UTM

UNIVERSITI TEKNOLOGI MALAYSIA

MINI PROJECT

DATA STRUCTURE AND ALGORITHM

SECJ 2013 – 02

CASE STUDY: VACCINE QUEUE SIMULATION SYSTEM FOR ELDERLY

Video presentation: https://youtu.be/Lc7_FRte7_w

GROUP NO: 6

NO.	NAME	MATRIC NUMBER
1	MADINA SURAYA BINTI ZHARIN	A20EC0203
2	GOO YE JUI	A20EC0191
3	HONG PEI GEOK	A20EC0044
4	QAISARA BINTI ROHZAN	A20EC0133

Table of Contents

Part 1 Project synopsis and objectives

1.1 Synopsis

In this project, we will develop a vaccine queue simulation for elderly. This system will have two users who are staff and customers. Staff is the person to manage the vaccine queue simulation system while customers can register themselves through the system. Basically, if the user is a staff, he can view the current list of vaccination, search for a particular customer, set priority for pending customers based on age and print out the highest priority customer. For customers, they can register in order to be listed in the pending queue list, cancel the registration and view the information registered.

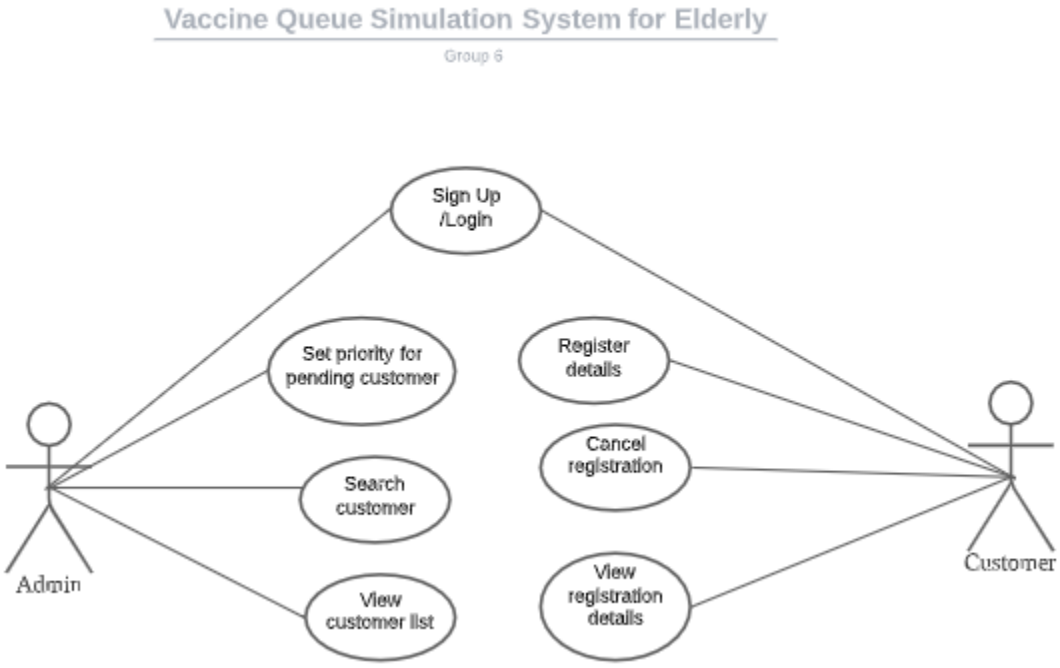
Overall, the data structure used in this project is queue which involves the operations enqueue, dequeue, searching and display. The customers' information that will be stored in the queue list are name, age, identification card (IC) and address. In order to ensure the security, if the customer wants to cancel and display his personal information, he or she must key in the IC to verify.

1.2 Objectives

- To simplify the process of vaccination arrangement for elderly
- Apply linked list implementation of queue to store the data of customers

Part 2 System requirement, analysis and design

2.1 Use Case Diagram

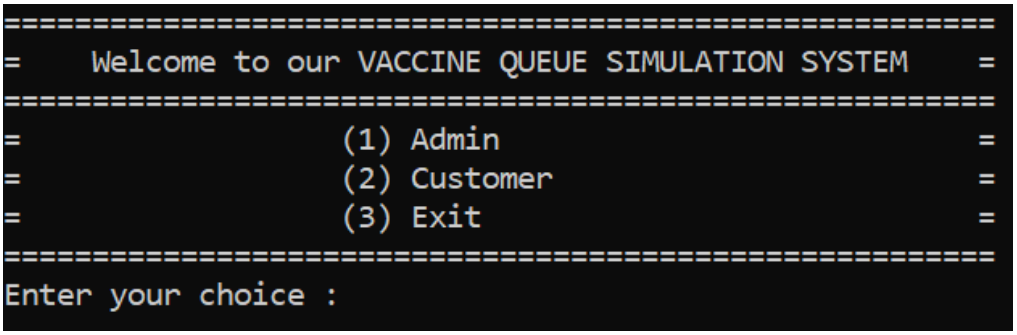
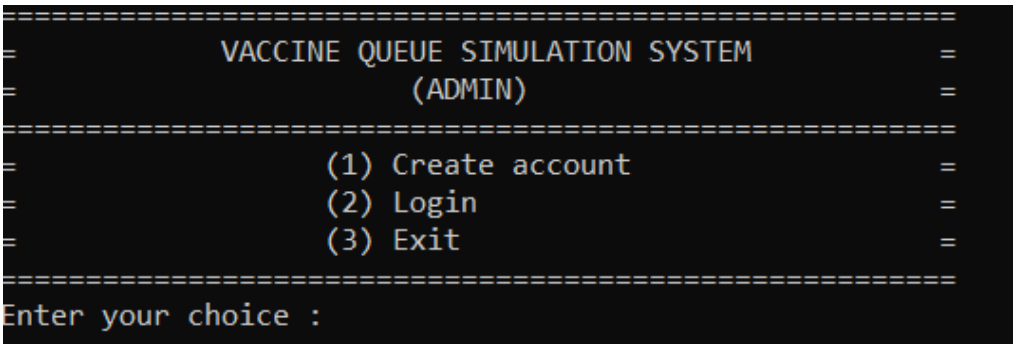
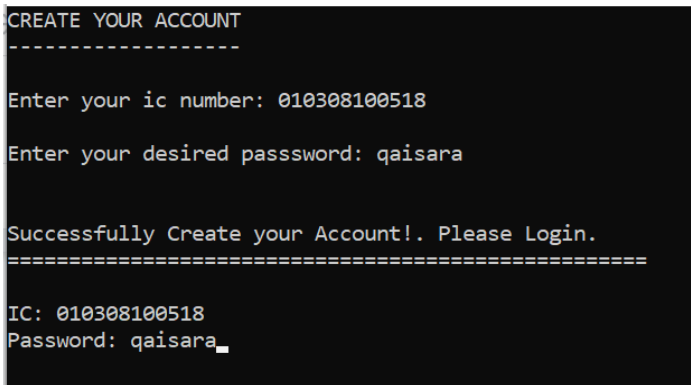


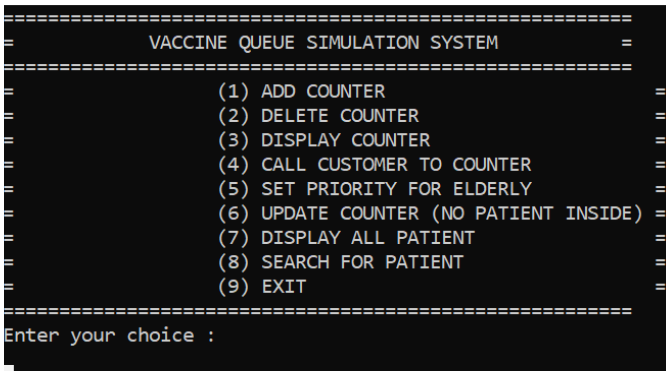
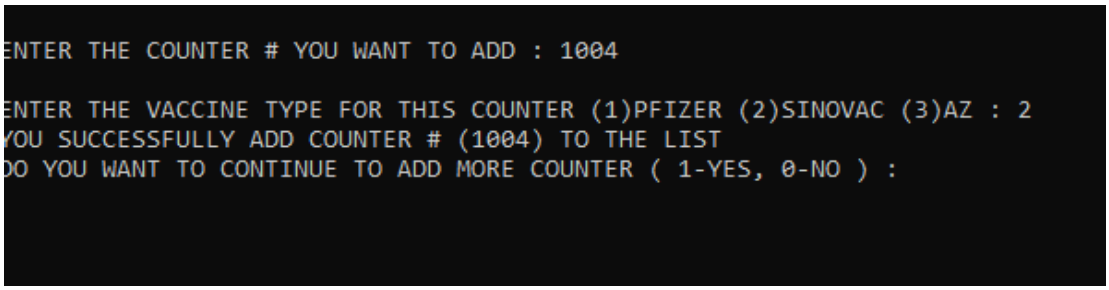
2.2 Use Case Description

Actor	Task
Admin	Admin firstly needs to sign up and login into the system. Then, they could search for customers and view customer lists. Next, set priority for pending customers and view most priority customers based on age for vaccine.
Customer	Customer firstly needs to sign up and login into the system. Then, they could register and view their own details entered.. If they want to cancel, registration details will be deleted.

Use Case	Purpose
Sign up / Login	Create an account by entering ic number and desired password. Upon having an account, enter ic number and password into the system.
Set priority for pending customer	Set priority for pending customers based on age. Older people will be prioritized.
Search customer	Search for customers/patients who register for vaccination.
View customer list	View all customers/patients list who register for vaccination.
Register details	Register own details (Name, ic number, age, address, and contact number)
Cancel appointment	Remove registered details.
View appointment	View registered details.

Part 3 Development code steps/activities

<table>
<tr><th colspan="2">Admin View</th></tr>
<tr><th>Steps</th><th>Output and Explanation</th></tr>
<tr><td rowspan="2">1.</td><td> 
</td></tr>
<tr><td>The above picture shows the first interface of the system, which is the main menu. Three options being displayed, either admin (1), customer (2) or exit (3).</td></tr>
<tr><td rowspan="2">2.</td><td> 
</td></tr>
<tr><td><u>Main Menu for Admin (Choice 1)</u>
Admin can create their account (1) as a first time user. If they choose to login (2), they can login into the system using their current account. Else, exit from the system.</td></tr>
<tr><td>3.</td><td> 
</td></tr>
</table>

	Create Account for Admin (Choice 1) The above picture shows an admin creating an account form for the first time. After registering, they can login onto the Vaccine Queue Simulation System. If the login input is the same as register input, login will be successful.
4.	
	Login for Admin (Choice 1) If the user logs in as a customer, the above page will show and provide some options which are Register for Add Vaccine Counter (1), Delete Vaccine Counter (2), Display Vaccine Counter (3),Call Customer to Counter (4), Set Priority For Elderly (5), Update Vaccine Counter (6), Display all Patients(7), Search for Patient (8) and Exit (9).
5.	
	Add Counter (Choice 1) This option enable admin to add another vaccine counter to the PPV
6.	**Delete Counter (Choice)** This option enable admin to delete any current vaccine counter of the PPV
7.	**Display Counter (Choice 3)** This option enable admin to display all the vaccine counter of the PPV
8.	**Call Customer to Counter (Choice 4)** This option enable admin to display all the vaccine counter of the PPV

9.

```
The latest registered customer is 123
Are you sure you want to set priority for this customer? (1)YES  (0)NO
1

ENTER THE PRIORITY NUMBER YOU WANT THE CUSTOMER TO BE SET : 1

Press any key to continue...
```

Set Priority for Elderly (Choice 5)

This option enables the admin to set priority for elderly in the queue. This option can only set the latest registered customer.

10.

```
ENTER THE COUNTER # YOU WANT TO UPDATE : 1001
COUNTER # 1001 IS SUCCESSFULLY UPDATED
DO YOU WANT TO CONTINUE TO UPDATE MORE COUNTER( 1-YES, 0-NO ) :
```

Update Counter (Choice 6)

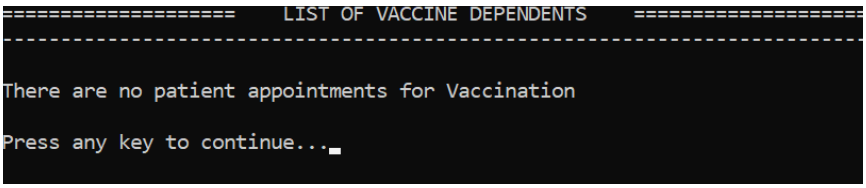
This option enables the admin to Update the counter status to available when there is no more customer at the counter.

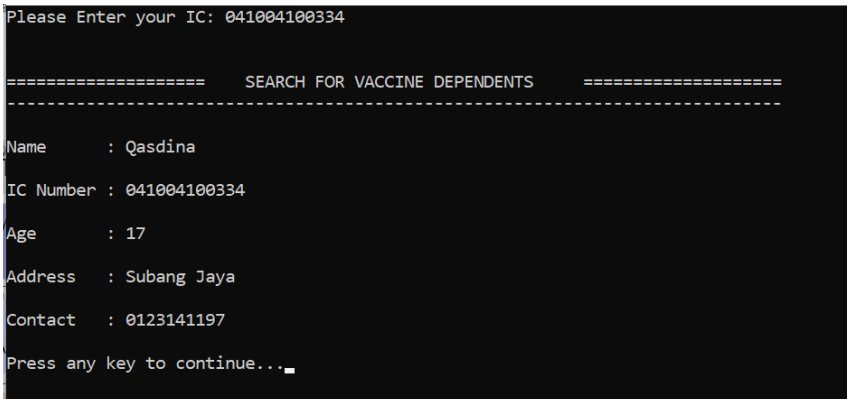
11.

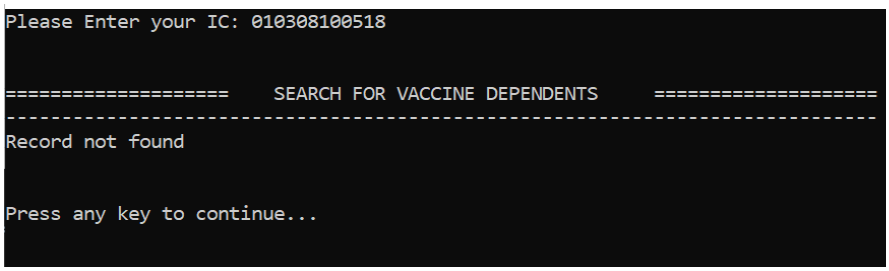
```
====================    LIST OF VACCINE DEPENDENTS    ====================
------------------------------------------------------------------------



- Qasdina, 041004100334, 17, 11, Jalan USJ 17/16, 0123141197
- Noor Azlin, 740914055002, 47, 11, Jalan USJ 17/16, 0182180203
- Rohzan, 730223015118, 49, 11, Jalan USJ 17/16, 0193389505


Press any key to continue...
```

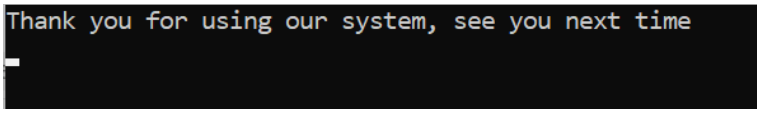
Display all Patient (Choice 7)

This option enables the admin to access the list of patients that have successfully registered for the vaccination. The list includes the name of the patients, their IC number, age, address and contact number.

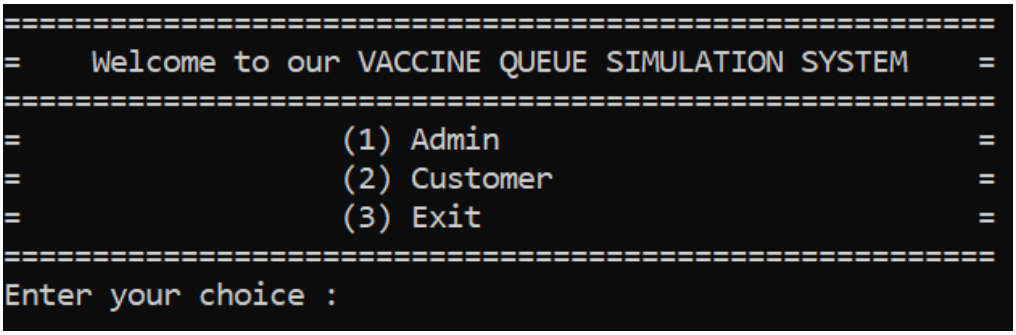
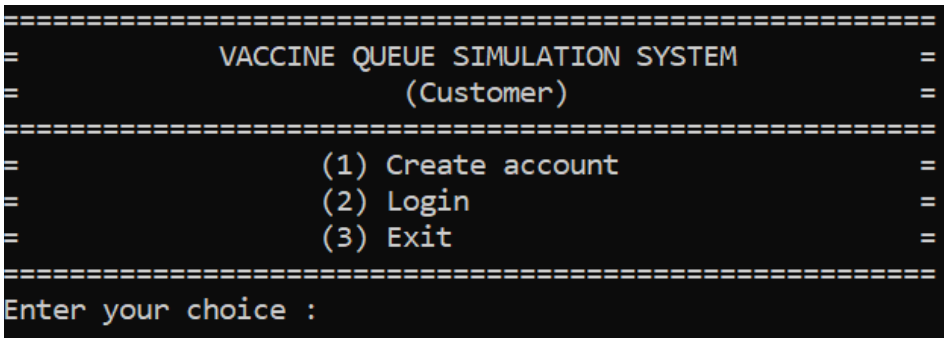
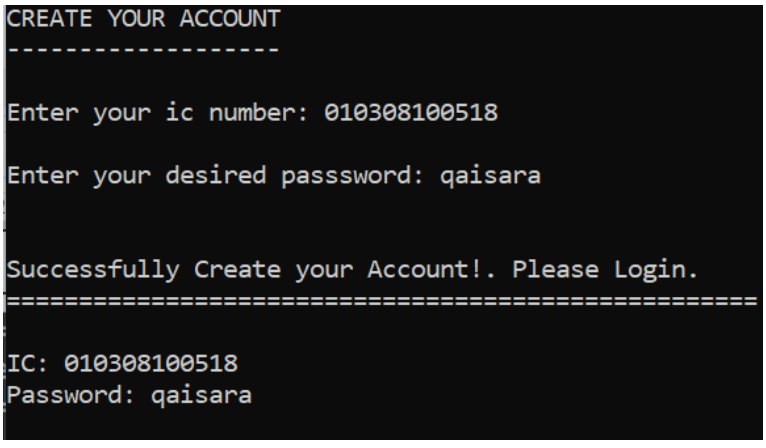
<table>
<tr><td></td><td>


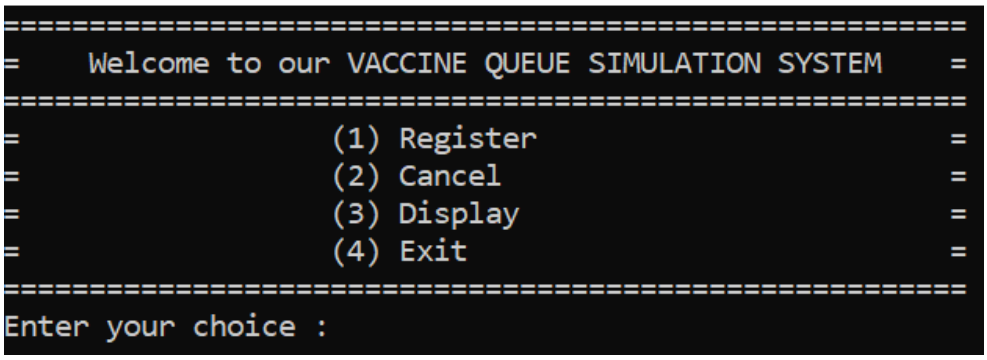
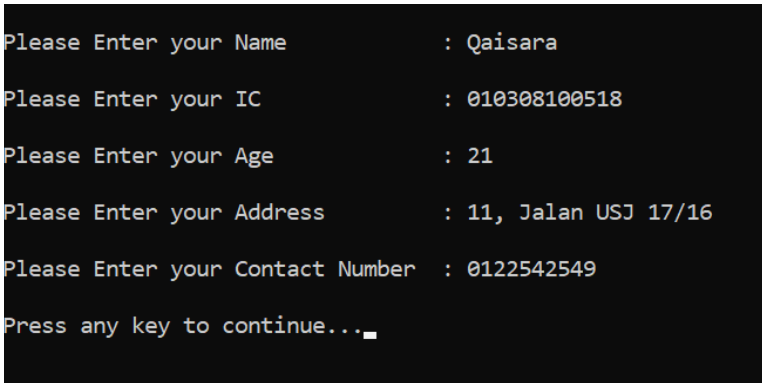
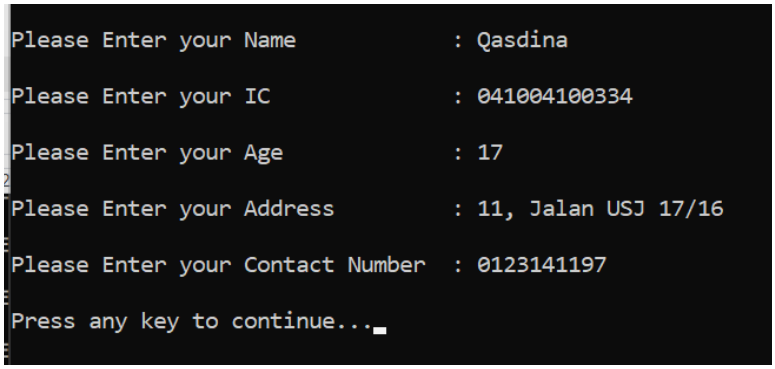
</td></tr>
<tr><td></td><td><u>If there are no patients registered by the customer</u>
An error message will be prompt if there are no registered vaccination patients in the system to notify the administrator.</td></tr>
<tr><td>12.</td><td>


</td></tr>
<tr><td></td><td><u>Search for Patient (Choice 8)</u>
This option enables the admin to search for patients that have successfully registered for the vaccination.</td></tr>
<tr><td></td><td>


</td></tr>
<tr><td></td><td><u>If there are no patients registered by the customer</u>
An error message will be prompt if there are no registered vaccination patients in the system to notify the administrator.</td></tr>
<tr><td>13.</td><td>


</td></tr>
<tr><td></td><td><u>Exit the Vaccine Queue Simulation System (Choice 9)</u></td></tr>
</table>

Customer View	
Steps	**Output and Explanation**
1.	
	The above picture shows the first interface of the system, which is the main menu. Three options being displayed, either admin (1), customer (2) or exit (3).
2.	
	<u>Main Menu for Customer (Choice 2)</u> Customer or Vaccine dependent can create their account (1) as a first time user. If they choose to login (2), they can login into the system using their current account. Else, exit from the system.
	

	Create Account for Customer (Choice 1) The above picture shows a customer creating an account form for the first time. After registering, they can login onto the Vaccine Queue Simulation System. If the login input is the same as register input, login will be successful.
	
	Login for Customer (Choice 2) If the user logs in as a customer, the above page will show and provide some options which are Register for Vaccine appointment (1), Cancel the Vaccine appointment (2), Display Vaccine Appointment (3) and Exit (4).
3.	 

```
Please Enter your Name            : Noor Azlin

Please Enter your IC              : 740914055002

Please Enter your Age             : 47

Please Enter your Address         : 11, Jalan USJ 17/16

Please Enter your Contact Number  : 0182180203

Press any key to continue...
```

```
Please Enter your Name            : Rohzan

Please Enter your IC              : 730223015118

Please Enter your Age             : 49

Please Enter your Address         : 11, Jalan USJ 17/16

Please Enter your Contact Number  : 0193389505

Press any key to continue...
```

Register for Vaccine Appointment (Choice 1)

If the customer chooses Register (1), he or she is required to fill in some information such as their name, IC, age, address and contact number.

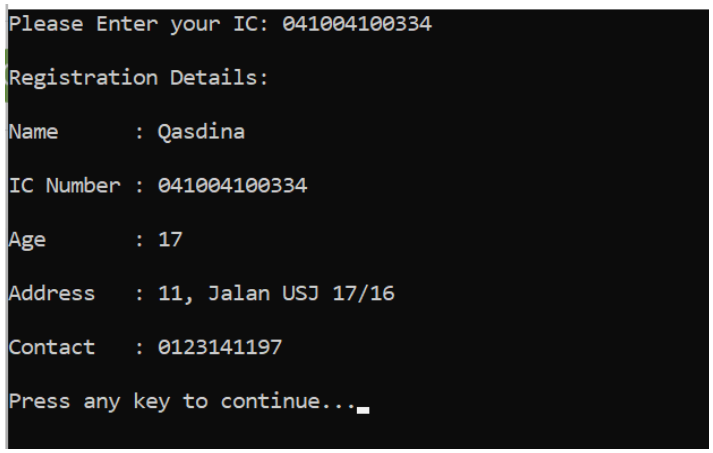
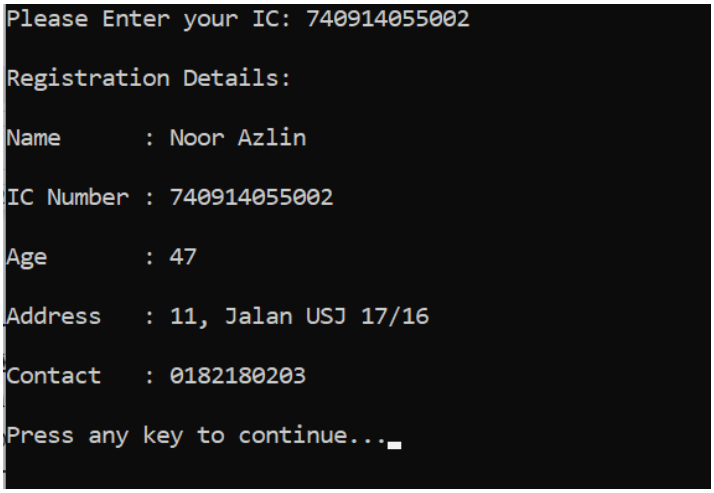
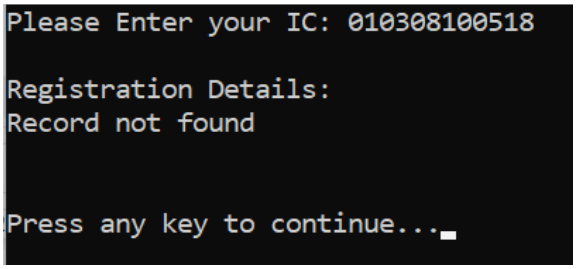
4.

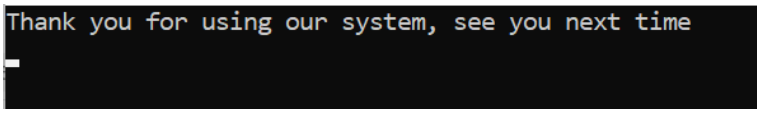
```
Please Enter your IC: 010308100518
Qaisara is deleted from vaccine registration.

Press any key to continue...
```

Cancel Vaccine Appointment (Choice 2)

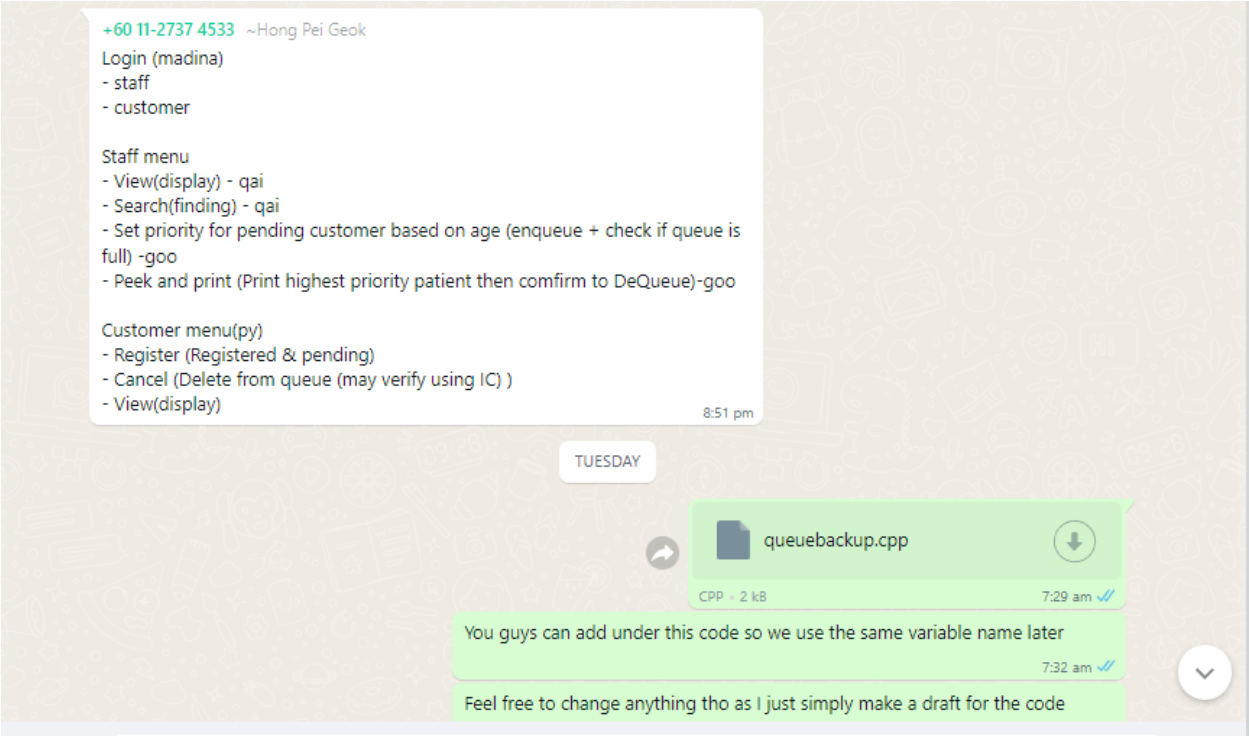
If the customer chooses Cancel (2), he or she is required to fill in some information such as their IC number to ensure security for the system to identify their Vaccine Appointment Booking and delete it. If a customer enters the correct IC number, the vaccine registration of the corresponding customer will be removed from the system.

```
Please Enter your IC: 010837465486
Record not found


Press any key to continue...
```

	If customer input wrong IC number The above figure shows the message if a customer enters the wrong IC. An error message will prompt saying 'Record Not Found'.
5.	 
	Display all Vaccine Appointment (Choice 3) If a customer wants to see his or her registration details, they can choose option 3 which is display. Before showing the personal details, the vaccine dependents should enter their IC correctly for the system to display their details. The personal details of the customer will be shown if the correct IC number is entered.
	

	<u>If customer input Cancelled Vaccine Appointment</u> Same as the cancel option, if a customer enters the wrong IC number, he or she cannot see their Vaccine Registration details.
6.	
	<u>Exit the Vaccine Queue Simulation System (Choice 4)</u>

Part 4 Other elements or efforts in completing this mini project



Due to the COVID-19 pandemic, we could not meet face-to-face to discuss this mini project. We have adapted to this tough situation by discussing it online through Whatsapp. To maximize our team working, we decided to divide the task through Whatsapp. In order to do collaborative coding, we set the header and variable together to ease the process of combining them together later.

Appendix

Code

```
#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <conio.h>

using namespace std;
/*
* Node Declaration
*/

class Register{
	public:
		string ic, password;
};

class Login{
	public:
		string inputIC, inputPass;
};

struct customer
{
	int priority;
	int age;
	string name;
	string address;
	string ic;
	string tele_num;
	customer *link;

		customer(int age,string name, string address, string ic, string tele_num)
		{
			this->age = age;
			this->name = name;
			this->address = address;
			this->ic = ic;
			this->tele_num = tele_num;
		}
```

```
                customer(){

                }
};

class Queue
{
        private:
                customer *front;
                customer *temp1;
                int prioritytemp=1;

        public:

                Queue()
                {
                        front = NULL;
                }


        //set priority for the latest registered customer
        void setpriority()
        {
                customer *q;
                int priority,choice;

                cout << "The latest registered customer is " << temp1->name << endl;

                cout << "Are you sure you want to set priority for this customer? (1)YES  (0)NO"
<< endl;  //COMFIRMATION TO SET FOR THIS CUSTOMER
                cin >> choice;


                if(choice == 1){
                        cout<<"\nENTER THE PRIORITY NUMBER YOU WANT THE
CUSTOMER TO BE SET : "; //ADMIN DECLARE PRIORITY FOR LATEST REGISTERED
CUSTOMER
                        cin>>priority;
                        temp1->priority = priority;
```

```
                        if (front == NULL || priority < front->priority) //TO SET WHEN priority
entered is less than front->priority
                        {
                                temp1->link = front;
                                front = temp1;
                        }
                        else  //TO SET WHEN priority entered is larger than front->priority
                        {
                                q = front;
                                while (q->link != NULL && q->link->priority <= priority)
                                        q=q->link;
                                temp1->link = q->link;
                                q->link = temp1;
                        }
                }else if (choice == 0){

                }else{
                        cout << "INVALID INPUT" << endl;
                }

        }

        bool dequeue()
        {
                customer *tmp=new customer;

                if(front == NULL){   // ERROR MESSAGE IF NULL IN QUEUE
                        cout<<"THERE IS NO CUSTOMER IN THE QUEUE"<<endl;
                        return false;
                }

                else  //TO CALL OUT CUSTOMER AND DEQUEUE
                {
                        int choice;
                        tmp = front;

                        cout<<"Mr/Mrs " <<tmp->name<<" Please Come Forward To Get Your
Vaccine!"<<endl;
```

```
                front = front->link;
                free(tmp);
                return true;
        }
}

//customer enqueue
void cEnqueue()
{
  string n, ic, address, hp;
  int a;
        customer *newNode = new customer;

  cin.ignore();
  cout << "\nPlease Enter your Name           : ";
  getline(cin, n);
  cout << "\nPlease Enter your IC             : ";
  getline(cin, ic);
  cout << "\nPlease Enter your Age            : ";
  cin >> a;
  cin.ignore();
  cout << "\nPlease Enter your Address        : ";
  getline(cin, address);
  cout << "\nPlease Enter your Contact Number  : ";
  cin >> hp;

  newNode->name = n;
  newNode->ic = ic;
  newNode->age = a;
  newNode->address = address;
  newNode->tele_num = hp;
  newNode->priority = prioritytemp;
  prioritytemp++;

  temp1 = newNode;

  if (front == NULL) {     //queue is empty
    front = newNode;
    front->link = NULL;
  } else {            //queue not empty
```

```
        customer *temp=front;

        while(temp->link!=NULL)
        {
            temp = temp->link;
              }
            temp->link = newNode;
            newNode->link = NULL;
            }
    }

    //customer cancel vaccine registration
    void remove(string IC) {
      customer *curr = front;
      customer *prev = NULL;

      if (front == NULL) {
        cout<<"Record not found"<<endl;
        return;
      }

            while (curr && curr->ic != IC) {
             prev = curr;
             curr = curr->link;
            }

            if(curr==NULL)
            {
                    cout << "Record not found" << endl;
                    cout<<endl;
                    return;
            }
            if (curr) {
                     if (prev) {
                      cout << curr->name << " " << "is deleted from vaccine registration."<<
endl;
                      prev->link = curr->link;
                      delete curr;
                      return;
                    }
```

```
                        else
                         { //front node
                         cout << curr->name << " " << "is deleted from vaccine registration."<<
endl;
                         front = curr->link;
                         delete curr;
                         return;
                        }
                }
        }

        //Display the registration details of specific customer
        void display(string IC) {
          customer *temp = front;

          cout<<"\nRegistration Details: " << endl;

          while (temp != NULL) {
                  if(temp->ic==IC)
                  {
                          cout << "\nName      : "<< temp->name << "\n\nIC Number : " <<
temp->ic << "\n\nAge       : "
                                  << temp->age << "\n\nAddress   : " << temp->address <<
"\n\nContact   : " << temp->tele_num << endl;
                          return;
            }
            else
            {
                temp = temp->link;
                 }
          }
          cout << "Record not found" << endl;
          cout<<endl;
        }

//Display all registration details of customer
        void displayList()
        {
                customer *temp = front;
                customer *rear = NULL;
```

```
                if ((front == NULL) && (rear == NULL))
                {
        cout<<"=====================    LIST OF VACCINE DEPENDENTS
====================="<<endl;

cout<<"---------------------------------------------------------------------------"<<endl << endl;

        cout<<"There are no patient appointments for Vaccination"<<endl;
        return;
                }

                  cout<<"=====================    LIST OF VACCINE DEPENDENTS
====================="<<endl;

cout<<"---------------------------------------------------------------------------"<<endl;
                  cout<< endl << endl;

                        while (temp != NULL)
                        {
                cout << "- " << temp->name << ", " << temp->ic << ", " << temp->age << ", " <<
temp->address << ", " << temp->tele_num << endl;
                temp = temp->link;
                        }
                cout<<endl;
        }
//search for specific patient
        void search(string IC) {
          customer *temp = front;

          cout << endl << endl;
          cout<<"=====================    SEARCH FOR VACCINE DEPENDENTS
====================="<<endl;
          cout<<"---------------------------------------------------------------------------"<<endl;

          while (temp != NULL) {
                  if(temp->ic==IC)
                  {
                        cout << "\nName      : "<< temp->name << "\n\nIC Number : " <<
temp->ic << "\n\nAge       : "
```

```
                                << temp->age << "\n\nAddress   : " << temp->address <<
"\n\nContact   : " << temp->tele_num << endl;
                        return;
            }
            else
            {
                temp = temp->link;
                  }
          }
          cout << "Record not found" << endl;
          cout<<endl;
        }


};
//DECLARATION OF CUSTOMER NODE
class counterNode{
        public:
                int counterNo;
                string vaccinetype;
                int status;
                counterNode *next;
                counterNode *prev;
};

string status(int status){
        if(status==1){
                return "PATIENT INSIDE";
        }
        else if(status==0){
                return "AVAILABLE";
        }
        else{
                return "DEFAULT";
        }
}

//DECLARATION OF COUNTERNODE
class counter{
        private:
```

```
        counterNode *head;

    public:
        counter(){
            head=NULL;
        }

        void insertCounter(int counterNo,int vaccinetype){  //INSERT A NEW
COUNTER
            int currIndex=0;
            counterNode *currNode=head;
            counterNode *prevNode=NULL;
                while(currNode && counterNo > currNode->counterNo){
                    prevNode=currNode;
                    currNode=currNode->next;
                    currIndex++;
                }
            counterNode *newNode=new counterNode;
            newNode->counterNo=counterNo;
            if (vaccinetype == 1){
                newNode->vaccinetype="Pfizer BioNTech";
            }
            else if (vaccinetype == 2){
                newNode->vaccinetype="Sinovac CoronaVac";
            }
            else if (vaccinetype == 3){
                newNode->vaccinetype="Oxford AstraZeneca";
            }
            newNode->status=0;
            if (currIndex==0){
                newNode->next=head;
                head=newNode;
            }
            else{
                newNode->next=prevNode->next;
                prevNode->next=newNode;
            }
        }

        void deleteCounter(int cn){ //DELETE A COUNTER
```

```
                counterNode* prevNode = NULL;
                counterNode* currNode = head;
                int currIndex = 1;
                while (currNode && currNode->counterNo != cn) {
                        prevNode = currNode;
                        currNode = currNode->next;
                        currIndex++;
                }
                if (currNode) {
                        if (prevNode) {
                                prevNode->next = currNode->next;
                                delete currNode;
                        }
                        else {
                                head = currNode->next;
                                delete currNode;
                        }
                        cout<<"YOU SUCCESSFULLY DELETE COUNTER #
:"<<cn<<endl;
        }
        else{
          cout<<"COUNTER #("<<cn<<") DOES NOT EXIST."<<endl;
                }
          }
          bool toCounter(int cn){ //TO ASSIGN CUSTOMER TO THIS COUNTER
                counterNode *newNode=new counterNode;
                newNode=findCounter(cn);
                if(!newNode){
                        cout<<"COUNTER # "<<cn<<" CANNOT BE FOUND\n"<<endl;
                        return false;
                }
                else{
                        if(newNode->status==0){
                                newNode->status=1;
                                cout<<"COUNTER # "<<cn<<" IS
AVAILABLE\n\n"<<endl;
                                return true;
                        }
                        else{
```

```
                                        cout<<"COUNTER # "<<cn<<" IS NOT
AVAILABLE\n\n"<<endl;
                                        return false;
                                }
                        }
                }

                bool updateCounter(int cn){ // to UPDATE THE COUNTER AS AVAILABLE
WHEN NO MORE PATIENT INSIDE
                        counterNode *newNode=new counterNode;
                        newNode=findCounter(cn);
                        if(!newNode){
                                cout<<"COUNTER # "<<cn<<" CANNOT BE FOUND"<<endl;
                                return false;
                        }
                        else{
                                if(newNode->status==1){
                                        newNode->status=0;
                                        cout<<"COUNTER # "<<cn<<" IS SUCCESSFULLY
UPDATED"<<endl;
                                        return true;
                                }
                        }
                }

                counterNode* findCounter(int cn){               //TO CHECK IF COUNTER EXIST
                        counterNode* currNode = head;
                        int currIndex = 1;
                        while (currNode && currNode->counterNo != cn) {
                        currNode = currNode->next;
                        currIndex++;
                        }
                        if (currNode)
                                return currNode;
                        else
                        return 0;
                }

                void displayCounter() // TO DISPLAY COUNTERS
                {
```

```
                counterNode* currNode = head;
                cout<<"====================    COUNTERS IN THIS PPV
====================="<<endl;
                cout<<"-----------------------------------------------------------------"<<endl;
                cout<<"COUNTER #\tVACCINE TYPE\tSTATUS"<<endl;


                while (currNode != NULL){
                        cout <<currNode->counterNo<<"\t"
                                <<currNode->vaccinetype<<"\t\t"
                                <<status(currNode->status)<<"\t"
                                << endl;
                        currNode = currNode->next;
                }
        }
};

int main()
{
        int ch;                 //choice for customer menu
        Queue pq;
        counter c;
        //INSERT PRE-DEFAULT COUNTERS
        c.insertCounter(1001,1);
        c.insertCounter(1002,2);
        c.insertCounter(1003,3);


        Register newUser;
        int choice;

        mainmenu:
         system("CLS");

cout<<"=========================================================\n"
                <<"=    Welcome to our VACCINE QUEUE SIMULATION SYSTEM    =\n"

<<"=========================================================\n";
          cout<<"=             (1) Admin                  ="<<endl;
          cout<<"=             (2) Customer                ="<<endl;
```

```
        cout<<"=              (3) Exit                    ="<<endl;

cout<<"============================================================="<<endl;
        cout<<"Enter your choice :"<<endl;
        cin >> choice;

  if(choice==1)
  {
              admin:
              system("CLS");

cout<<"============================================================\n"
                        <<"=            VACCINE QUEUE SIMULATION SYSTEM
=\n"
                        <<"=                    (ADMIN)                 =\n"

<<"============================================================\n";
                cout<<"=            (1) Create account          ="<<endl;
                cout<<"=            (2) Login                  ="<<endl;
                cout<<"=            (3) Exit                   ="<<endl;

cout<<"============================================================="<<endl;
                cout<<"Enter your choice :"<<endl;
                cin>>ch;

                if(ch==1)
                {
                        system("CLS");
                        //register
                        cout << "CREATE YOUR ACCOUNT" << endl;
                        cout << "-------------------" << endl;
                        cout << "\nEnter your ic number: ";
                        cin >> newUser.ic;
                        cout << "\nEnter your desired passsword: ";
                        cin >> newUser.password;

                        cout << "\n\nSuccessfully Create your Account!. Please Login."
<< endl;
                        cout <<
"=====================================================" << endl << endl;
```

```
                goto loginadmin;
        }
        else if(ch==2)
        {
                system("CLS");
                goto loginadmin;
        }
        else{
                exit(0);
        }

        loginadmin:
        Login newLogin;
        cout << "IC: ";
        cin >> newLogin.inputIC;
        cout << "Password: ";
        cin >> newLogin.inputPass;

        if(newLogin.inputIC == newUser.ic && newLogin.inputPass ==
newUser.password){
                cout << "\nSuccessfully Login!";
        adm:
        system("CLS");

cout<<"============================================================\n"
                <<"=          VACCINE QUEUE SIMULATION SYSTEM
=\n"

<<"============================================================\n";
        cout<<"=          (1) ADD COUNTER                  ="<<endl;
        cout<<"=          (2) DELETE COUNTER               ="<<endl;
        cout<<"=          (3) DISPLAY COUNTER              ="<<endl;
        cout<<"=          (4) CALL CUSTOMER TO COUNTER     ="<<endl;
        cout<<"=          (5) SET PRIORITY FOR ELDERLY     ="<<endl;
        cout<<"=          (6) UPDATE COUNTER(NO PATIENT INSIDE)
="<<endl;
        cout<<"=          (7) EXIT                         ="<<endl;

cout<<"============================================================"<<endl;
```

```
		cout<<"Enter your choice :"<<endl;
		cin>>ch;

		int cn;
		int vt;
		if(ch==1) // OPTION 1
		{
			system("CLS");
			int control = 1;
			do{
				cout<<"\nENTER THE COUNTER # YOU WANT TO
ADD : ";
				cin>>cn;

				cout<<"\nENTER THE VACCINE TYPE FOR THIS
COUNTER (1)PFIZER (2)SINOVAC (3)AZ : ";
				cin>>vt;

				if(!c.findCounter(cn)){
					c.insertCounter(cn,vt);
					cout<<"YOU SUCCESSFULLY ADD COUNTER
# ("<<cn<<") TO THE LIST"<<endl;
				}
				else{
					cout<<"COUNTER # "<<cn<<" EXISTED.
ENTER OTHER COUNTER #."<<endl;
				}
				cout<<"DO YOU WANT TO CONTINUE TO ADD
MORE COUNTER ( 1-YES, 0-NO ) :";
				cin>>control;

			}while(control==1);
		}
		else if(ch==2) // OPTION 2
		{
			system("CLS");
			int control=1;
			do{
				cout<<"\nENTER THE COUNTER # YOU WANT TO
DELETE : ";
```

```
                                        cin>>cn;

                                        if(c.findCounter(cn)){
                                                c.deleteCounter(cn);
                                        }
                                        else{
                                                cout<<"COUNTER CANNOT BE FOUND!
ENTER OTHER COUNTER #."<<endl;
                                        }

                                        cout<<"DO YOU WANT TO CONTINUE TO DELETE
MORE COUNTER( 1-YES, 0-NO ) : ";
                                        cin>>control;
                                }while(control==1);
                }
                else if(ch==3) // OPTION 3
                {
                                system("CLS");
                                c.displayCounter();
                }
                else if(ch==4) // OPTION 4
                {
                                system("CLS");
                                int control = 1;
                                do{
                                        cout<<"\nENTER THE COUNTER # YOU WANT THE
CUSTOMER TO GO : ";
                                        cin>>cn;
                                        bool to=c.toCounter(cn);

                                        if(to){
                                                bool dq = pq.dequeue();
                                                if(dq){
                                                        cout << "Please proceed to Counter #" << cn
<< endl;

                                                }
                                        }
                                        cout<<"DO YOU WANT TO CALL THE NEXT
CUSTOMER( 1-YES, 0-NO ) :";
                                        cin>>control;
```

```
                        }while(control==1);
                }
                 else if(ch==5) // OPTION 5
                {
                        system("CLS");

                        pq.setpriority();
                }
                else if(ch==6) // OPTION 6
                {
                        system("CLS");

                        system("CLS");
                        int control=1;
                        do{
                                cout<<"\nENTER THE COUNTER # YOU WANT TO
UPDATE : ";
                                cin>>cn;

                                if(c.findCounter(cn)){
                                        c.updateCounter(cn);
                                }
                                else{
                                        cout<<"COUNTER CANNOT BE FOUND!
ENTER OTHER COUNTER #."<<endl;
                                }

                                cout<<"DO YOU WANT TO CONTINUE TO UPDATE
MORE COUNTER( 1-YES, 0-NO ) : ";
                                cin>>control;
                        }while(control==1);
                }
                else if(ch==7) // OPTION 7
                {
                        system("CLS");
                        pq.displayList();
        }
        else if(ch==8) // OPTION 8
                {
                        string ic="";
```

```
                        system("CLS");
                        cout << "Please Enter your IC: ";
                        cin >> ic;
                        pq.search(ic);


        }
            else if(ch==9) // OPTION 9
            {
                        system("CLS");
                        cout<<"Thank you for using our system, see you next
time"<<endl;
                        getch();
                        system("CLS");
                        goto mainmenu;
            }
            else // INVALID OPTION
            {
                system("CLS");
                cout<<"Invalid choice"<<endl;
            }

                cout << endl << "Press any key to continue..." ;
             getch();
            goto adm;
                                }
                                else{
                                        cout << "\nLogin failed!";
                                        goto admin;
                                }
  }
  else if(choice==2) //CUSTOMER SIGN IN PAGE
  {
            customer:
            system("CLS");

cout<<"=========================================================\n"
                <<"=       VACCINE QUEUE SIMULATION SYSTEM         =\n"
                <<"=                (Customer)                =\n"
```

```
<<"============================================================\n";
                cout<<"=             (1) Create account             ="<<endl;
                cout<<"=             (2) Login                      ="<<endl;
                cout<<"=             (3) Exit                       ="<<endl;

cout<<"============================================================"<<endl;
                cout<<"Enter your choice :"<<endl;
                cin>>ch;

                if(ch==1) //OPTION 1
          {
                        system("CLS");
                        //register
                        cout << "CREATE YOUR ACCOUNT" << endl;
                        cout << "--------------------" << endl;
                        cout << "\nEnter your ic number: ";
                        cin >> newUser.ic;
                        cout << "\nEnter your desired passsword: ";
                        cin >> newUser.password;

                        cout << "\n\nSuccessfully Create your Account!. Please Login." << endl;
                        cout <<
"============================================================" << endl << endl;

                        goto logincustomer;
                }
                else if(ch==2) // OPTION 2
                {
                                system("CLS");
                                goto logincustomer;
                }
                else{
                                exit(0);
                }

                        logincustomer:
                        Login newLogin;
                        cout << "IC: ";
                        cin >> newLogin.inputIC;
```

```
                        cout << "Password: ";
                        cin >> newLogin.inputPass;

                        if(newLogin.inputIC == newUser.ic && newLogin.inputPass ==
newUser.password){
                                cout << "\nSuccessfully Login!";
                                 goto cust;
                        }
                        else{
                                cout << "\nLogin failed!";
                                goto customer;
                        }
  }
        else if(choice==3)
        {
                system("CLS");
                cout<<"Thank you for using our system, see you next time"<<endl;
                return 0;
        }
        else
        {
                cout<<"Invalid choice"<<endl;
                cout << endl << "Press any key to continue..." ;
                getch();
                goto mainmenu;
        }

  /*Customer Menu*/
  cust:
  system("CLS");
  cout<<"============================================================\n"
          <<"=    Welcome to our VACCINE QUEUE SIMULATION SYSTEM    =\n"
          <<"============================================================\n";
  cout<<"=             (1) Register                =\"<<endl;
  cout<<"=             (2) Cancel                  =\"<<endl;
  cout<<"=             (3) Display                 =\"<<endl;
  cout<<"=             (4) Exit                   =\"<<endl;

cout<<"============================================================\"<<endl;
  cout<<"Enter your choice :"<<endl;
```

```
  cin>>ch;

  if(ch==1)                    //If ch which is customer choice is 1, customer will go to cEnqueue
operation
  {
        system("CLS");
        pq.cEnqueue();
  }
  else if(ch==2)               //Option 2-Cancel required customer enter the ic before go to
remove to cancel the registration
  {
        string ic="";
        system("CLS");
        cout << "Please Enter your IC: ";
        cin >> ic;
        pq.remove(ic);
  }
  else if(ch==3)               //Option 3-Display required customer enter the ic to display the
corresponding details
  {
        string ic="";
        system("CLS");
        cout << "Please Enter your IC: ";
        cin >> ic;
        pq.display(ic);
  }
  else if(ch==4)               //Customer will be redirect to main menu if enter 4-Exit
  {
        system("CLS");
        cout<<"Thank you for using our system, see you next time"<<endl;
        getch();
        system("CLS");
        goto mainmenu;
  }
  else                                 //Invalid choice message will be come out if customer enter
wrong option number
  {
        system("CLS");
        cout<<"Invalid choice"<<endl;
        goto cust;
```

```
   }

        cout << endl << "Press any key to continue..." ;
    getch();
        goto cust;                          //Bring customer to customer menu


return 0;
}
```