UTM
UNIVERSITI TEKNOLOGI MALAYSIA

FACULTY OF COMPUTING

UNIVERSITI TEKNOLOGI MALAYSIA

DATA STRUCTURE & ALGORITHM (SECJ2013)

SEMESTER 1 2022/2023

Mini Project Documentation
UTM PKU Appointment System

By
Wan Amirul Hafiq Bin Wan Huzaini (020509150035) - Group Leader
Ikmal Bin Khairulezuan (020326040175)
Muhammad Iqmal Bin Sis (021118131085)
Muhammad Amir Jamil Bin Jamlus (021107050239)

SECTION 02

Lecturer:
Dr. Pang Yee Yong

Date:
30/1/2023

For lecturer use:

Description	Mark Distribution	Mark
iv. Project Report ● System analysis ● Design ● Program code	10 15 25	
v. Presentation & Demo	25	
vi. System Prototype	25	
TOTAL	**100**	

Table Of Content

PART 1: INTRODUCTION

1.1 Synopsis Project:

For our Universiti Teknologi Malaysia (UTM) Pusat Kesihatan UTM (PKU) appointment system mini project, the system is made to ease the staff of PKU to track patients that make appointments to meet the doctor and limit the queue of patient to make sure there is no overloaded patient come to the PKU at the same time. Our system can insert a new patient, display a list of patients, delete selected patients, search patients and exit the system. In the system we implement a queue linked list for inserting new patients, displaying the list of patients and deleting selected patients while for searching a patient we use a linear search method of data structure in our coding.

1.2 Objective of the project

- Standardization of patients registration process
- Uniform data management
- Increase the efficiency of data entry, query and deletion, in terms of time
- Produce an easy to understand interface
- Ease of navigation to improve user experience
- Customization of size of patients intake

PART 2: SYSTEM ANALYSIS AND DESIGN (USE CASE, FLOWCHART AND CLASS DIAGRAM)

2.1 System Requirements

Use case diagram

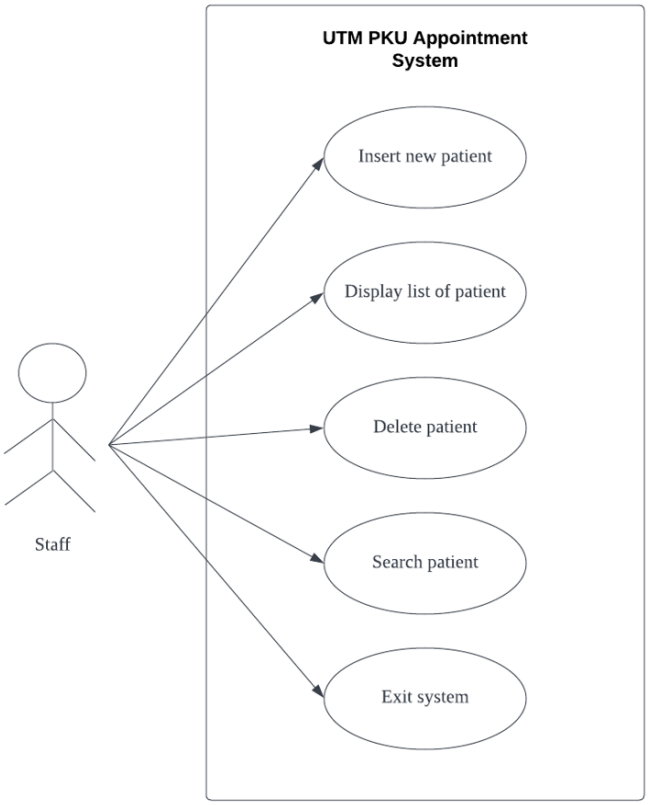


Figure 1: Use Case Diagram for UTM PKU Appointment System

Use Case Description for UTM PKU Appointment System

The system user is a PKU staff

Actor	**Task**
Staff	Insert new patient, Display list of patient, Delete patient, Search patient and Exit system.

Detail Description for Each Use Cases

The system has 5 main use cases.

Use Case	**Purpose**
Insert new patient	Insert new patient in the queue list. If the list is full, staff cannot enter new patient in the queue list.
Display list of patient	Provide choices for the user to perform certain operations in the system and display the list of all patients in the waiting list.
Delete patient	Delete patient from queue if the queue is not

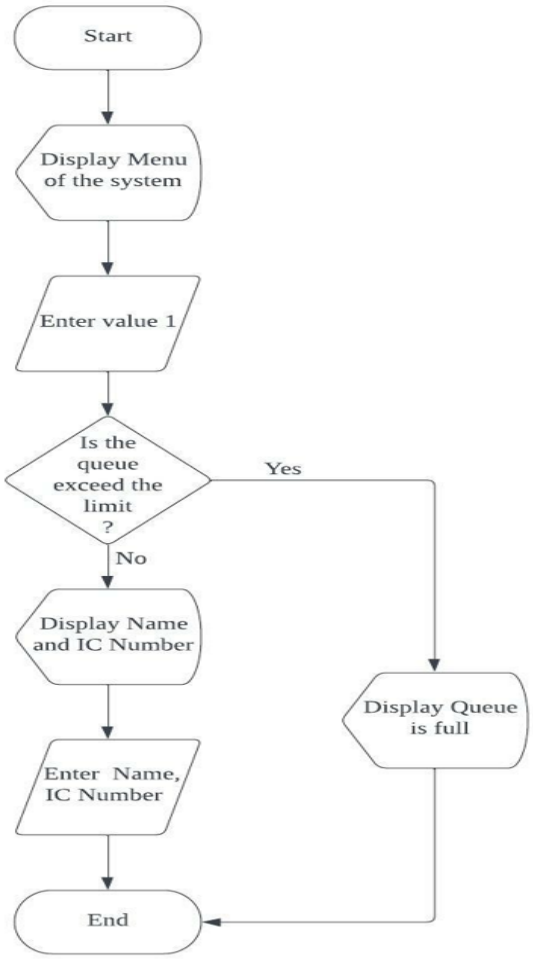
	empty
Search patient	Search for any existing patient inside the queue
Exit System	Exit the system

2.2 System Design

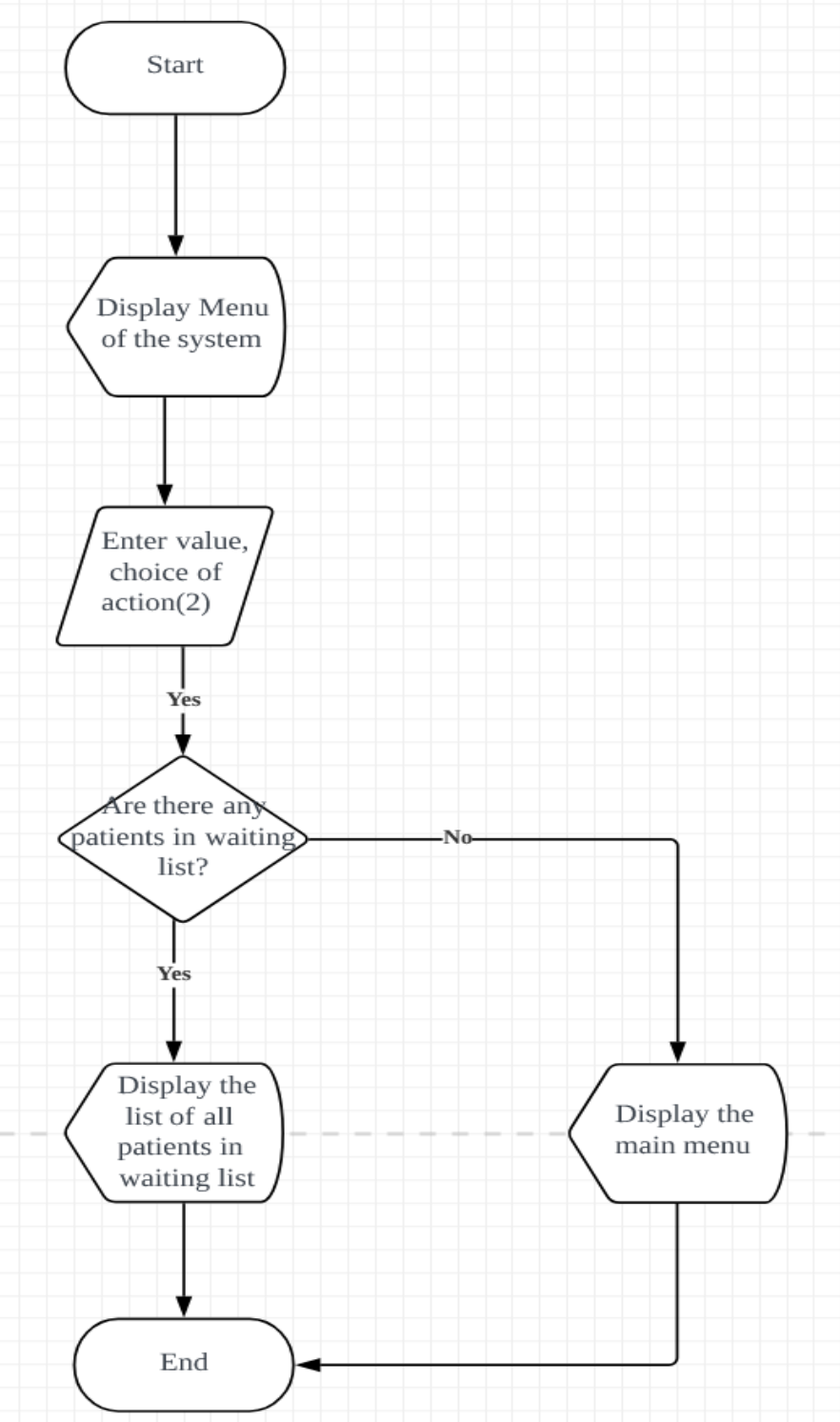
Algorithm: Flowchart for each module.

FlowChart 1: Insert new patient

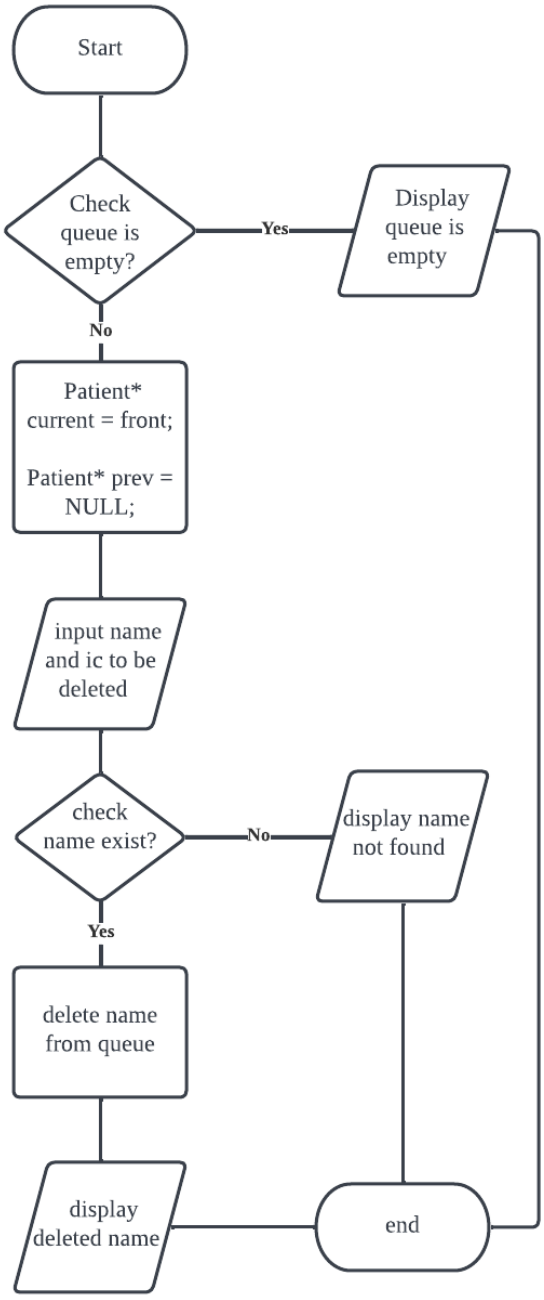
Prepared By: Ikmal Bin Khairulezuan



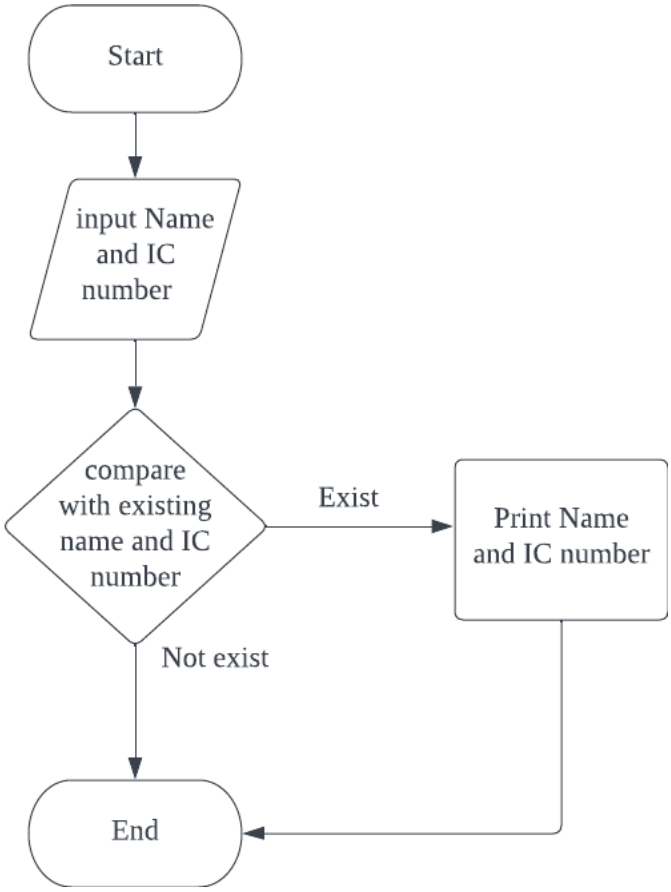
FlowChart 2: Display list of patient
Prepared By: MUHAMMAD AMIR JAMIL BIN JAMLUS



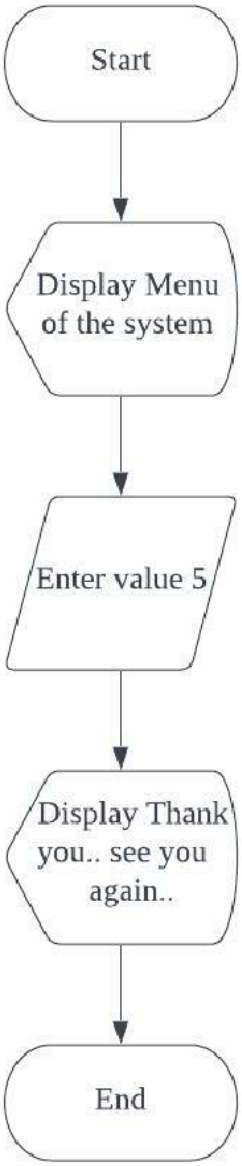
FlowChart 3: Delete Patient
Prepared By: Wan Amirul Hafiq bin Wan Huzaini



FlowChart 4: Search Patient
Prepared By: Muhammad Iqmal bin Sis

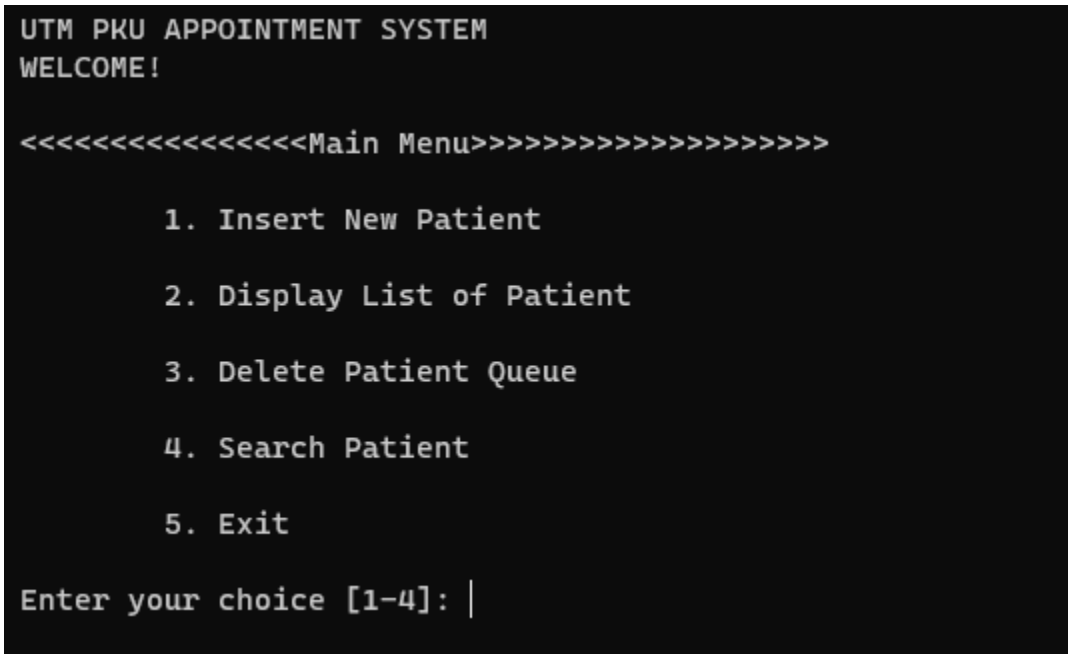


FlowChart 5: Exit System
Prepared By: Provide group members who prepare the algorithm.



PART 3: SYSTEM PROTOTYPE

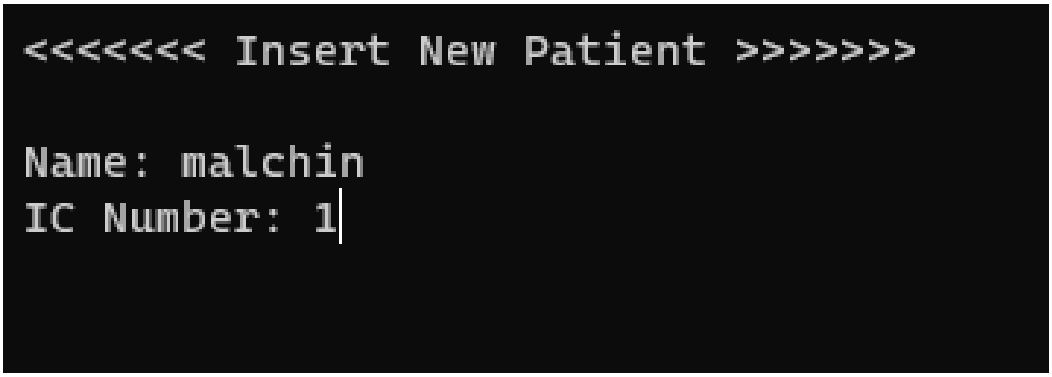
Main Menu



Screen 1: Main Menu

Screen 1: The user must insert an integer value in the range 1-5. If the user enter other number, the system will prompt an error message and user need to restart the display.
Prepared By:Ikmal Bin Khairulezuan.

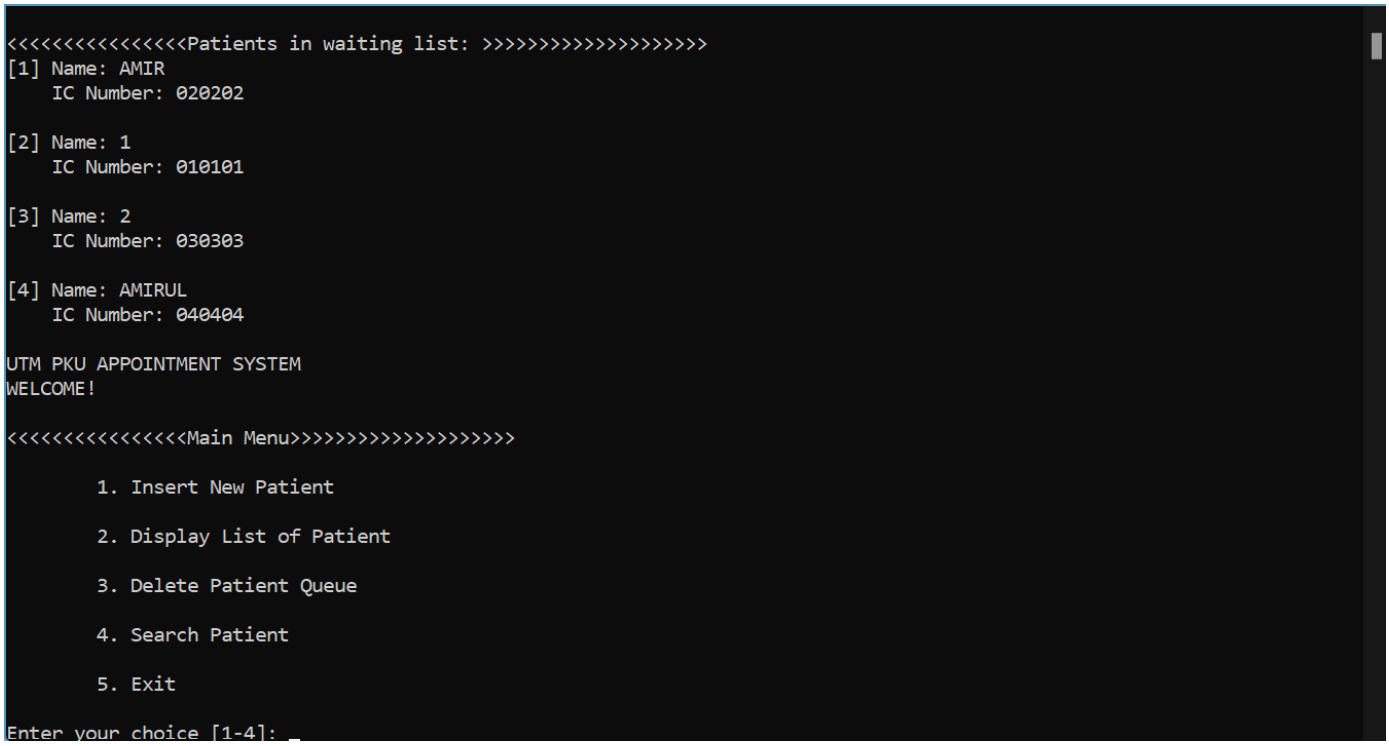
Insert New Patient



Screen 2: Insert new patient

Screen 2: The user enters number '1' to go to the insert new patient display. The user must enter the name and IC number that the user wants to add in the appointment queue. If the queue is full, it will display "Queue is full!" and user unable to insert a new patient in the appointment queue.
Prepared By: Ikmal Bin Khairulezuan

Display Patient waiting list

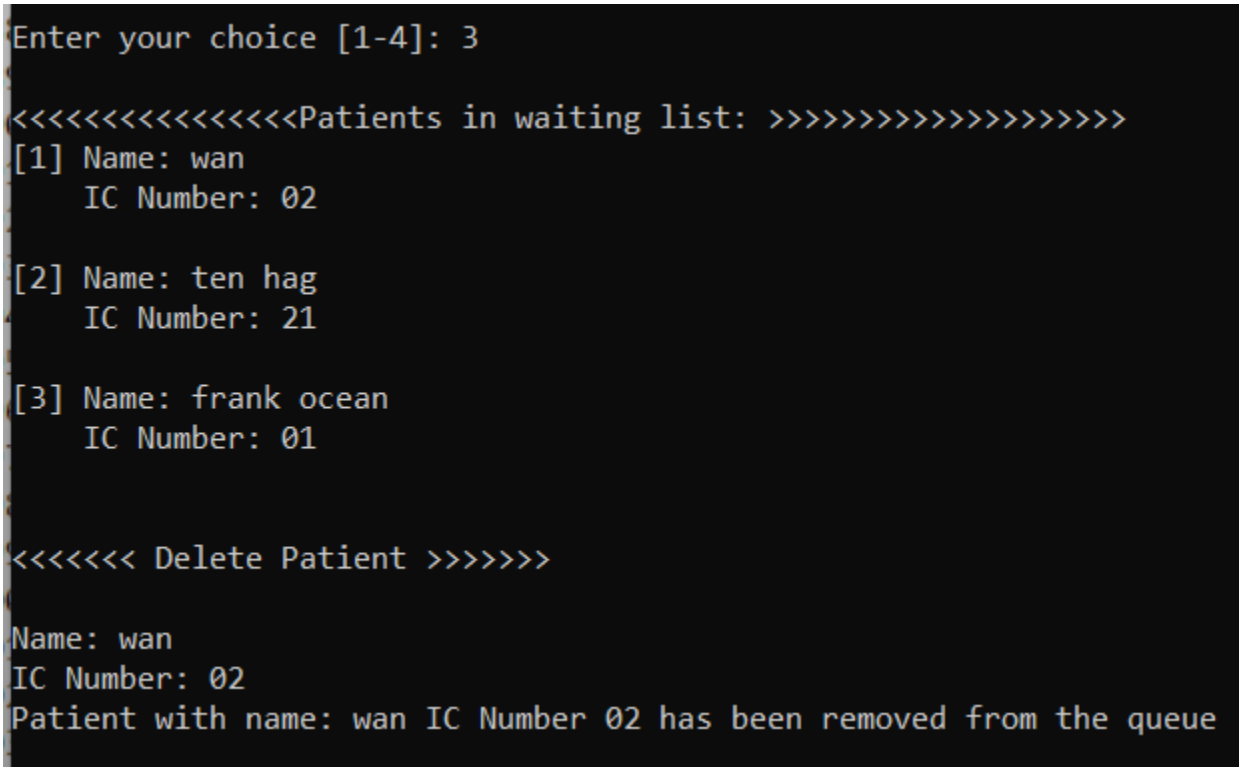


Screen 3: Main Menu

Screen 3: This screen assists in the process of displaying the queue. First, when the user enters value ‘2’ they will be directed to the display feature of our system. Here our system produces a screen where it displays all the users that are currently in the waiting queue, along with their information entered into the system like Name and IC Number. We can see from the screen above, the lists are displayed in order, where number [1] is the first person to get in the queue and into the list. This way, it mimics the real way of queueing.

Prepared By: Muhammad Amir Jamil Bin Jamlus

Delete Patient Queue

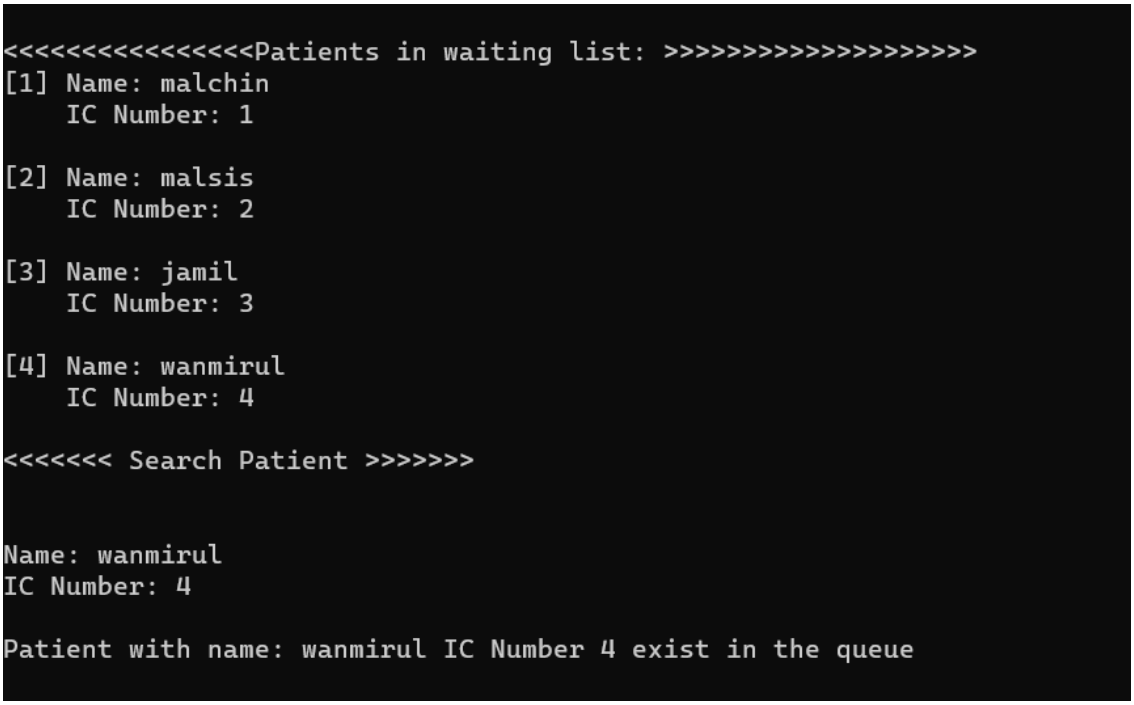


Screen 4: Delete Patient Queue

Screen 4: The user enters number ‘3’ to go to the Delete Patient Queue display. The user must enter the name and IC number that the user wants to delete in the appointment queue. If the queue is emptyl, it will display “Queue is empty, no patient to remove!” and will redirect the user to main menu. If the name that the user entered is not in the queue the system will display “ Patient is not found” and will redirect the user to main menu.

Prepared By: Wan Amirul Hafiq Bin Wan Huzaini

Search Patient

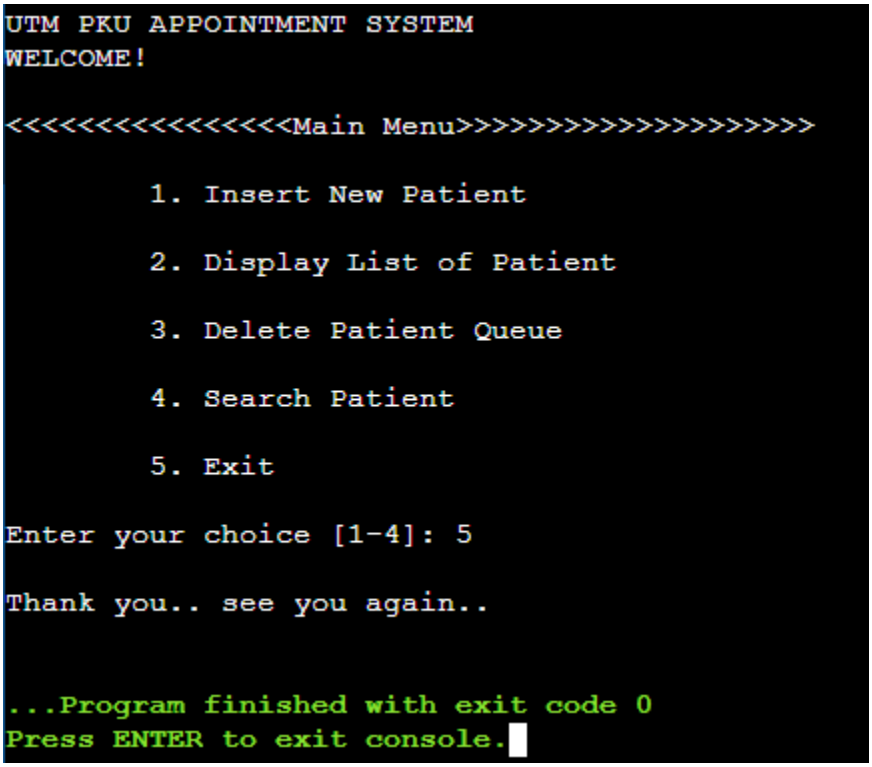


Screen 5: Shows Search Patient Display

Screen 5: The user must enter number '4' to go to the search patient display. The user must enter a specific name and IC number that the user wants to find. If the name and the IC number is the same as the system, it will display the name and IC of it. But, if the searched user is not found, it will display an error where it says the user is not found.

Prepared By: Muhammad Iqmal bin Sis

Exit System



Screen 5: Exit System Display

Screen 5: If the user want to exit the system, user need to enter number 5 and it will show "Thank you.. See you again.." and press ENTER to exit from the system.
Prepared By: Ikmal Bin Khairulezuan

PART 4: DEVELOPMENT ACTIVITIES

Shows every meeting conducted, meeting activity, task being assigned and whether task achieved or not.

Meeting Date	Members Participate in the meeting	Activity	Task for each member	Task Achieved (Yes/No)
20/1/2023	● Wan Amirul Hafiq Bin Wan Huzaini ● Ikmal Bin Khairulezuan ● Muhammad Iqmal Bin Sis ● Muhammad Amir Jamil Bin Jamlus	● Discuss how the system we choose work. ● Distribute the task for each group member.	● Wan Amirul Hafiq Bin Wan Huzaini - Delete patient ● Ikmal Bin Khairulezuan - Insert patient, Display main menu and Exit system ● Muhammad Iqmal Bin Sis - Search Patient ● Muhammad Amir Jamil Bin Jamlus - Display list of patient	Yes Yes Yes Yes
26/1/2023	● Wan Amirul Hafiq Bin Wan Huzaini ● Ikmal Bin Khairulezuan ● Muhammad Iqmal Bin Sis ● Muhammad Amir Jamil Bin Jamlus	● Make a report about the mini project	● Wan Amirul Hafiq Bin Wan Huzaini - flowchart ● Ikmal Bin Khairulezuan - Introduction and Objective ● Muhammad Iqmal Bin Sis - System prototype ● Muhammad Amir Jamil Bin Jamlus - Description of use case	Yes Yes Yes Yes

PART 5: APPENDIX

```
#include <iostream>
#include <iomanip>
#include <stdlib.h>
using namespace std;

class Patient{
public:
   string name;
   string IC;
   Patient* next;
   Patient(){
      name = "";
      IC = "";
   }
   Patient(string n, string ic){
      name=n;
      IC = ic;
   }
};
class Queue{
public:
   Patient* front ;
   int max_queue=5;//set maximum queue
   Patient* back;
   Queue()//constructor
         {
      front = NULL;
      back = NULL;
   }
   ~Queue()//destructor
         {
      Patient* temp = front;
      while(temp){
         front =  temp->next;
         delete temp;
         temp = front;
      }
   }
   bool isEmpty(){
      if(front== NULL && back== NULL)

         return true;
      else
         return false;
   }


   void enQueue(){
      string newName, newIC;
```

```
        cout<<endl;
    if(max_queue==0)//Check condition if queue is full
                {
                cout<<"Queue is full!"<<endl<<endl;
                return;}
                system("CLS");
    cout<<"<<<<<<< Insert New Patient >>>>>>>"<<endl<<endl;
    //insert at back
    cout<<"Name: ";
    getline(cin, newName);
    cout<<"IC Number: ";
    getline(cin, newIC);


    Patient* newPatient = new Patient(newName, newIC);
    if(isEmpty())//If queue is empty front and back will point to newPatient
                {
       newPatient->next=NULL;
       front = newPatient;
       back = newPatient;
    }else//If queue is not empty back will point to newPatient
                {
       newPatient->next=NULL;
       back->next = newPatient;
       back = newPatient;
    }
    max_queue--;//Decrese the queue size
    cout<<endl;
  }

       void displayQueue(){
                system("CLS");
                cout<<endl;
          if(isEmpty())//if queue is not empty
                {
             cout<<"Queue is empty"<<endl<<endl;
          }
          else{
             Patient* ptr=front;
             int num=1;
             cout<<"<<<<<<<<<<<<<<<<<Patients in waiting list:
>>>>>>>>>>>>>>>>>>>>>"<<endl;
             while(ptr!=NULL)
                        {
                cout<<"["<<num<<"] "<<"Name: "<<ptr->name<<endl;
                                cout<<"    IC Number: "<<ptr->IC<<endl<<endl;;
                ptr = ptr->next;
                num++;
             }
          }
```

```
	}

	void deQueue(){
		displayQueue();
	   if(isEmpty())//Check condition if queue is empty
		{
		cout<<endl;
	      cout<<"Queue is empty, no patient to remove"<<endl<<endl;
	   }
	   else
		{
	      Patient* current = front;
	      Patient* prev = NULL;

	      string oldName, oldIC;
	      cout<<endl;
	      cout<<"<<<<<<< Delete Patient >>>>>>>"<<endl<<endl;
	      cout<<"Name: ";
	      getline(cin, oldName);
	      cout<<"IC Number: ";
	      getline(cin, oldIC);

	      while(current != NULL){
	         if(current->name == oldName){
	            if(prev == NULL){
	               // this is the first element in the queue
	               front = current->next;
	            }
	            else{
	               prev->next = current->next;
	            }
	            delete current;
	            max_queue++;//increase the queue size
	            cout<<"Patient with name: "<<oldName<<" IC Number "<<oldIC<<" has been
removed from the queue"<<endl<<endl;
	            return;
	         }
	         prev = current;
	         current = current->next;
	      }
	      cout<<"Patient with name: "<<oldName<<" IC Number "<<oldIC<<" not found in the
queue"<<endl<<endl;
	   }
	}

	void searchPatient(){


	      if(isEmpty()){
	      cout<<endl;
	      cout<<"Queue is empty, can not proceed with searching!"<<endl<<endl;
```

```
		}else{
			displayQueue();
			string oldName, oldIC;
			cout<<"<<<<<<< Search Patient >>>>>>>"<<endl<<endl;
		cout<<endl;
		cout<<"Name: ";
		getline(cin, oldName);
		cout<<"IC Number: ";
		getline(cin, oldIC);
		Patient *curr = front;

		while(curr!=NULL){

			if(curr->name == oldName && curr->IC == oldIC){
				cout<<endl;
				cout<<"Patient with name: "<<oldName<<" IC Number "<<oldIC<<"
exist in the queue"<<endl<<endl;
				return;
					}

					curr = curr->next;
			}
			cout<<"Patient with name: "<<oldName<<" IC Number "<<oldIC<<" does not
exist in the queue"<<endl<<endl;

		}
	}


};

void dispMenu(){

	cout<<"UTM PKU APPOINTMENT SYSTEM"<<endl;

	cout<<"WELCOME!"<<endl<<endl;

	cout<<"<<<<<<<<<<<<<<<<<Main Menu>>>>>>>>>>>>>>>>>>>>"<<endl;

	cout<<"\n\t1. Insert New Patient"<<endl;

	cout<<"\n\t2. Display List of Patient"<<endl;

	cout<<"\n\t3. Delete Patient Queue"<<endl;

	cout<<"\n\t4. Search Patient "<<endl;

	cout<<"\n\t5. Exit"<<endl;
}
```

```
int main(){

    Queue patient;

        int choice;
        do
        {

                dispMenu();

                cout<<"\nEnter your choice [1-4]: ";

                cin>>choice;

                cin.ignore();

                switch(choice){
                //Insert Patient
                case 1: patient.enQueue();
                        break;
                //Display patient
                case 2: patient.displayQueue();
                        break;
                //Delete patient
                case 3: patient.deQueue();
                        break;
                //Search patient
                case 4: patient.searchPatient();
                        break;
                //End system
                case 5:
                cout<<"\nThank you.. see you again.."<<endl;
                break;
                //If user enter other than 1-5
                default: cout<<"\nWrong input. Please try again"<<endl;
                }
        }
    while((choice>0)&&(choice<5));
        return 0;
}
```