   AZ      UK        2         10.00            999500
Sinovac       SV      China     2         18.80            998545
Pfizer        PF      USA       2         50.00            920000
CanSinoBio    CS      China     1         10.90            909300
Going back to main menu...


What do you wish to do?
Type 1 to display current vaccine quantity.
Type 2 to update vaccine quantity.
Type 3 to search vaccine by code.
Type 4 to display distributed vaccine.
Type x to exit.

Command:
```

(Figure 6.3)

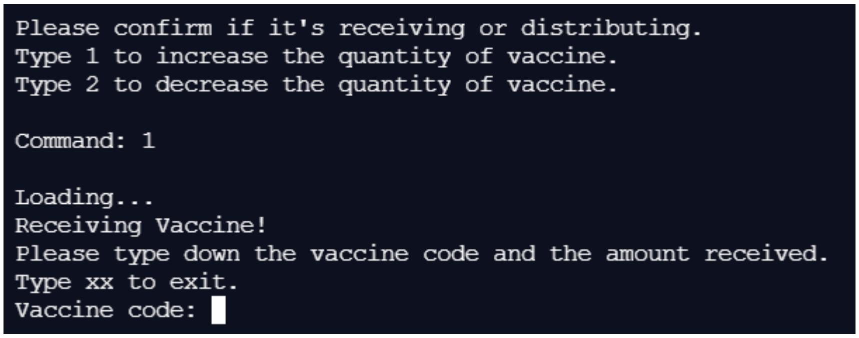
Command number 1 is display current vaccine quantity. Vaccine details are displayed.



```
What do you wish to do?
Type 1 to display current vaccine quantity.
Type 2 to update vaccine quantity.
Type 3 to search vaccine by code.
Type 4 to display distributed vaccine.
Type x to exit.

Command: 2


Please confirm if it's receiving or distributing.
Type 1 to increase the quantity of vaccine.
Type 2 to decrease the quantity of vaccine.

Command:
```

(Figure 6.4)

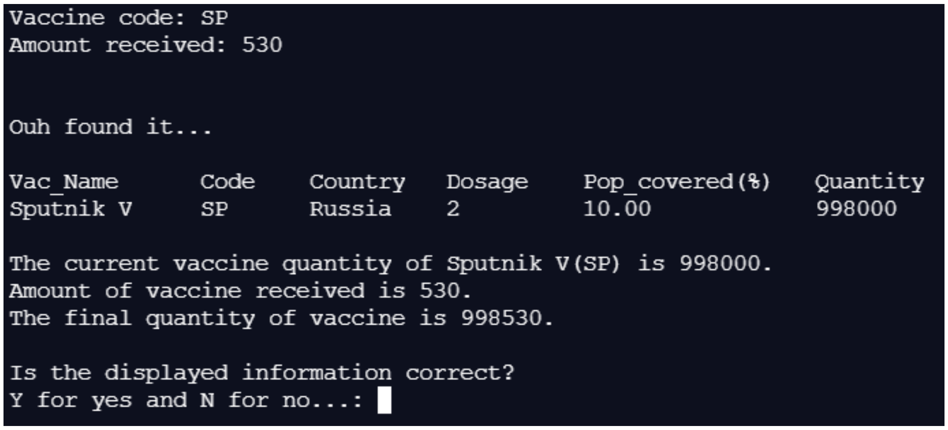
Command 2 concerns updating vaccine quantity. It directs user to specify if it’s receive or distribute vaccine.



```
Please confirm if it's receiving or distributing.
Type 1 to increase the quantity of vaccine.
Type 2 to decrease the quantity of vaccine.

Command: 1

Loading...
Receiving Vaccine!
Please type down the vaccine code and the amount received.
Type xx to exit.
Vaccine code: 
```

(Figure 6.5)

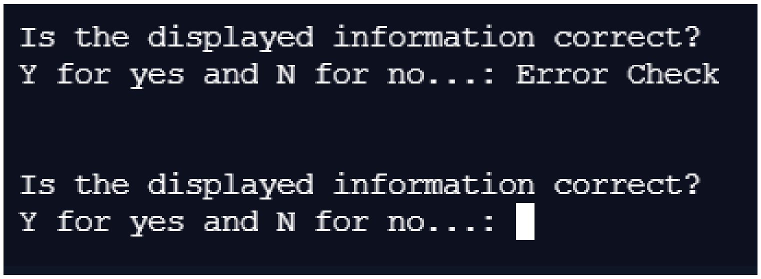
Since 1 is entered, it inquires for vaccine code and amount received.



```
Vaccine code: SP
Amount received: 530


Ouh found it...

Vac_Name      Code    Country   Dosage    Pop_covered(%)   Quantity
Sputnik V     SP      Russia    2         10.00            998000

The current vaccine quantity of Sputnik V(SP) is 998000.
Amount of vaccine received is 530.
The final quantity of vaccine is 998530.

Is the displayed information correct?
Y for yes and N for no...: 
```

(Figure 6.6)

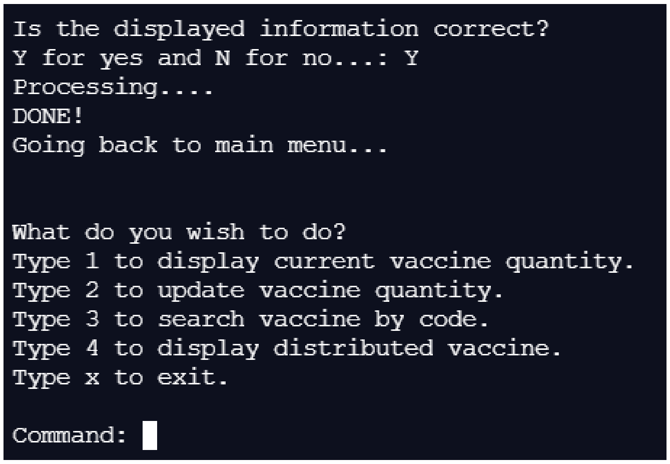
It will then display the vaccine found and details regarding vaccine quantity for validation from the user.



```
Is the displayed information correct?
Y for yes and N for no...: Error Check


Is the displayed information correct?
Y for yes and N for no...: 
```

(Figure 6.7)

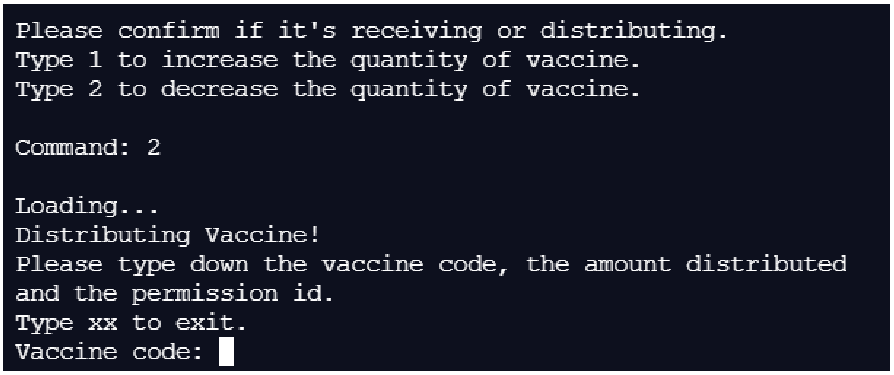
It will only accept Y or N.



```
Is the displayed information correct?
Y for yes and N for no...: Y
Processing....
DONE!
Going back to main menu...


What do you wish to do?
Type 1 to display current vaccine quantity.
Type 2 to update vaccine quantity.
Type 3 to search vaccine by code.
Type 4 to display distributed vaccine.
Type x to exit.

Command:
```

(Figure 6.8)

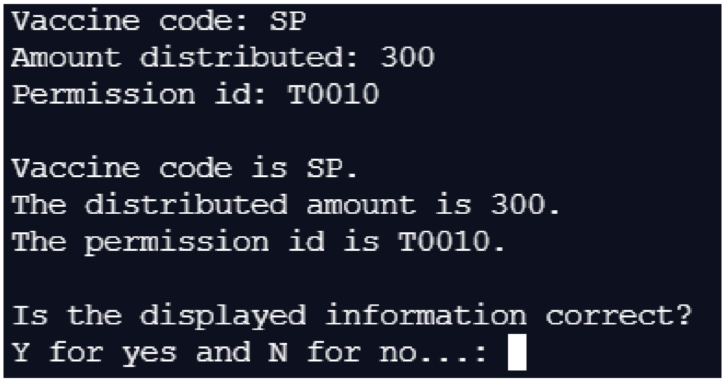
After Y is entered, it will return to main menu.



```
Please confirm if it's receiving or distributing.
Type 1 to increase the quantity of vaccine.
Type 2 to decrease the quantity of vaccine.

Command: 2

Loading...
Distributing Vaccine!
Please type down the vaccine code, the amount distributed
and the permission id.
Type xx to exit.
Vaccine code:
```

(Figure 6.9)

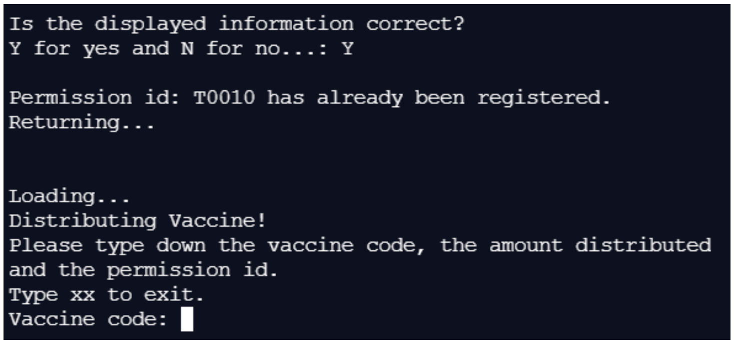
Going back to Figure 6.4 and typing command 2 for vaccine distribution.



```
Vaccine code: SP
Amount distributed: 300
Permission id: T0010

Vaccine code is SP.
The distributed amount is 300.
The permission id is T0010.

Is the displayed information correct?
Y for yes and N for no...:
```

(Figure 6.10)

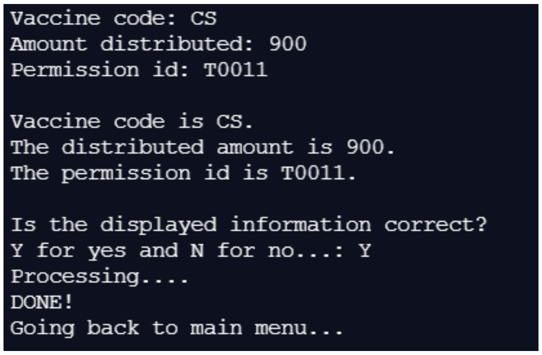
After the details is entered, it requires another validation from user.



```
Is the displayed information correct?
Y for yes and N for no...: Y

Permission id: T0010 has already been registered.
Returning...


Loading...
Distributing Vaccine!
Please type down the vaccine code, the amount distributed
and the permission id.
Type xx to exit.
Vaccine code: 
```

(Figure 6.11)

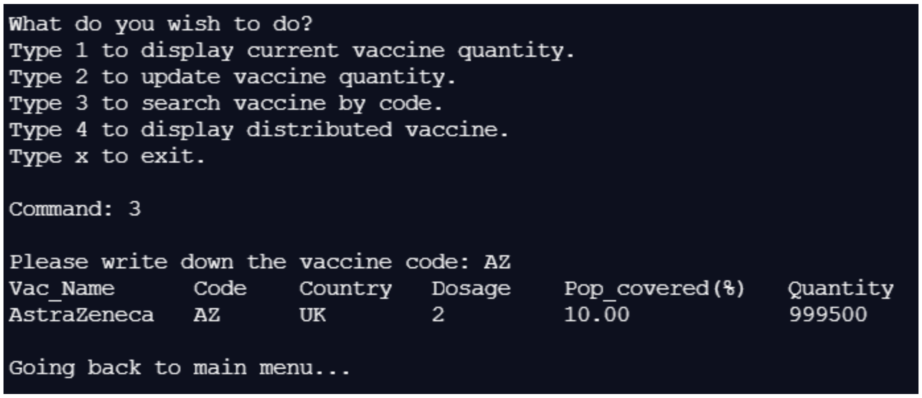
This system has a validation built to detect if the same permission id is entered. It returns to the previous menu.



```
Vaccine code: CS
Amount distributed: 900
Permission id: T0011

Vaccine code is CS.
The distributed amount is 900.
The permission id is T0011.

Is the displayed information correct?
Y for yes and N for no...: Y
Processing....
DONE!
Going back to main menu...
```

(Figure 6.12)

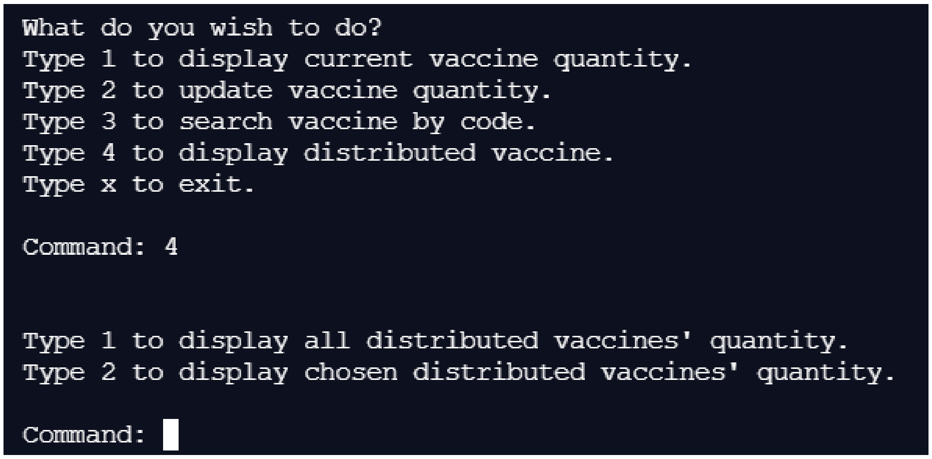
It returns to main menu when it is done.



```
What do you wish to do?
Type 1 to display current vaccine quantity.
Type 2 to update vaccine quantity.
Type 3 to search vaccine by code.
Type 4 to display distributed vaccine.
Type x to exit.

Command: 3

Please write down the vaccine code: AZ
Vac_Name      Code    Country    Dosage    Pop_covered(%)    Quantity
AstraZeneca   AZ      UK         2         10.00             999500

Going back to main menu...
```

(Figure 6.13)

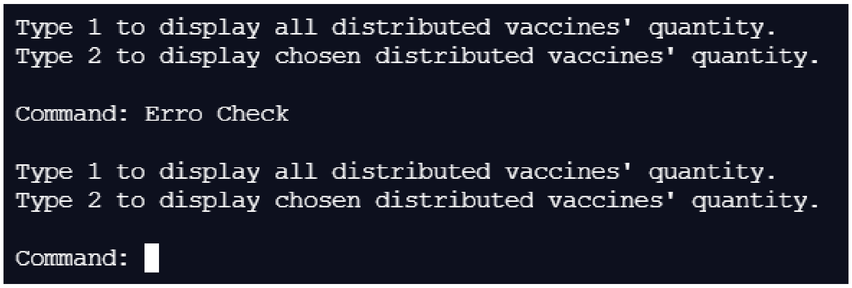
Searching vaccine by code.



```
What do you wish to do?
Type 1 to display current vaccine quantity.
Type 2 to update vaccine quantity.
Type 3 to search vaccine by code.
Type 4 to display distributed vaccine.
Type x to exit.

Command: 4


Type 1 to display all distributed vaccines' quantity.
Type 2 to display chosen distributed vaccines' quantity.

Command:
```

(Figure 6.14)

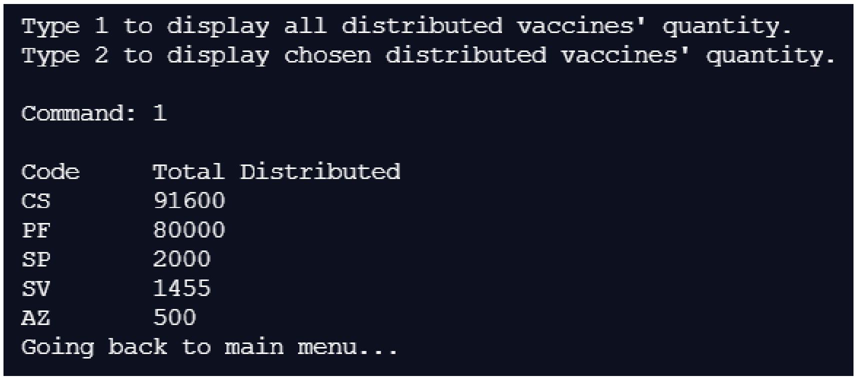
There are two selections for details regarding distributed vaccines.



```
Type 1 to display all distributed vaccines' quantity.
Type 2 to display chosen distributed vaccines' quantity.

Command: Erro Check

Type 1 to display all distributed vaccines' quantity.
Type 2 to display chosen distributed vaccines' quantity.

Command:
```

(Figure 6.15)

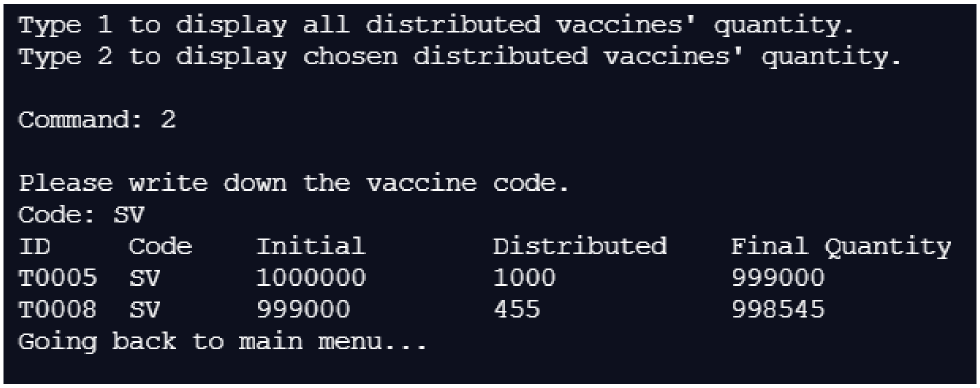
Error check.



```
Type 1 to display all distributed vaccines' quantity.
Type 2 to display chosen distributed vaccines' quantity.

Command: 1

Code     Total Distributed
CS       91600
PF       80000
SP       2000
SV       1455
AZ       500
Going back to main menu...
```

(Figure 6.16)

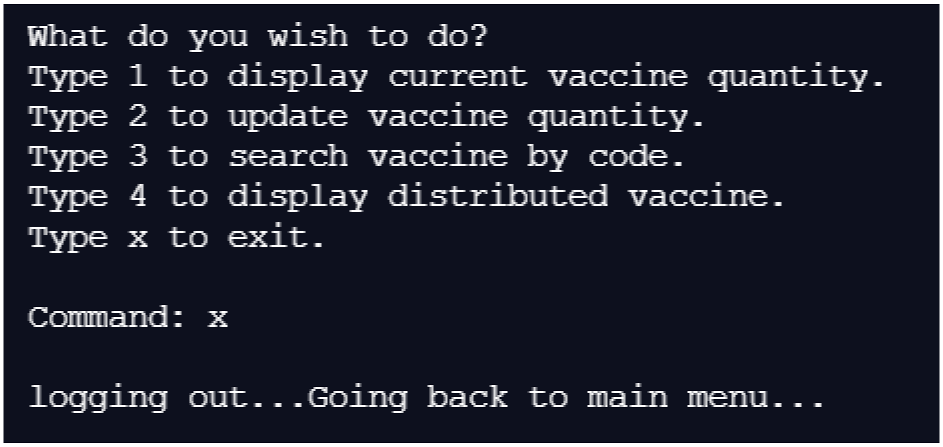
Displays all vaccine code and their total distributed quantity.



```
Type 1 to display all distributed vaccines' quantity.
Type 2 to display chosen distributed vaccines' quantity.

Command: 2

Please write down the vaccine code.
Code: SV
ID      Code    Initial         Distributed     Final Quantity
T0005   SV      1000000         1000            999000
T0008   SV      999000          455             998545
Going back to main menu...
```

(Figure 6.17)

Command 2 allows user to search for distribution record according to the inputted vaccine code.



```
What do you wish to do?
Type 1 to display current vaccine quantity.
Type 2 to update vaccine quantity.
Type 3 to search vaccine by code.
Type 4 to display distributed vaccine.
Type x to exit.

Command: x

logging out...Going back to main menu...
```

(Figure 6.18)

Logging out.

Conclusion

The pharmaceutical company will have no problem functioning the system as it is menu based and extremely user friendly. Each step is labeled in detailed, every content is displayed tidily, and will indicate when a process is done. Furthermore, all the requirement specified by the company is embedded within the system. Not only that, in the case of wrong input, the system is fully capable of handling it by either returning to previous menu or informs the user regarding the error. Moreover, although not specified, a validation regarding permission id was added to the dist.txt as a precaution for when the same process is entered twice. If and when the same permission id was entered, the system will automatically reject the process. All in all, the system is surely up to the company’s standard.

References

Banex, n.d. *Why do we use NULL in strtok()?.* [Online]
Available at: https://stackoverflow.com/questions/23456374/why-do-we-use-null-in-strtok/23456549
[Accessed 5 6 2021].

programiz, n.d. *C strcmp().* [Online]
Available at: https://www.programiz.com/c-programming/library-function/string.h/strcmp
[Accessed 1 6 2021].

tutorialspoint, n.d. *C library function - atof().* [Online]
Available at: https://www.tutorialspoint.com/c_standard_library/c_function_atof.htm
[Accessed 17 6 2021].

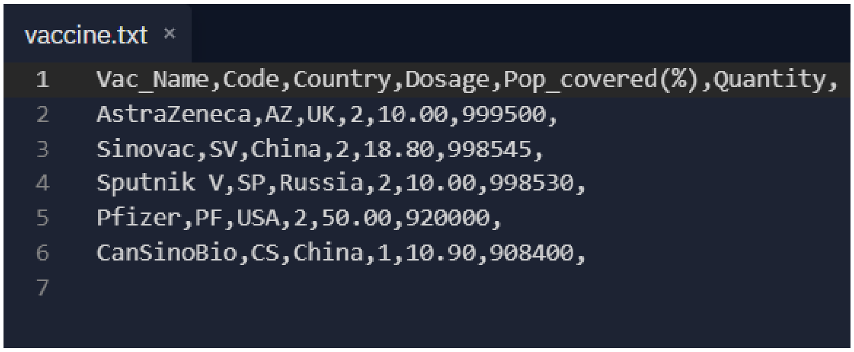
tutorialspoint, n.d. *C library function - atoi().* [Online]
Available at: https://www.tutorialspoint.com/c_standard_library/c_function_atoi.htm
[Accessed 5 6 2021].

tutorialspoint, n.d. *C library function - memset().* [Online]
Available at: https://www.tutorialspoint.com/c_standard_library/c_function_memset.htm
[Accessed 18 6 2021].

tutorialspoint, n.d. *C library function - strcpy().* [Online]
Available at: https://www.tutorialspoint.com/c_standard_library/c_function_strcpy.htm
[Accessed 17 6 2021].

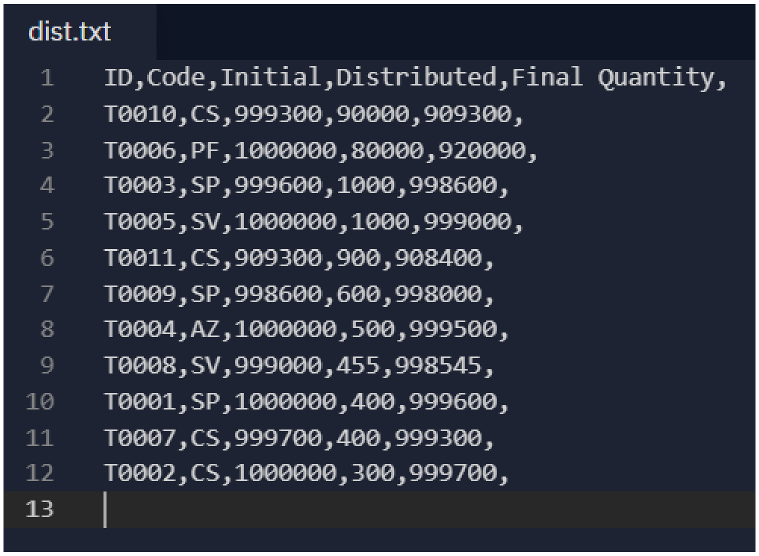
tutorialspoint, n.d. *C library function - strtok().* [Online]
Available at: https://www.tutorialspoint.com/c_standard_library/c_function_strtok.htm
[Accessed 15 6 2021].

Appendix



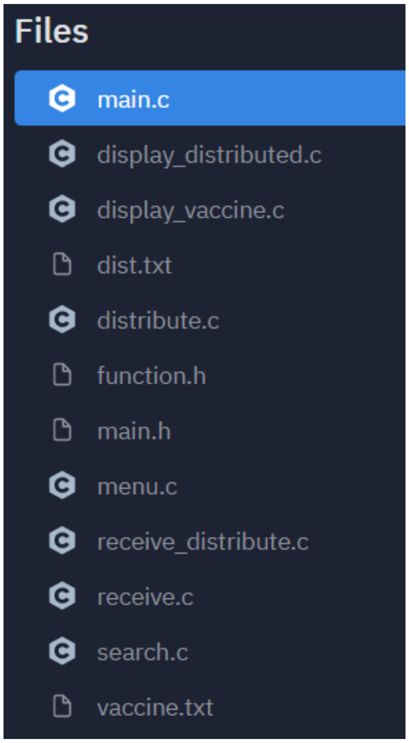
```
Vac_Name,Code,Country,Dosage,Pop_covered(%),Quantity,
AstraZeneca,AZ,UK,2,10.00,999500,
Sinovac,SV,China,2,18.80,998545,
Sputnik V,SP,Russia,2,10.00,998530,
Pfizer,PF,USA,2,50.00,920000,
CanSinoBio,CS,China,1,10.90,908400,

```

(Figure 7.1: *Vaccine.txt*)



```
ID,Code,Initial,Distributed,Final Quantity,
T0010,CS,999300,90000,909300,
T0006,PF,1000000,80000,920000,
T0003,SP,999600,1000,998600,
T0005,SV,1000000,1000,999000,
T0011,CS,909300,900,908400,
T0009,SP,998600,600,998000,
T0004,AZ,1000000,500,999500,
T0008,SV,999000,455,998545,
T0001,SP,1000000,400,999600,
T0007,CS,999700,400,999300,
T0002,CS,1000000,300,999700,
```

(Figure 7.2: *Dist.txt*)



(Figure 7.3: *All Files*)