Project Report

Nadra Ticket System

CAPITAL UNIVERSITY OF SCIENCE & TECHNOLOGY

B.S.COMPUTER SCIENCE, 3rd Semester

Section 3

DATA STRUCTURES

(CS-2143)

Course Instructor: Shahzad Qaiser

Submitted by:

Muhammed Haseeb Khan (Team Leader)

Reg # BCS193049

Faaiz Mehmood

Reg # BCS193007

Mustafa Hafeez

Reg # BCS193026

Project Title

Nadra Ticket System

Abstract

The Project is fundamentally of an idea of first in first out methods one start things out and one he leaves the framework however its information and its own data ought to be remained careful and secure from another third individual. For the implementation of this venture we are utilizing programming Microsoft visual studio 2019 coding programming. Record taking care of for putting away individual data is utilized with the concept of File Handling. The main component of this undertaking is that the framework will naturally eliminate or erase the individual symbolic number when its task is finished and welcome the other individual and show various decisions for a user.

Problem Statement:

The problem before implementing this project is that in Nadra as you see there is no concept about who come first and who will come before or after him/her and people also come into misunderstanding. Due to this problem and employees also got so many confusions during this scenarion so to overcome this problem we have implemented a project based on this scenario which will help the employee's as well as customer's which have many more concepts and features and will help to improve the performance of the system.

List of Requirements:

The following are the recquirements of the project which should be used during implementing of this project:

1. Microsoft Visual Software C++ / DEV C++.
2. Concept of Link list.
3. Concept link list based Queue.
4. Designing functions.
5. File Handling.

Data Structures (With Justification):

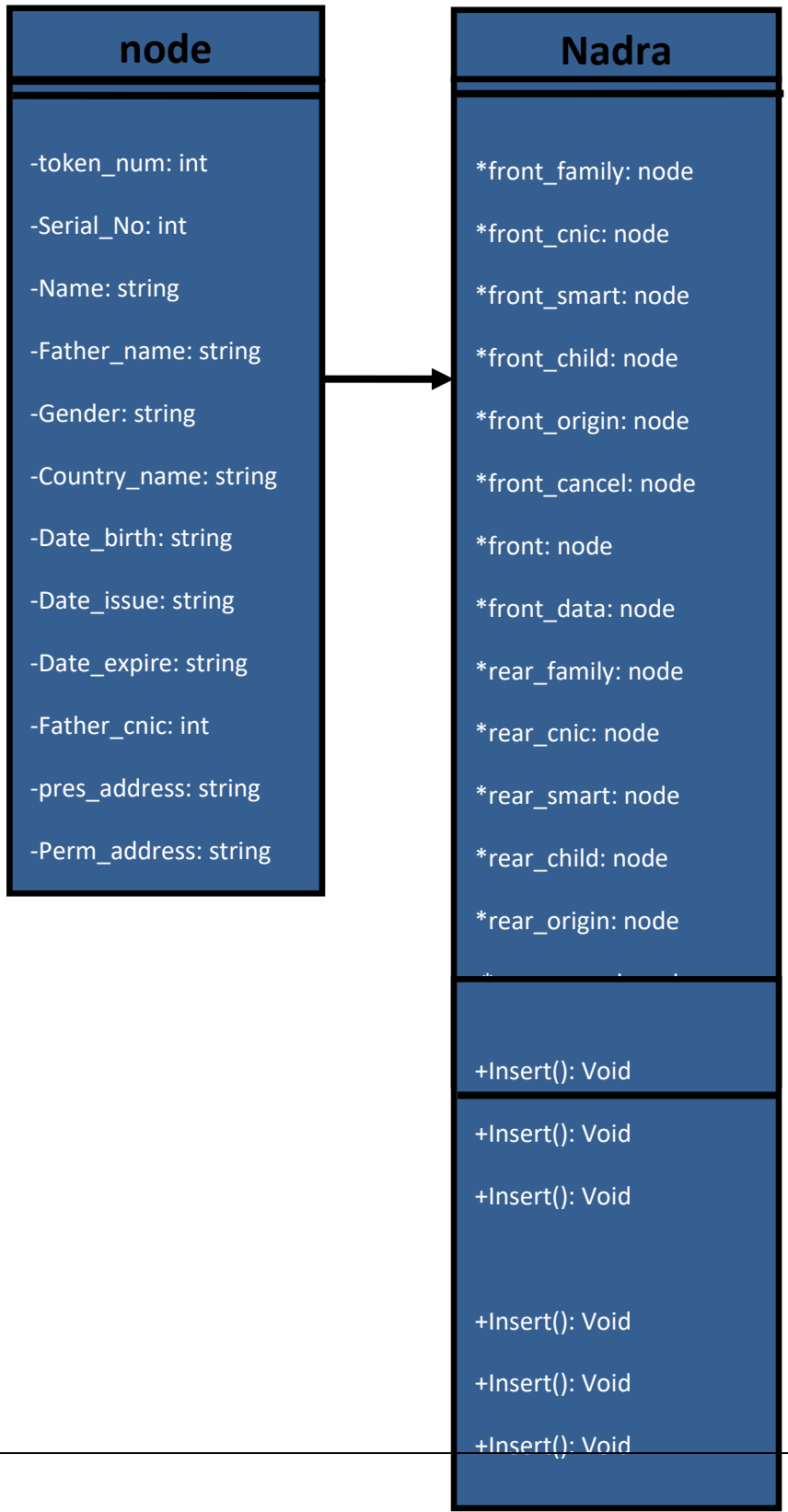
The following are the data structure's are used during implementing of this project Microsoft Visual Software C++ / DEV C++ for Coding purpose we need a software some concept of Link list so that we can easily delete or insert a customer information, concept of link list based Queue implementation so that it will work on the principal fifo (first come first out) that will not create any misunderstanding, some designing functions for a beautifull display

File Handling:

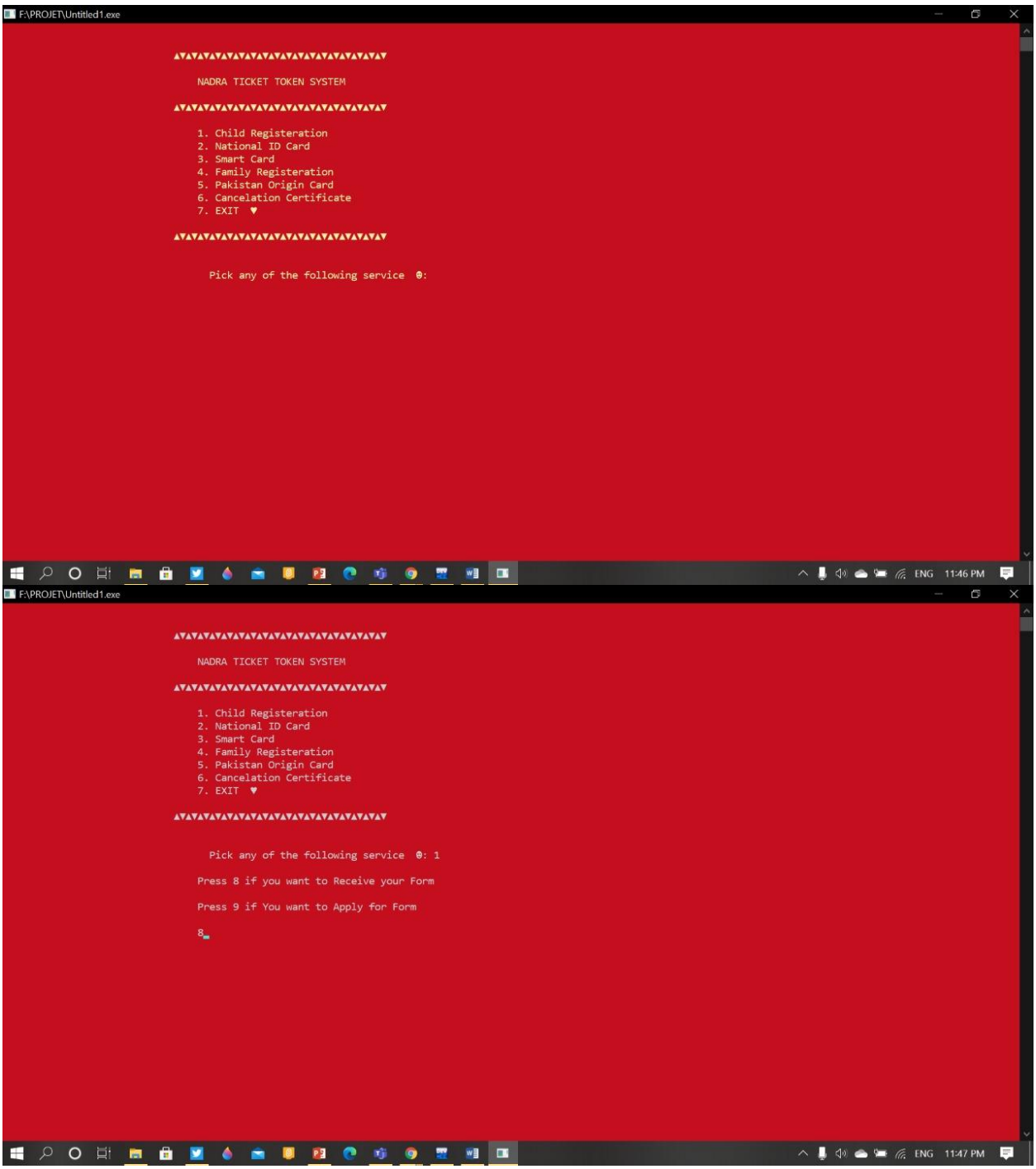
The file handling is used in this project in order to store the information of the customer and all the data that is inserted in the software once will save auotmatically in the file. The fileadling is also used in case if any problem occur regarding customers data when the record is entered so that we can compare it with the previous data in our file or all data that is inserted during entering of data. This

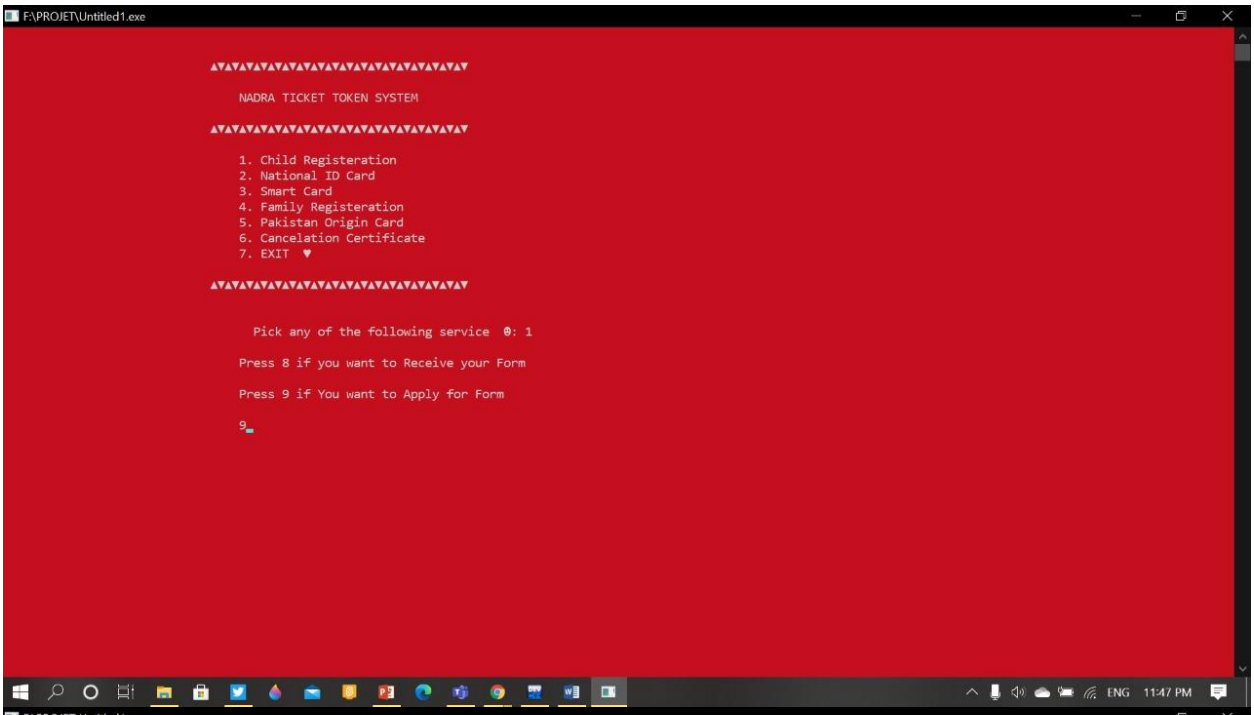
will also consume less time as well the efficiency of the system and increase the betterment of the organization.

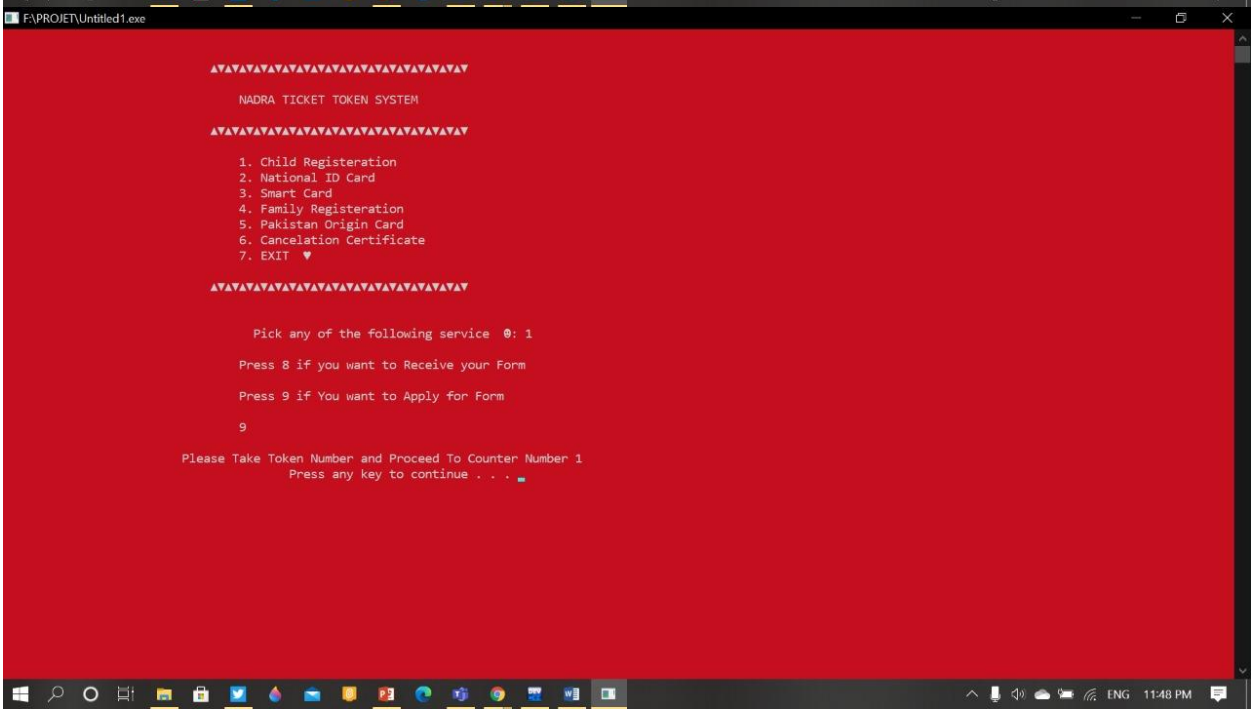
UML (Basic Class Diagram

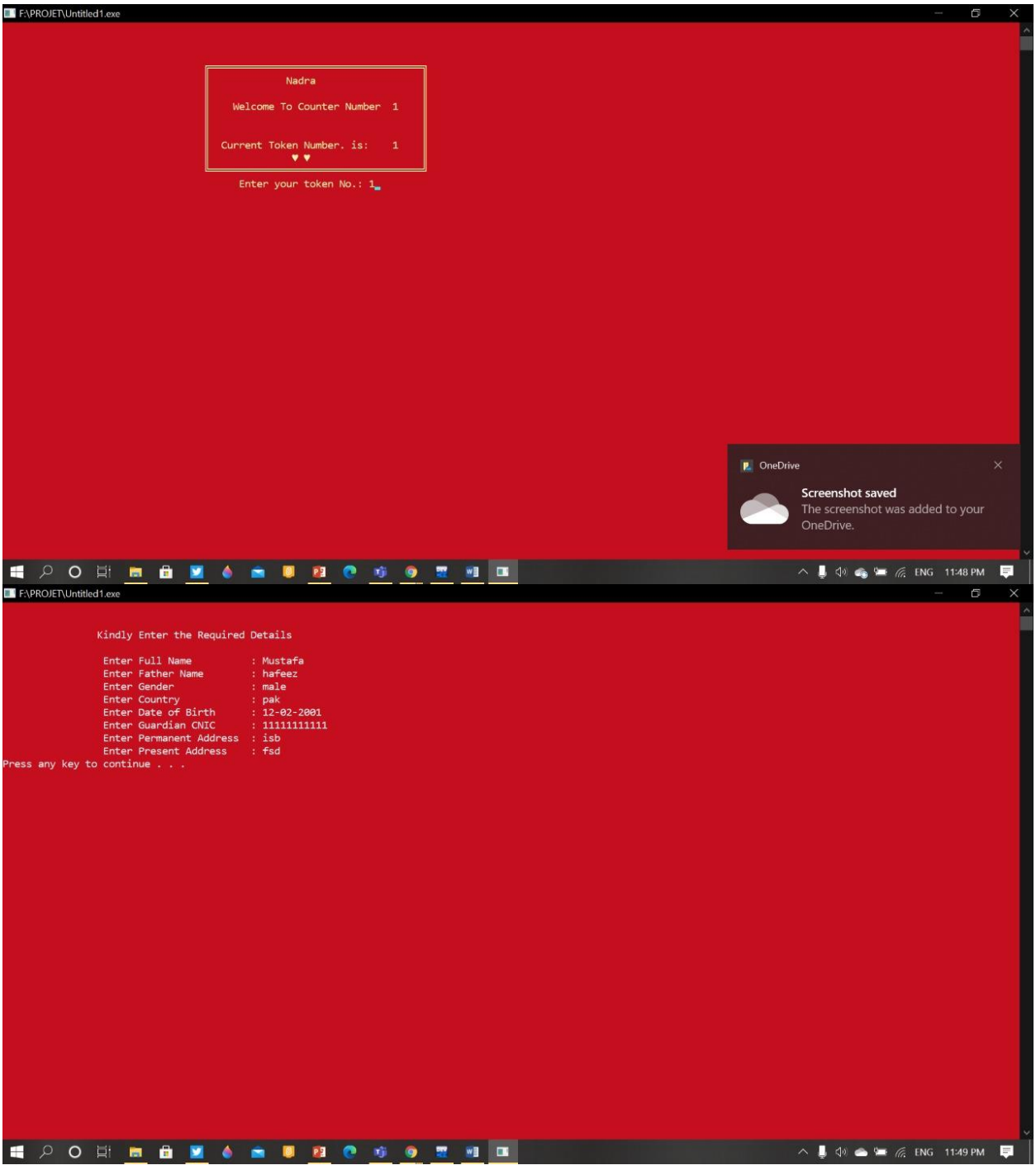


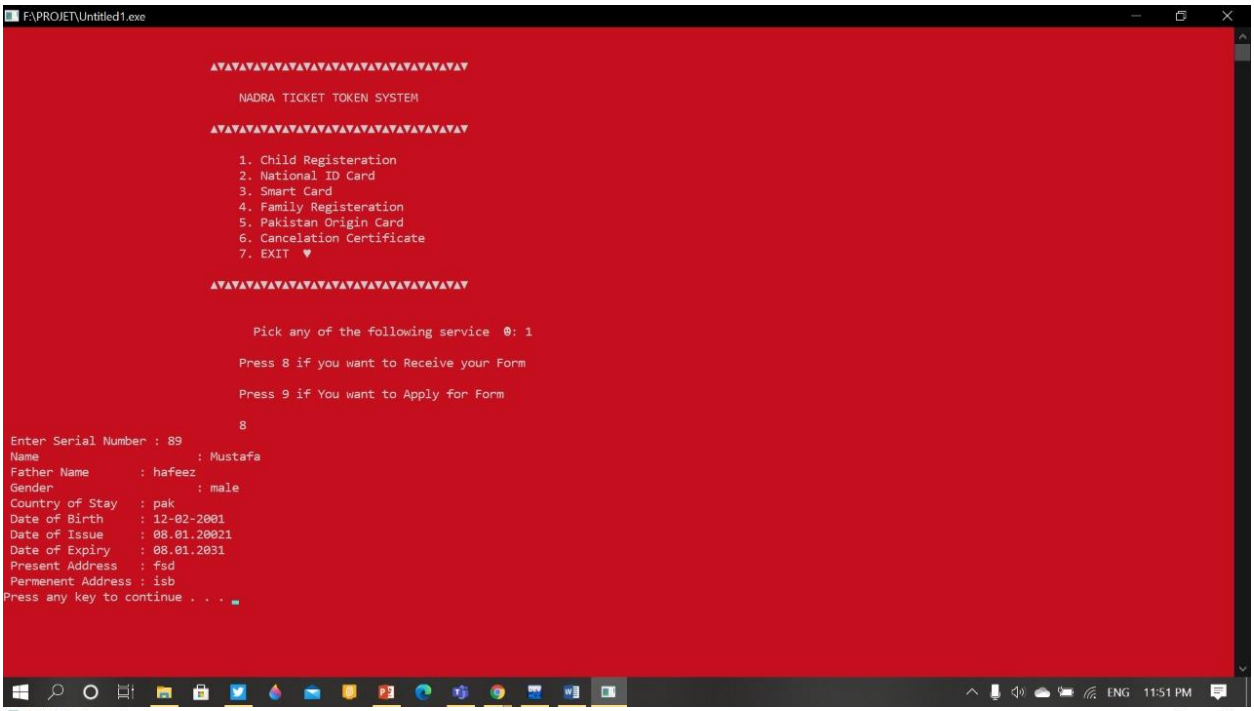
Project Outputs:

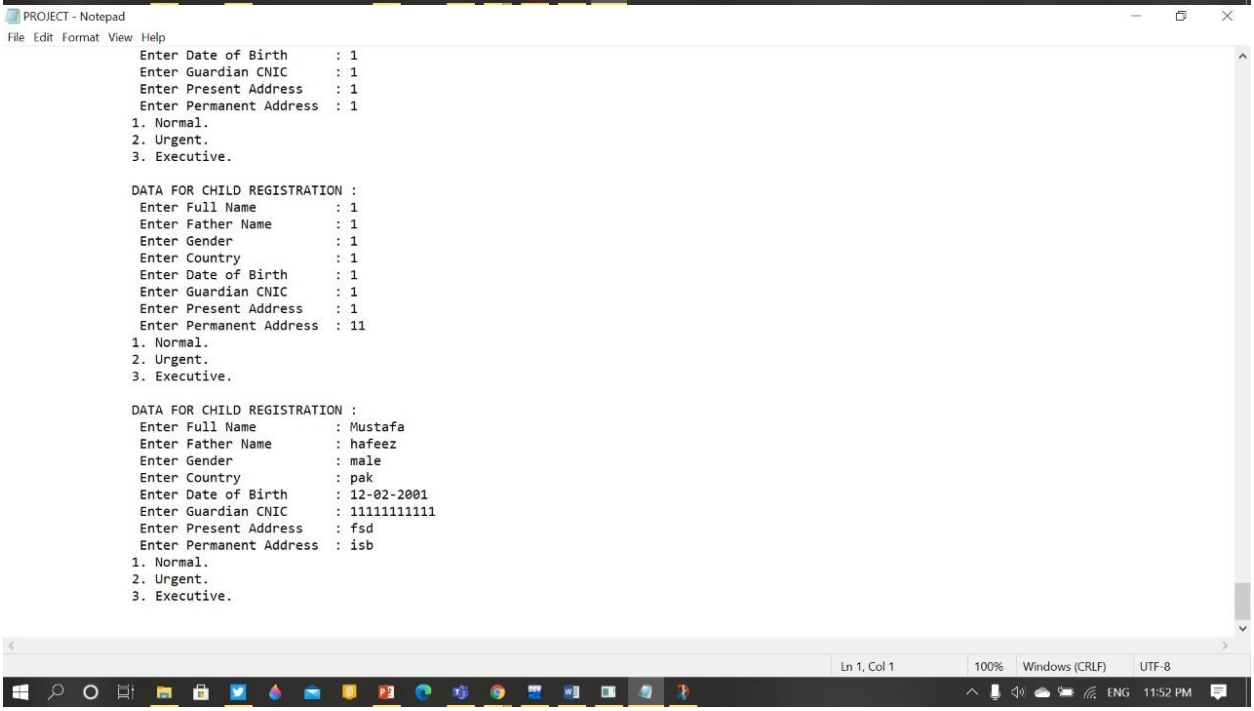



F:\PROJET\Untitled1.exe
NADRA TICKET TOKEN SYSTEM
1. Child Registeration
2. National ID Card
3. Smart Card
4. Family Registeration
5. Pakistan Origin Card
6. Cancelation Certificate
7. EXIT
Pick any of the following service 0: 1
Press 8 if you want to Receive your Form
Press 9 if You want to Apply for Form
9
ENG 11:47 PM


F:\PROJET\Untitled1.exe
NADRA TICKET TOKEN SYSTEM
1. Child Registeration
2. National ID Card
3. Smart Card
4. Family Registeration
5. Pakistan Origin Card
6. Cancelation Certificate
7. EXIT
Pick any of the following service 0: 1
Press 8 if you want to Receive your Form
Press 9 if You want to Apply for Form
9
Please Take Token Number and Proceed To Counter Number 1
Press any key to continue . . .
ENG 11:48 PM


F:\PROJET\Untitled1.exe
Nadra
Welcome To Counter Number 1
Current Token Number. is: 1
♥ ♥
Enter your token No.: 1
OneDrive
Screenshot saved
The screenshot was added to your OneDrive.
ENG 11:48 PM
F:\PROJET\Untitled1.exe
Kindly Enter the Required Details
Enter Full Name : Mustafa
Enter Father Name : hafeez
Enter Gender : male
Enter Country : pak
Enter Date of Birth : 12-02-2001
Enter Guardian CNIC : 11111111111
Enter Permanent Address : isb
Enter Present Address : fsd
Press any key to continue . . .
ENG 11:49 PM





Complete Code:

```
#include<iostream>

#include<string>

#include <fstream>
```

```
#include <ctime>
#include <stdlib.h>
using namespace std;
char heart =286,h1=287;

struct node
{
	int token_num,serial_no;
	string name,father_name,gender,country_name,date_birth,date_issue,date_expire,father_cnic,pres_address,perm_address;
	node *next;
};
class Nadra
{
private:
	node *front_family, *front_cnic, *front_smart, *front_child, *front_origin, *front_cancel, *front,*front_data;
	node *rear_family, *rear_cnic, *rear_smart, *rear_child, *rear_origin, *rear_cancel, *rear, *rear_data;
public:

	Nadra()
	{
```

```
        front_family=front_cnic=front_smart=front_child=front_origin=front_cance
l=front=front_data=NULL;

        rear_family=rear_cnic=rear_smart=rear_child=rear_origin=rear_cancel=rear
,rear_data=NULL;
        }

        // Designing Functions
        void clear();
        void designverti();
        void packages(string);

        //Program Functions
        void family_counter();
        void cnic_counter();
        void smart_counter();
        void child_counter();
        void origin_counter();
        void cancel_counter();

        //Receipt Function
        void normal_receipt(string);
        void urgent_receipt(string);
```

```
        void executive_receipt(string);


        void search_data();
        void display_data(string,int);
        void enqueue(int);
        void search(node*,int);
        void insert_data();
        void dequeue();
        void menu();
        void choices();
        void display_rear(node*,int);
        void dequeue_counter(node*);
};

void Nadra::search_data()
{
        int num;
        cout<< " Enter Serial Number : ";
        cin>>num;
        node *temp=front_data;
        if(front_data==NULL)
                cout<< " No Data To Show "<<endl;
        else
```

```
{
        while(temp!=NULL)
        {
                if(temp->serial_no==num)
                        break;
                else
                        temp=temp->next;
        }
        if(temp!=NULL)
        {
                cout<< " Name                   : "<<temp->name<<endl;
                cout<< " Father Name        : "<<temp->father_name<<endl;
                cout<< " Gender                 : "<<temp->gender<<endl;
                cout<< " Country of Stay   : "<<temp->country_name<<endl;
                cout<< " Date of Birth        : "<<temp->date_birth<<endl;
                cout<< " Date of Issue        : "<<"08.01.20021"<<endl;
                cout<< " Date of Expiry     : "<<"08.01.2031"<<endl;
                cout<< " Present Address   : "<<temp->pres_address<<endl;
                cout<< " Permenent Address : "<<temp->perm_address<<endl;
        }
        else
                cout<< " No record Found \n Your Cars is not Found "<<endl;
}
```

```
}


// Receipt Functions
void Nadra::display_data(string str,int price)
{
        node *temp=front_data;
        if(front_data==NULL)
                cout<< " Empty Record "<<endl;
        else
        {
                cout <<endl<<endl<<endl;
                char rb=187;
                char midbar = 186;
                char bar = 205;
                char downleftconr=200;
                char doright =188;
                char rightvetbar=187;
                char leftvetbar = 201;
                        cout<<"\t\t\t\t  ";
                cout <<leftvetbar;
                for (int i=0;i,i<54;i++)
                {
```

```
                system ("color 4e");
                cout <<bar;
        }



    cout<<"\n\t\t\t\t  "<<midbar<<"      Seril No. :            "<<temp->serial_no;
    cout<<"\n\t\t\t\t  "<<midbar<<"                          "<<"          ";
    cout<<"\n\t\t\t\t  "<<midbar<<"      Name :                "<<temp->name;
    cout<<"\n\t\t\t\t  "<<midbar<<"      Application Type :    "<<str<<" ";
    cout<<"\n\t\t\t\t  "<<midbar<<"      Application Fee  :    "<<price;
    cout<<"\n\t\t\t\t  "<<midbar<<"      Service Fee      :    "<<"0.00";
    ;
    cout<<"\n\t\t\t\t  "<<midbar<<"      Total Fee        :    "<<price;

    cout<<"\n\t\t\t\t  "<<midbar<<"           \3 \3 \3 \3 \3          "<<"
"<<endl;

        cout<<"\t\t\t\t  ";
        cout <<downleftconr;
        for (int i=0;i,i<54;i++)
        {
                system ("color 4e");
```

```
                cout <<bar;
            }
            cout<<"\n\t\t\t\t  ";
            temp=temp->next;

        }
}

// Inserting Data of a Person
void Nadra::insert_data()
{
        ofstream out ("PROJECT.txt",ios::app | ios::out| ios::in);
        if (out.is_open())
{
        system ("pause");
        system("cls");

        system("color 4f");
        cout <<endl<<endl;
        cout<< " \t\tKindly Enter the Required Details "<<endl<<endl;

        if(front_data==NULL)
        {
```

```
        node *lat=new node;
        srand((unsigned)time(0));
        for(int i=0;i<1;i++)
                lat->serial_no=(rand()%100)+1;

cout<< " \t\t Enter Full Name         : ";
cin>>lat->name;
cout<< " \t\t Enter Father Name       : ";
cin>>lat->father_name;
cout<< " \t\t Enter Gender           : ";
cin>>lat->gender;
cout<< " \t\t Enter Country          : ";
cin>>lat->country_name;
cout<< " \t\t Enter Date of Birth     : ";
cin>>lat->date_birth;
cout<< " \t\t Enter Guardian CNIC     : ";
cin>>lat->father_cnic;
cout<< " \t\t Enter Permanent Address  : ";
cin>>lat->perm_address;
cout<< " \t\t Enter Present Address    : ";
cin>>lat->pres_address;
front_data=lat;
rear_data=lat;
```

```
lat->next=NULL;


        out<< " \t\t Enter Full Name         : ";
        out << lat->name<<endl;
        out<< " \t\t Enter Father Name       : ";
        out <<lat->father_name<<endl;
        out<< " \t\t Enter Gender           : ";
        out <<lat->gender<<endl;
        out<< " \t\t Enter Country          : ";
        out <<lat->country_name<<endl;
        out<< " \t\t Enter Date of Birth     : ";
        out <<lat->date_birth<<endl;
        out<< " \t\t Enter Guardian CNIC     : ";
        out <<lat->father_cnic<<endl;
        out<< " \t\t Enter Present Address   : ";
        out <<lat->pres_address<<endl;
        out<< " \t\t Enter Permanent Address  : ";
        out <<lat->perm_address<<endl;
}
else
{
        node *lat=new node;
```

```
cout<< " \t\t Enter Full Name         : ";
cin>>lat->name;
cout<< " \t\t Enter Father Name       : ";
cin>>lat->father_name;
cout<< " \t\t Enter Gender            : ";
cin>>lat->gender;
cout<< " \t\t Enter Country           : ";
cin>>lat->country_name;
cout<< " \t\t Enter Date of Birth     : ";
cin>>lat->date_birth;
cout<< " \t\t Enter Guardian CNIC     : ";
cin>>lat->father_cnic;
cout<< " \t\t Enter Permanent Address  : ";
cin>>lat->perm_address;
cout<< " \t\t Enter Present Address    : ";
cin>>lat->pres_address;
rear_data->next=lat;
rear_data=lat;
lat->next=NULL;

out<< " \t\t Enter Full Name         : ";
out << lat->name<<endl;
out<< " \t\t Enter Father Name       : ";
```

```
                out <<lat->father_name<<endl;
                out<< " \t\t Enter Gender            : ";
                out <<lat->gender<<endl;
                out<< " \t\t Enter Country           : ";
                out <<lat->country_name<<endl;
                out<< " \t\t Enter Date of Birth     : ";
                out <<lat->date_birth<<endl;
                out<< " \t\t Enter Guardian CNIC     : ";
                out <<lat->father_cnic<<endl;
                out<< " \t\t Enter Present Address   : ";
                out <<lat->pres_address<<endl;
                out<< " \t\t Enter Permanent Address  : ";
                out <<lat->perm_address<<endl;
        }
}
        else
                cout <<"THERE IS AN ERROR TO OPEN FILE : ";

}

void Nadra::search(node *temp,int counter)
{
                display_rear(temp,counter);
```

```
            int num;
            cout<< "\n\t\t\t\t\tEnter your token No.: ";
            cin>>num;
            enqueue(num);
            if(num==temp->token_num)
            {
                  cout <<"\n\t\t\t\t\t You may Proceed \n ";
                  system("color 5b");
                  system("color 9f");
                  insert_data();
                  dequeue();
            }
            else if(num>temp->token_num)
            {
                  cout<< "\n\t\t\t\t\t Please Wait for your Turn ";
                  cout<< "\n\t\t\t\t\t People After You "<<num-temp-
>token_num<<endl;
            }
            else if(num>temp->token_num)
                  cout<< "\t\t\t\t\t Your turn is Lost \n Take Token Number Again
\1\2"<<endl;

            else
```

```
                cout<< " \t\t\t\t\t Invalid Number \1\2";


                cout <<"\n\t\t\t\t\t"<<system ("pause");


}
void Nadra::packages(string str)
{
        system("cls");
        ofstream out ("PROJECT.txt",ios::app | ios::out);
        if (out.is_open())
          {
                int pack ;
                again:
                cout <<endl<<endl<<endl;
                char rb=187;
                char midbar = 186;
                char bar = 205;
                char downleftconr=200;
                char doright =188;
                char rightvetbar=187;
                char leftvetbar = 201;
                        cout<<"\t\t\t\t  ";
                cout <<leftvetbar;
```

```
        for (int i=0;i,i<36;i++)
        {
            system ("color 4e");
            cout <<bar;
        }

    cout <<rightvetbar;
    cout<<"\n\t\t\t\t  "<<midbar<<"      SELECT your package.       "<<"
"<<midbar;
    cout<<"\n\t\t\t\t  "<<midbar<<"                                 "<<"   "<<midbar;
    cout<<"\n\t\t\t\t  "<<midbar<<"      Press 1 for Normal         "<<"
"<<midbar;
    cout<<"\n\t\t\t\t  "<<midbar<<"      Press 2 for Urgent         "<<"
"<<midbar;
    cout<<"\n\t\t\t\t  "<<midbar<<"      Press 3 for Executive      "<<"
"<<midbar;

    cout<<"\n\t\t\t\t  "<<midbar<<"            \3 \3           "<<"
"<<midbar<<endl;

    cout<<"\t\t\t\t  ";
        cout <<downleftconr;
        for (int i=0;i,i<36;i++)
        {
```

```
                system ("color 4e");
                cout <<bar;
        }
cout <<doright;
        out <<"\t\t1. Normal. "<<endl;
        out <<"\t\t2. Urgent. "<<endl;
        out <<"\t\t3. Executive. "<<endl;
        cout<<"\n\t\t\t\t  "<<"            "<<"   ";
        cin >>pack ;
        system("cls");
        if(pack==1)
        {
                display_data(str,1000);
        }
        else if(pack==2)
        {
                display_data(str,1500);
        }
        else if(pack==3)
        {
                display_data(str,3000);
        }
        else
```

```
                {
                        cout<< " Please Enter Valid Option "<<endl;
                        goto again;
                }

          }
        else
                cout <<"Error creating file ... "<<endl;
}

// Designing Functions
void Nadra::designverti()
{
        for (int i=0;i<18;i++)
        {
                system ("color 4e");
                cout <<heart<<h1;
        }
}
void Nadra::clear()
{
                cout <<"\n\t\t\t\t";system("pause");
                system("cls");
```

```
}
void Nadra:: display_rear(node* curr,int counter_no)
{
	cout <<endl<<endl<<endl;
		char rb=187;
		char midbar = 186;
		char bar = 205;
		char downleftconr=200;
		char doright =188;
		char rightvetbar=187;
		char leftvetbar = 201;
			cout<<"\t\t\t\t  ";
		cout <<leftvetbar;
		for (int i=0;i,i<36;i++)
		{
			system ("color 4e");
			cout <<bar;
		}

	cout <<rightvetbar;
	cout<<"\n\t\t\t\t  "<<midbar<<"          Nadra          "<<"   "<<midbar;
	cout<<"\n\t\t\t\t  "<<midbar<<"                         "<<"   "<<midbar;
```

```
	cout<<"\n\t\t\t\t  "<<midbar<<"    Welcome To Counter Number "<<counter_no<<"    "<<midbar;

	cout<<"\n\t\t\t\t  "<<midbar<<"                              "<<"   "<<midbar;

	cout<<"\n\t\t\t\t  "<<midbar<<"                              "<<"   "<<midbar;

	cout<<"\n\t\t\t\t  "<<midbar<<"  Current Token Number. is:    "<<curr->token_num<<"    "<<midbar;

	cout<<"\n\t\t\t\t  "<<midbar<<"            \3 \3           "<<"  "<<midbar<<endl;


	cout<<"\t\t\t\t  ";

		cout <<downleftconr;

		for (int i=0;i,i<36;i++)

		{

			system ("color 4e");

			cout <<bar;

		}

	cout <<doright;

};


//Counter Functions

void Nadra::cancel_counter()

{

	if(front_cancel==NULL)

	{
```

```
                node *lat=new node;
                lat->token_num=1;
                lat->next=NULL;
                front_cancel=lat;
                rear_cancel=lat;
        }
        else
        {
                node *lat=new node;
                lat->token_num=rear_cancel->token_num;
                lat->token_num++;
                rear_cancel->next=lat;
                rear_cancel=lat;
                lat->next=NULL;
        }
}
void Nadra::origin_counter()
{
        if(front_origin==NULL)
        {
                node *lat= new node;
                lat->token_num=1;
                lat->next=NULL;
```

```
            front_origin=lat;
            rear_origin=lat;
        }
        else
        {
            node *lat=new node;
            lat->token_num=rear_origin->token_num;
            lat->token_num++;
            rear_origin->next=lat;
            rear_origin=lat;
            lat->next=NULL;
        }
}
void Nadra::child_counter()
{
        if(front_child==NULL)
        {
            node *lat=new node;
            lat->token_num=1;
            lat->next=NULL;
            front_child=lat;
            rear_child=lat;
        }
```

```
        else
        {
                node *lat=new node;
                lat->token_num=rear_child->token_num;
                lat->token_num++;
                rear_child->next=lat;
                rear_child=lat;
                lat->next=NULL;
        }
}
void Nadra::smart_counter()
{
        if(front_smart==NULL)
        {
                node *lat=new node;
                lat->token_num=1;
                lat->next=NULL;
                front_smart=lat;
                rear_smart=lat;
        }
        else
        {
                node *lat=new node;
```

```
            lat->token_num=rear_smart->token_num;
            lat->token_num++;
            rear_smart->next=lat;
            rear_smart=lat;
            lat->next=NULL;
        }
}
void Nadra::cnic_counter()
{
        if(front_cnic==NULL)
        {
            node *lat=new node;
            lat->token_num=1;
            lat->next=NULL;
            front_cnic=lat;
            rear_cnic=lat;
        }
        else
        {
            node *lat=new node;
            lat->token_num=rear_cnic->token_num;
            lat->token_num++;
            rear_cnic->next=lat;
```

```
            rear_cnic=lat;
            lat->next=NULL;
        }
}
void Nadra::family_counter()
{
        if(front_family==NULL)
        {
            node *lat=new node;
            lat->token_num=1;
            lat->next=NULL;
            front_family=lat;
            rear_family=lat;
        }
        else
        {
            node *lat=new node;
            lat->token_num=rear_family->token_num;
            lat->token_num++;
            rear_family->next=lat;
            rear_family=lat;
            lat->next=NULL;
        }
```

```
}

// Show Choices Functions
void Nadra::menu()
{

        cout <<"\n\n\t\t\t     ";designverti();cout <<endl;
        cout << "\t\t\t    "<<"\t                          "<<endl;
        cout << "\t\t\t    "<<"\t NADRA TICKET TOKEN SYSTEM       "<<endl ;
        cout << "\t\t\t    "<<"\t                          "<<endl;
        cout <<"\t\t\t     ";designverti();cout <<endl;
        cout << "\t\t\t    "<<"\t                          "<<endl;
    cout << "\t\t\t    "<<"\t 1. Child Registeration          "<< endl;
        cout << "\t\t\t    "<<"\t 2. National ID Card            "<< endl;
        cout << "\t\t\t    "<<"\t 3. Smart Card                  "<< endl;
        cout << "\t\t\t    "<<"\t 4. Family Registeration        "<< endl;
        cout << "\t\t\t    "<<"\t 5. Pakistan Origin Card        "<< endl;
        cout << "\t\t\t    "<<"\t 6. Cancelation Certificate      "<< endl;
        cout << "\t\t\t    "<<"\t 7. EXIT  \3                    "<< endl;
        cout << "\t\t\t    "<<"\t                          "<< endl;
        cout <<"\t\t\t     ";designverti();cout <<endl;
        cout <<"\n\n\t\t\t    "<<"\t   Pick any of the following service  \2: ";
}
```

```
void Nadra::choices()
{
	ofstream out ("PROJECT.txt",ios::app | ios::out| ios::in);
	int choice,option;
	do
{
		system("cls");
		menu();
		cin>>choice;
		system("color 47");
	if(choice==1)
	{
		string str="Child Registeration ";
		cout<< "\n\t\t\t\t Press 8 if you want to Receive your Form "<<endl;
		cout<< "\n\t\t\t\t Press 9 if You want to Apply for Form"<<endl;
		cout<< "\n\t\t\t\t ";
		cin>>option;
		if(option==8)
		{
			search_data();
			system("pause");
		}
		else if(option==9)
```

```
            {
            child_counter();
            node *curr=rear_child;
            cout<< "\n\t\t\t Please Take Token Number and Proceed To Counter Number 1 \n\t\t\t\t\t";
            system("pause");
            system("cls");
            system("color 4f");
            out <<"\n  \t\tDATA FOR CHILD REGISTRATION : "<<endl;

            search(curr,choice);
            packages(str);
            system("pause");
            }
        }
        else if(choice==2)
        {
            cout<< "\n\t\t\t\t Press 8 if you want to Receive your Form "<<endl;
            cout<< "\n\t\t\t\t Press 9 if You want to Apply for Form"<<endl;
            cout<< "\n\t\t\t\t ";
            cin>>option;
            if(option==8)
            {
```

```
                search_data();
                system("pause");


            }
            else if(option==9)
            {
            cnic_counter();
            node *curr=rear_cnic;
            cout<< "\n\t\t\t Please Take Token Number and Proceed To Counter Number 2 \n\t\t\t\t\t";
            system("pause");
            system("cls");
            system("color 4f");
            out <<"\n  \t\tDATA FOR CNIC REGISTRATION : "<<endl;
            search(curr,choice);


            string str=" National ID Card ";
            packages(str);
            system("pause");
            }
        }
        else if(choice==3)
        {
```

```
cout<< "\n\t\t\t\t Press 8 if you want to Receive your Form "<<endl;
cout<< "\n\t\t\t\t Press 9 if You want to Apply for Form"<<endl;
cout<< "\n\t\t\t\t ";
cin>>option;
if(option==8)
{
	search_data();
	system("pause");

}
else if(option==9)
{
smart_counter();
node *curr=rear_smart;
cout<< "\n\t\t\t Please Take Token Number and Proceed To Counter Number 3 \n\t\t\t\t\t";
system("pause");
system("cls");
system("color 4f");
out <<"\n  \t\tDATA FOR SMART CARD REGISTRATION : "<<endl;
search(curr,choice);
```

```
            string str="Smart Card" ;
            packages(str);
            system("pause");
            }
        }
        else if(choice==4)
        {
            cout<< "\n\t\t\t\t Press 8 if you want to Receive your Form "<<endl;
            cout<< "\n\t\t\t\t Press 9 if You want to Apply for Form"<<endl;
            cout<< "\n\t\t\t\t ";
            cin>>option;
            if(option==8)
            {
                search_data();
                system("pause");
            }
            else if(option==9)
            {
            family_counter();
            node *curr=rear_family;
            cout<< "\n\t\t\t Please Take Token Number and Proceed To Counter
Number 4 \n\t\t\t\t\t";
            system("pause");
```

```
        system("cls");
        system("color 9f");
        out <<"\n  \t\tDATA FOR FAMILY REGISTRATION \3 : "<<endl;

        search(curr,choice);

        string str="Family Registeration ";
        packages(str);
        system("pause");
        }
}
else if(choice==5)
{
        cout<< "\n\t\t\t\t Press 8 if you want to Receive your Form "<<endl;
        cout<< "\n\t\t\t\t Press 9 if You want to Apply for Form"<<endl;
        cout<< "\n\t\t\t\t ";
        cin>>option;
        if(option==8)
        {
                search_data();
                system("pause");
        }
        else if(option==9)
```

```
            {
            origin_counter();
            node *curr=rear_origin;
            cout<< "\n\t\t\t Please Take Token Number and Proceed To Counter Number 5 \n\t\t\t\t\t";
            system("pause");
            system("cls");
            system("color 9f");
            out <<"\n  \t\tDATA FOR ORIGIN REGISTRATION  \3 : "<<endl;

            search(curr,choice);

            string str=" Origin Card ";
            packages(str);
            system("pause");
            }
        }
        else if(choice==6)
        {
            cout<< "\n\t\t\t\t Press 8 if you want to Receive your Form "<<endl;
            cout<< "\n\t\t\t\t Press 9 if You want to Apply for Form"<<endl;
            cout<< "\n\t\t\t\t ";
            cin>>option;
```

```
            if(option==8)
            {
                search_data();
                system("pause");
            }
            else if(option==9)
            {
            cancel_counter();
            node *curr=rear_cancel;
            cout<< "\n\t\t\t Please Take Token Number and Proceed To Counter Number 6 \n\t\t\t\t\t";
            system("pause");
            system("cls");
            system("color 9f");
            out <<"\n  \t\tDATA FOR CACELATION REGISTRATION \3 : "<<endl;
            search(curr,choice);

            string str="Cancellation Certificate";
            packages(str);
            system("pause");
            }
        }
```

```
        else if(choice>7 || choice<=0 )
        {
                system("cls");
                system("color 9f");
                cout<< " \n  \t\t Invalid Option \n \t\t Please Choose Valid Options
"<<endl;
                system("pause");
        }
}
        while(choice!=7);
}

// Token Insertion and Deletion Function
void Nadra::enqueue(int num)
{
        if(front==NULL)
        {
                node *lat=new node;
                lat->token_num=num;
                lat->next=NULL;
                front=lat;
                rear=lat;
        }
```

```
        else
        {
                node *lat=new node;
                lat->token_num=num;
                rear->next=lat;
                lat->next=NULL;
        }
}
void Nadra::dequeue()
{
        if (front==NULL)
                cout<< " List is Empty ";
        else
        {
                node *kill_me=front;
                front=kill_me->next;
                delete kill_me;
        }
}

//Driver Function
int main()
{
```

```
    Nadra n;
    n.choices();
    system("pause");
    return 0;
}
```