DOMAIN : PAYMENT

GROUP 2

TITLE : Zoo Park

**REPORT**

Systematic observation and record keeping are essential for consistent advances in the management of zoos and related facilities. This project aims to develop a program of payment in order to generate a better system with better features and qualities. Through the use of modern technologies in evolving a system to multi-function structure will increase employee efficiency in terms of customer services. Advance and modern systems with special characteristic will surely benefits the company in making work faster and easier with higher productivity. Thus, improvement in systems should always be updated to maintain a good and excellent service.

This program created with services include the ability to add customers’ name and age, capability to edit, search, display and delete names besides recording customer’s categories. As such, this will bring excellent service and satisfaction to customers. Lastly, we wish you to have an unforgettable holiday adventure in Zoo Taiping, experiencing an awesome wildlife encounter.

**Pseudocode**

Start

**define** main( )

Declaration

choice, count=0, []name, []numOfChildren, []numOfAdult, []numOfSeniorCitizens, []price,

[]statusOfPayment, totalNumOfChildren, totalNumOfAdult, totalNumOfSeniorCitizens,

totalTicket,totalSales

do while(true)

menu()

Input choice

Switch(choice)

Case 0

then System.exit(0)

END

Case 1

then count=addName(statusOfPayment, name, count, numOfChildren,

numOfAdult, numOfSeniorCitizens, price)

break

Case 2

then editName(statusOfPayment, name, count, numOfChildren,

numOfAdult, numOfSeniorCitizens)

break

Case 3

then searchName(statusOfPayment, name, count, numOfChildren,

numOfAdult, numOfSeniorCitizens)

break

Case 4

then displayName(statusOfPayment, totalNumOfChildren,

totalNumOfAdult, totalNumOfSeniorCitizens,

totalSales, totalTicket, name, numOfChildren,

numOfAdult, numOfSeniorCitizens, count, price)

break

Case 5

then count=deleteName(statusOfPayment, name, count, numOfChildren,

numOfAdult, numOfSeniorCitizens)

break

default

then Display Error!Invalid.

break

endwhile

End

**define** menu()

Start

Display ‘menu’

End

**define** addName(int [] statusOfPayment, String []name, int count, int [] numOfChildren,

int [] numOfAdult, int [] numOfSeniorCitizens, double [] price)

Start

if count equal to name.length

Display ‘ticket is full’

else

then Input name, numOfChildren, numOfAdult, numOfSeniorCitizens

name[count]=name[count].toUpperCase()

statusOfPayment[count]=1

count++

**return** count

End

**define** editName(int [] statusOfPayment, String []name, int count, int [] numOfChildren,

int [] numOfAdult, int [] numOfSeniorCitizens)

Start

Declaration

flag=0,n,yN

if count equal to 0

Display ‘Ticket is Empty’

else

Input n

then n=toUpperCase()

for int i equal to 0; i less than count; i++

if name[i].equals(n)

if statusOfPayment[i] equals to 0

then Display ‘Cannot edit because customer have already paid.’

flag=1

else

Input Edit the name

name[i]=name[i].toUpperCase()

Input y/n to edit the number of Children, Adult and Senior Citizens

While(true)

if yN equals(“y”) or yN equals(“Y”)

Input Edit the number of Children, Adult and Senior Citizens

break

else if

yN equals(“n”) or yN equals(“N”)

break

else

Display “Invalid input,please try again”

Input y/n to edit the number of Children, Adult and Senior

Citizens

endWhile

Display successfully edit

flag=1

endWhile

if flag equals to 0

Display ‘n is not in the list, please re-enter to edit again’

End

**define** searchName(statusOfPayment, name, count numOfChildren, numOfAdult,

numOfSeniorCitizens)

Start

Declaration

flag=0,payment, PRICE\_OF\_CHILDREN, PRICE\_OF\_ADULT,

PRICE\_OF\_SENIOR\_CITIZENS, n,

yN, balance

if count equal to 0

Display ‘ticket is empty’

else

Input n

n=n.toUpperCase()

for int i equal to 0; i less than count; i++

if name[i].equals(n)

Display successful search! n is in the list of ticket

if statusOfPayment[i] equals to 0

Display “Already paid”

then flag=1

else

then totalPrice = (PRICE\_OF\_CHILDREN\*numOfChildren[i])

+(PRICE\_OF\_ADULT\*numOfAdult[i])

+(PRICE\_OF\_SENIOR\_CITIZENS\*

numOfSeniorCitizens[i]);

Display Number of Children, Number of Adult,

Number of Senior Citizens, Total Price

Input y/n for customer payment

while(true)

if yN.equals(“y”) or yN.equals(“Y”)

Input payment

while(true)

then balance=payment-totalPrice

if balance equals to 0

Display Num, Name, Number of Children,

Number of Adult, Number of Senior

Citizens, Total Price

break

else if balance more than 0

Display Num, Name, Number of Children

Number of Adult, Number of

Senior Citizens, Total Price, balance

break

else

Display “Sorry, this is not enough, you need

to pay more.”

Input Extra Payment

then payment+=input

endWhile

break

else if yN.equals(“n”) or yN.equals(“N”)

break

else

Display “Invalid input, please try again”

Input y/N to edit number of Children, Adult and Senior

Citizens

endWhile

flag=1

endFor

if flag equals to 0

Display n is not in the list, please re-enter to search again.

End

**define** displayName(int [] statusOfPayment, int totalNumOfChildren, int totalNumOfAdult,

int totalNumOfSeniorCitizens, int totalSales, int totalTicket,

String []name, int [] numOfChildren, int [] numOfAdult,

int [] numOfSeniorCitizens, int count, double [] price )

Start

Declaration

list[][], totalPrice, num

if count equals to 0

Display ‘ticket is empty’

else

num=1

Display “Num, Name, Num Of Children, Num Of Adult, Num Of Senior

Citizens, Price(RM), Status Of Payment(1=no, 0=yes)”

n=n.toUpperCase()

for int j equal to 0; j less than price.length; j++

then totalPrice=(numOfChildren[j]\*8)+(numOfAdult[j]\*16)+

(numOfSeniorCitizens[j]\*0)

price[j]=totalPrice

endWhile

for int i equal to 0; i less than price.length; i++

Display num, name[i]

endWhile

for int j equals to 0; j less than 5; j++

then list[i][0]=numOfChildren[i]

list[i][1]=numOfAdult[i]

list[i][2]=numOfSeniorCitizens[i]

list[i][3]=(int)price[i]

list[i][4]=statusOfPayment[i]

Display list[i][j]

num++

endWhile

for int i equals to 0; i less than count; i++

then totalNumOfChildren add and assign to numOfChildren[i]

totalNumOfAdult add and assign to numOfAdult[i]

totalNumOfSeniorCitizens add and assign to numOfSeniorCitizens[i]

totalSales add and assign to price[i]

endWhile

then totalTicket = totalNumOfChildren + totalNumOfAdult +

totalNumOfSeniorCitizens

Display totalNumOfChildren, totalNumOfAdult,

totalNumOfSeniorCitizens, totalTicket, totalSales

End

**define** deleteName(int [] statusOfPayment, String []name, int count, int [] numOfChildren,

int [] numOfAdult, int [] numOfSeniorCitizens)

Start

Declaration

right=false

n

if count equal to 0

Display ‘ticket is empty’

else

Display ‘Enter a name to be deleted’

Input n

then n=n.toUpperCase()

for int i equal to 0; i less than count; i++

if name[i].equals(n)

if statusOfPayment[i] equals to 0

Display “Cannot be deleted, customer have already paid.”

right=true

else

for int j=i; j<count-1; j++

then name[j]=name[j+1]

statusOfPayment[j]=statusOfPayment[j+1]

numOfChildren[j]=numOfChildren[j+1]

numOfAdult[j]=numOfAdult[j+1]

numOfSeniorCitizens[j]=numOfSeniorCitizens[j+1]

endWhile

Display “Successful deleted!”

then count--

right=true

endWhile

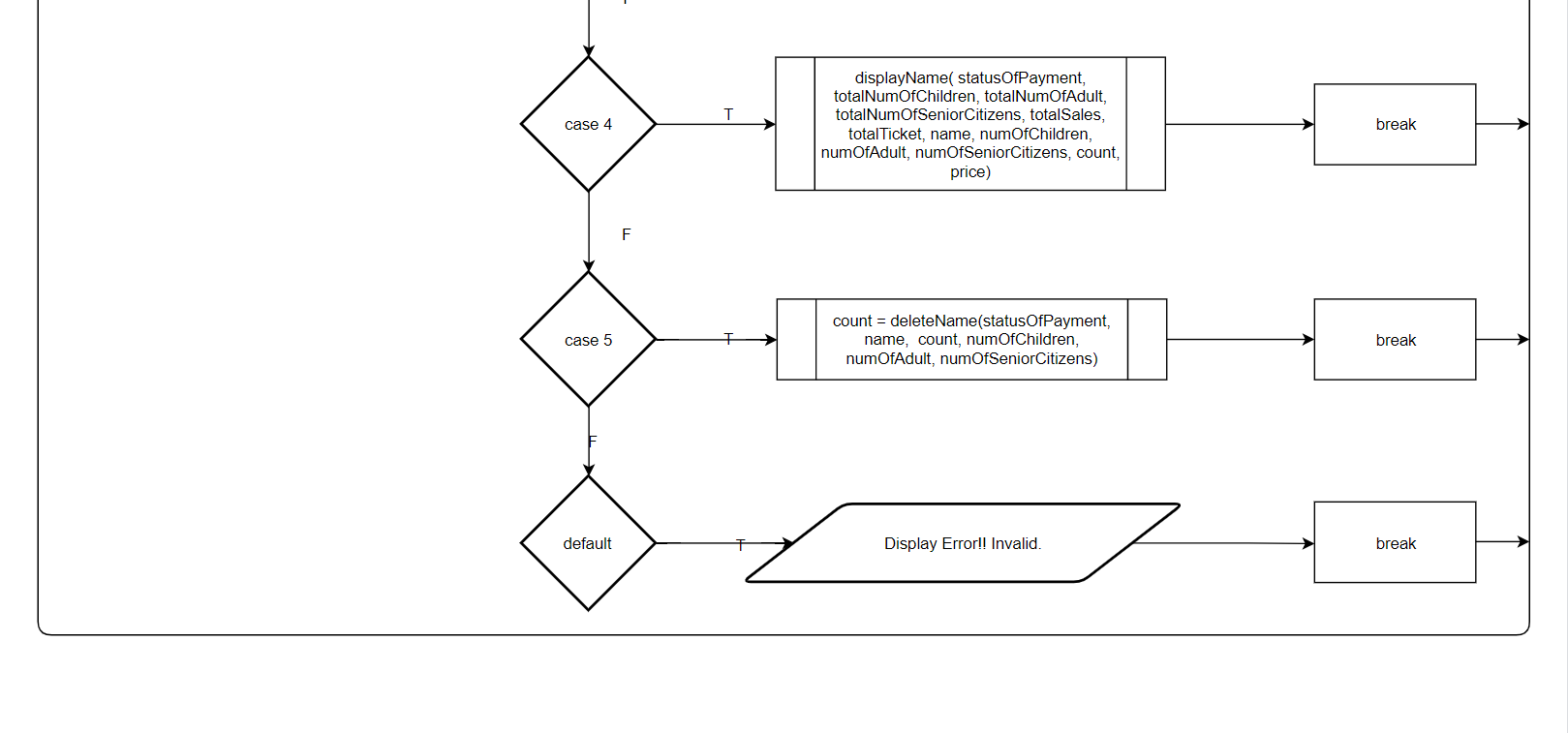
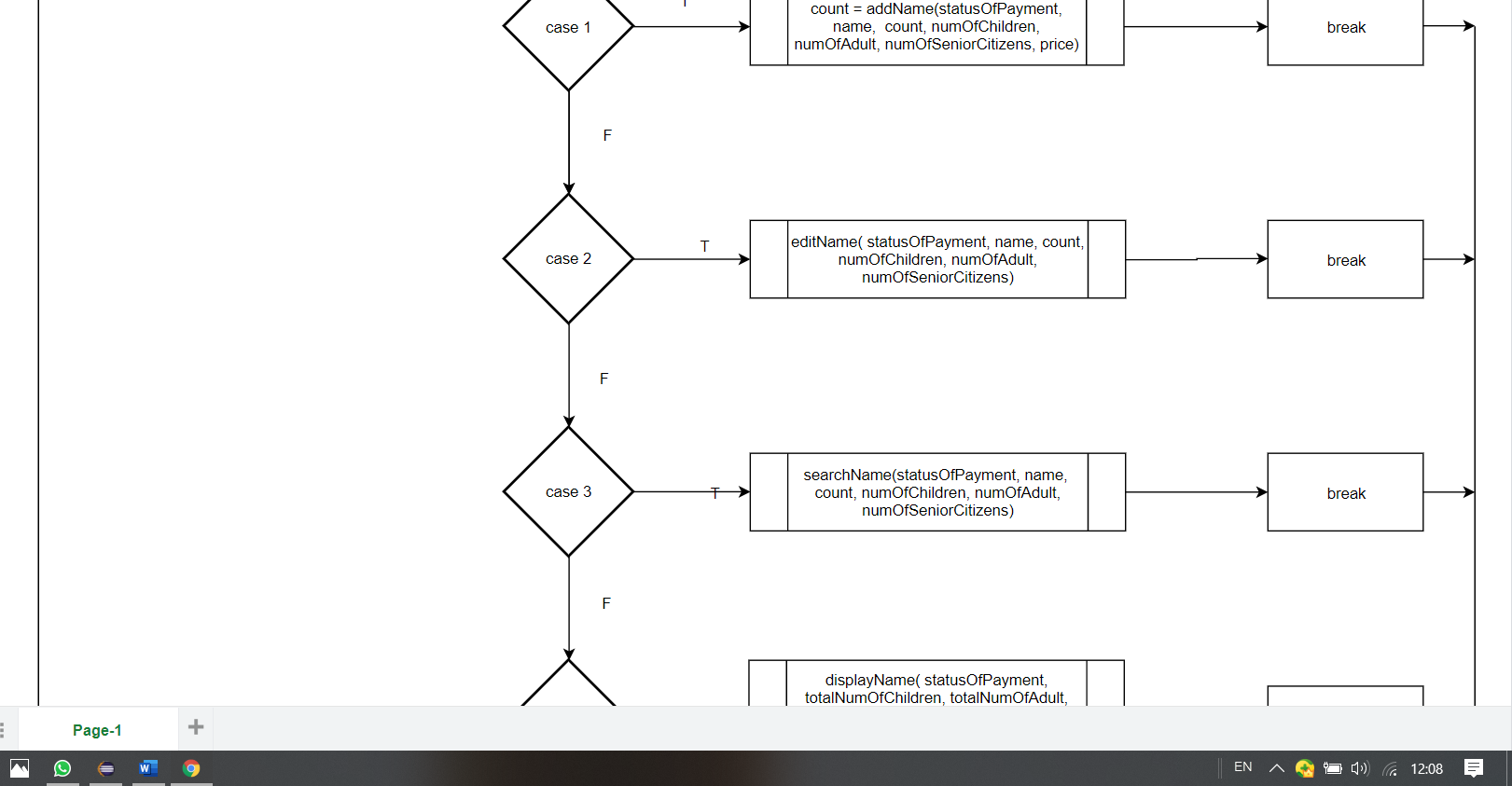
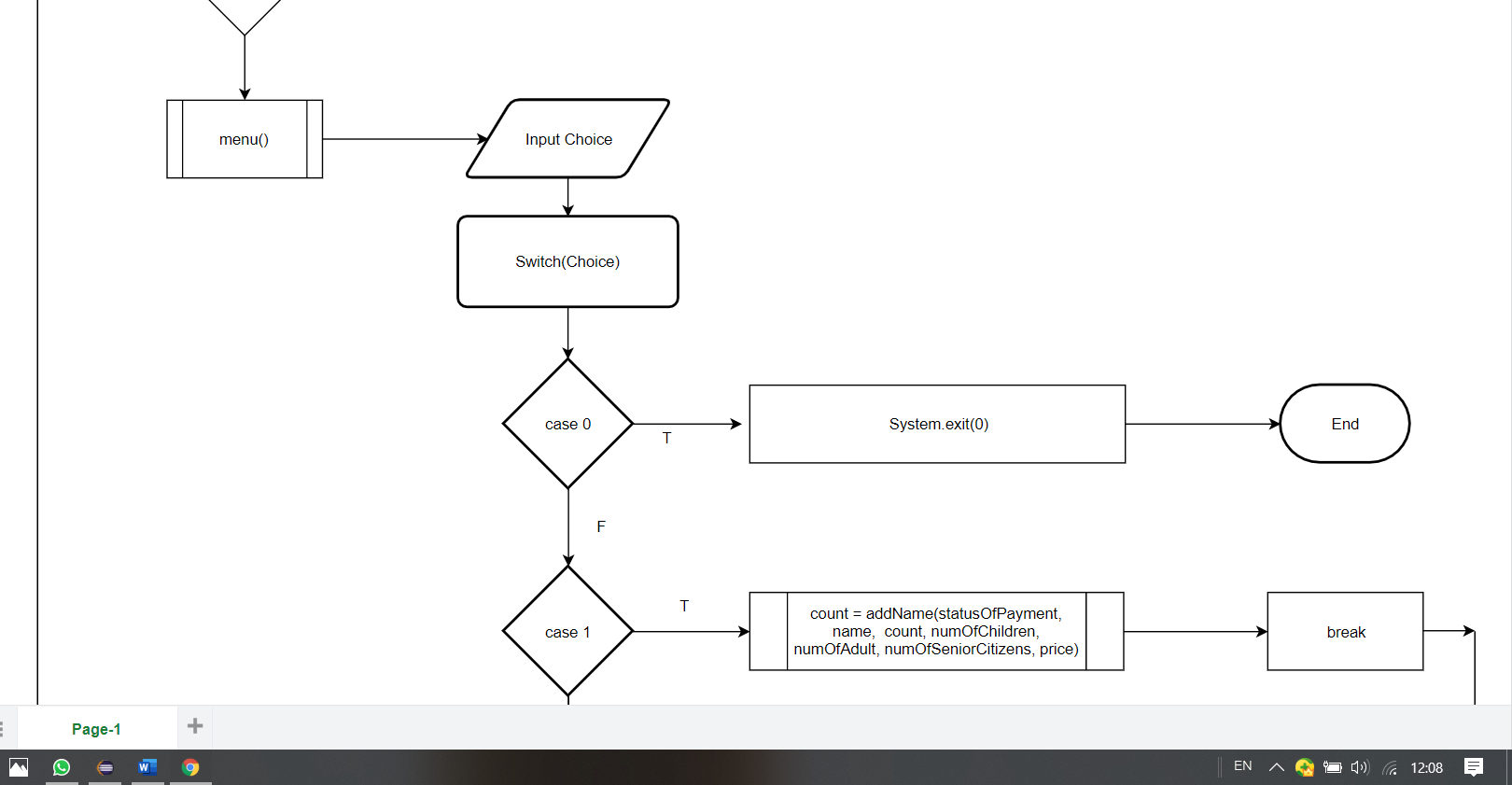
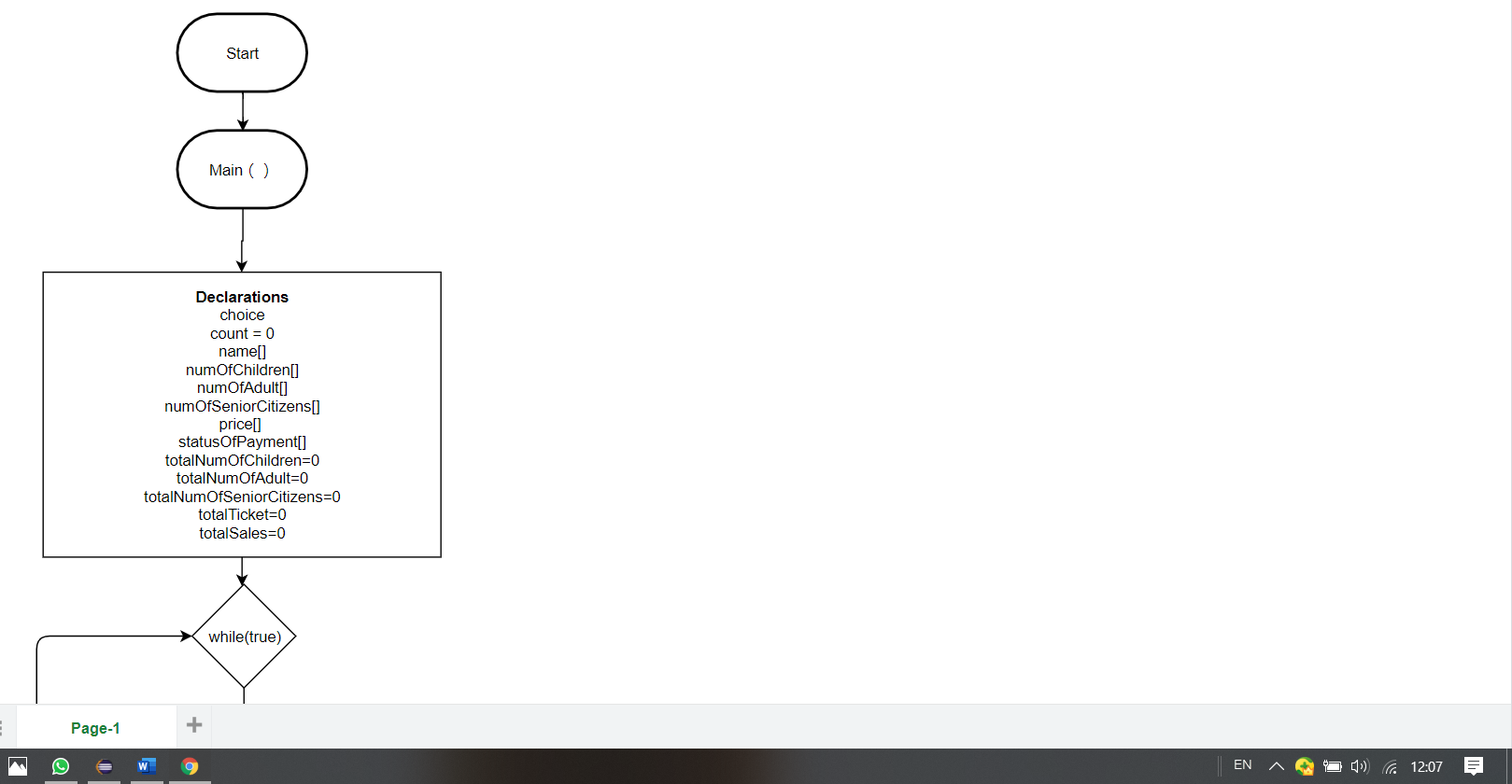
if right equal to false

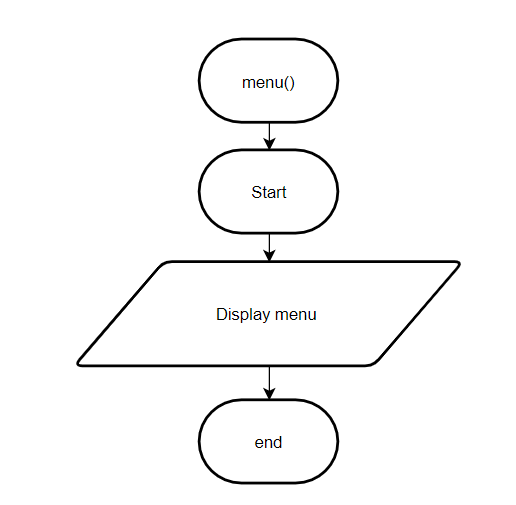
Display ‘n is not in the list, please re-enter to search again’

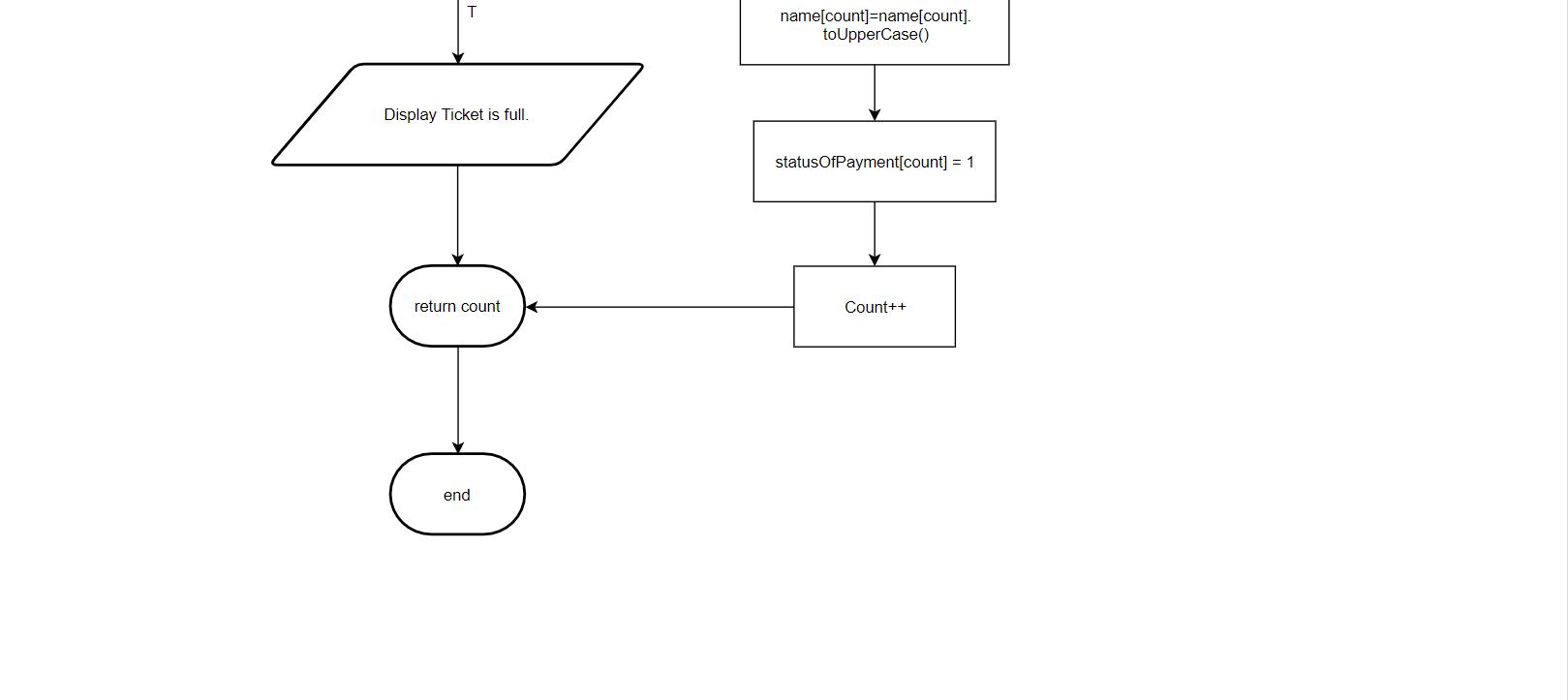
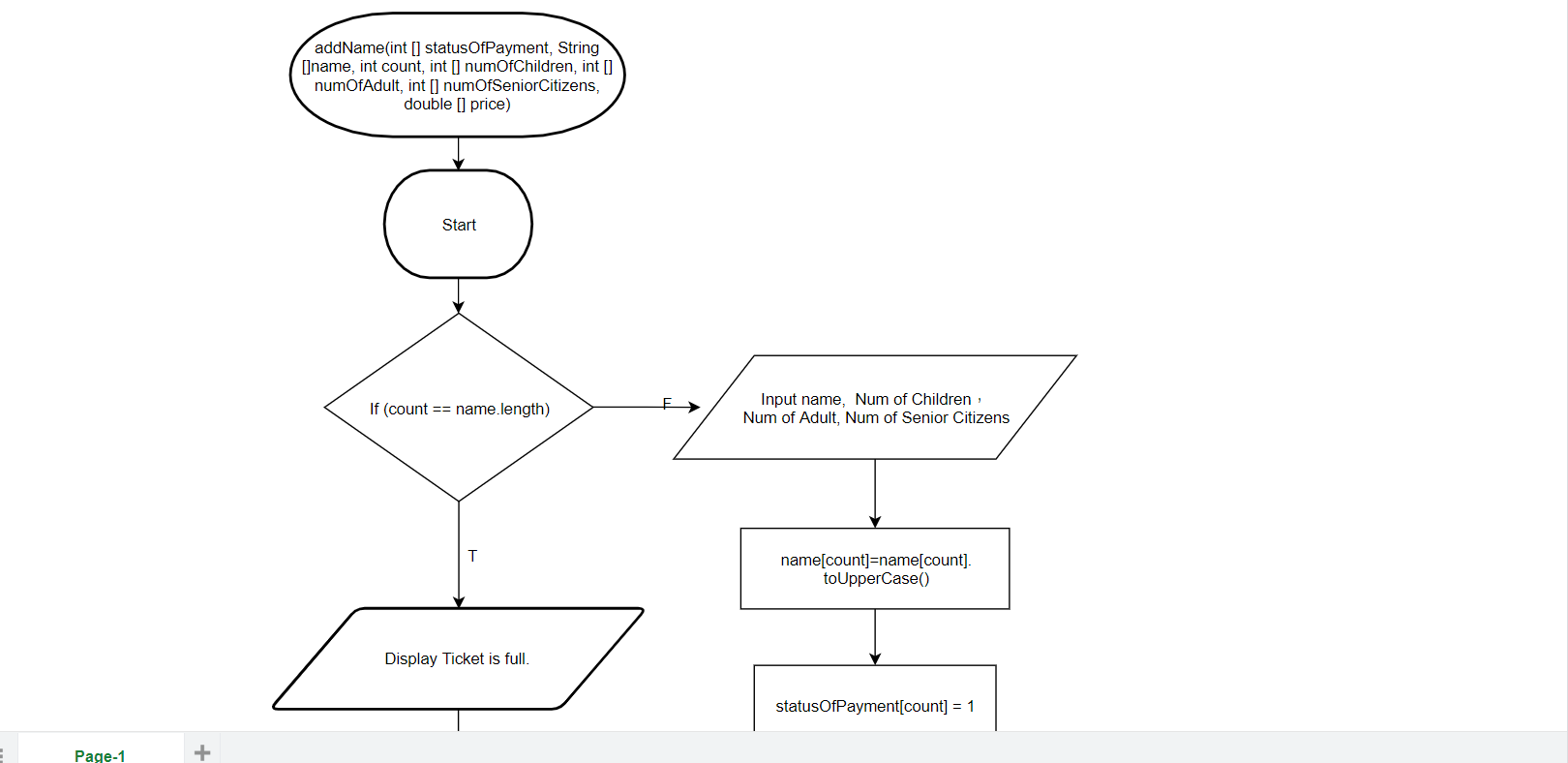
**return** count

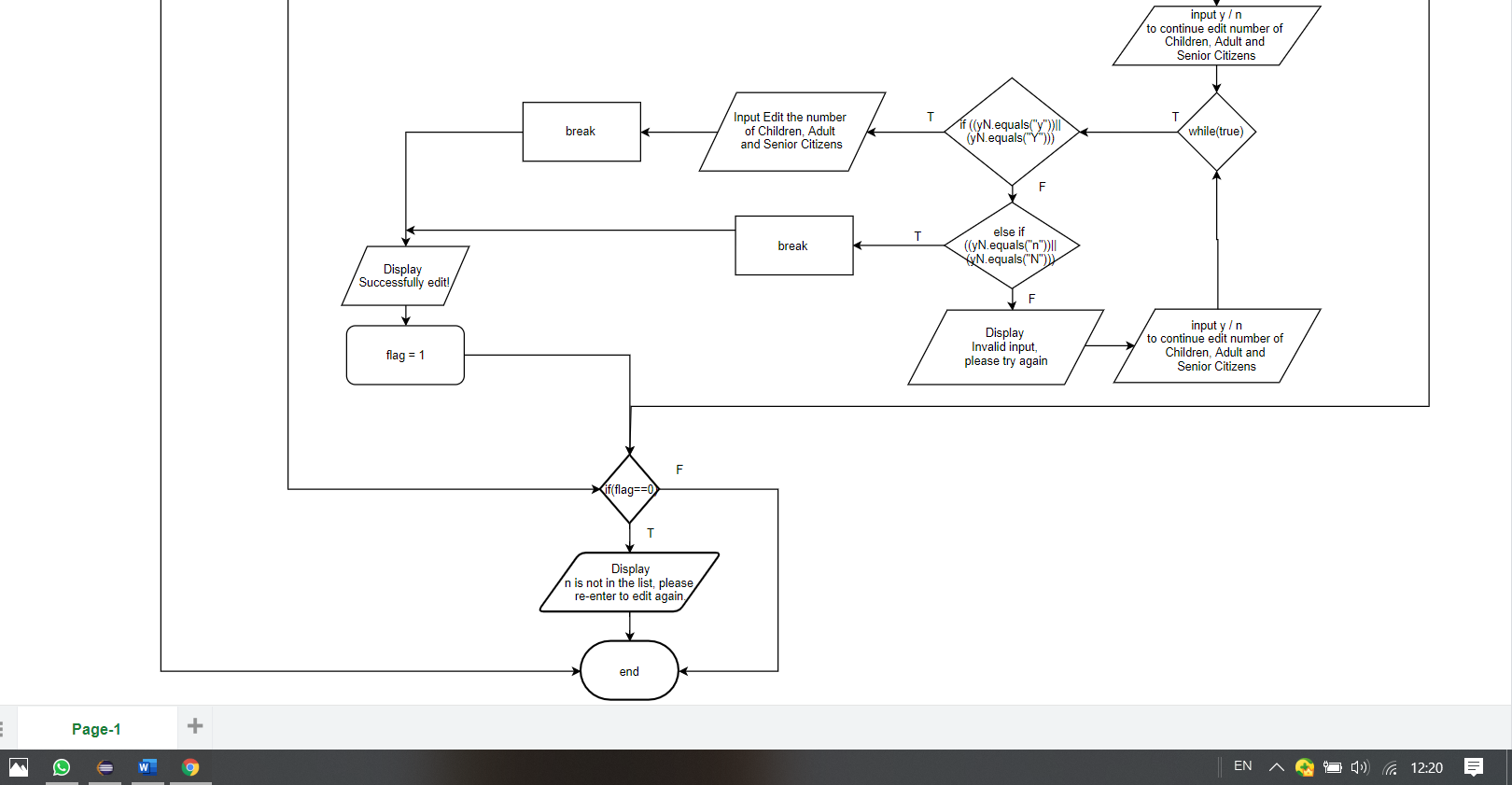
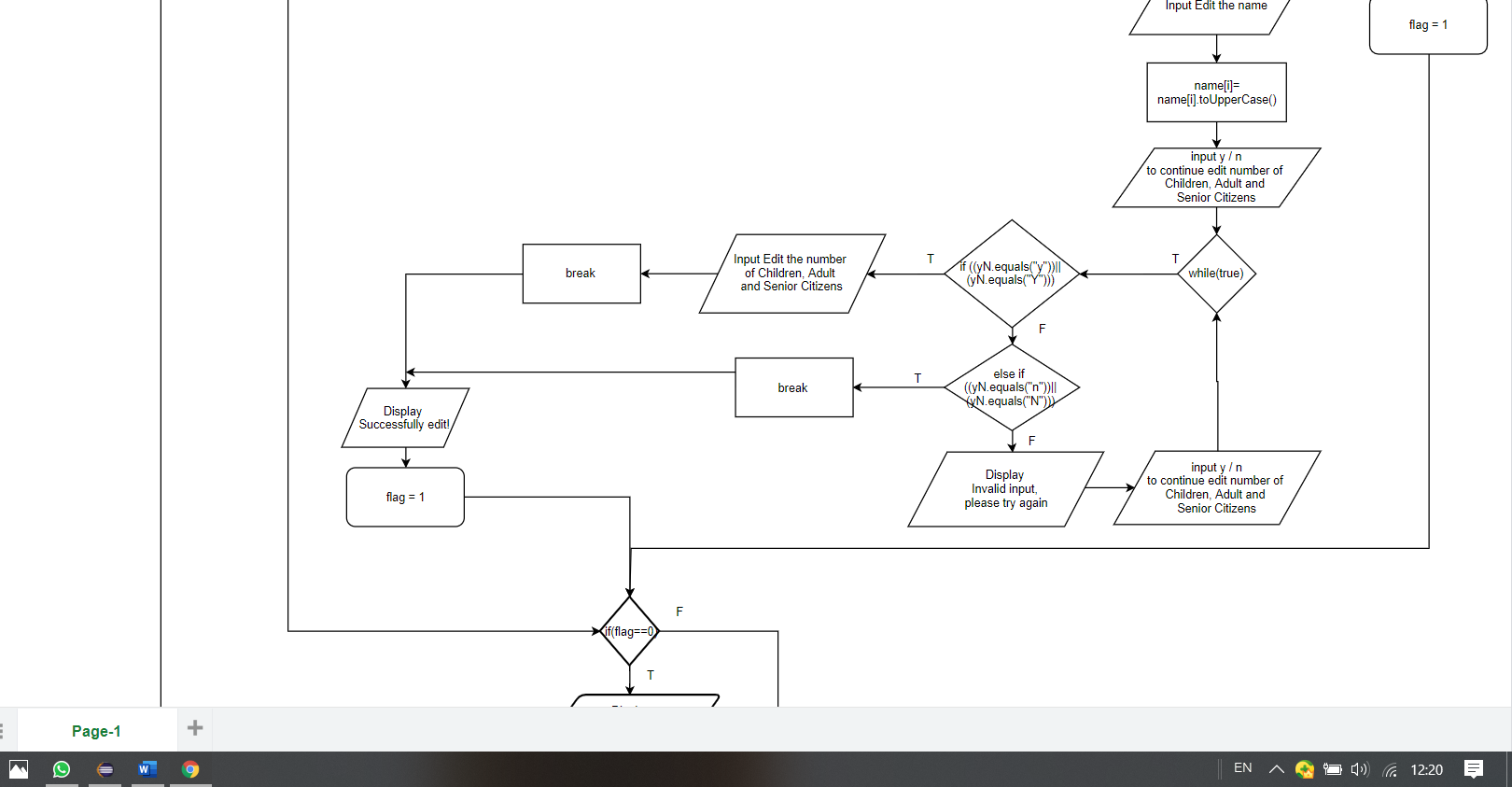
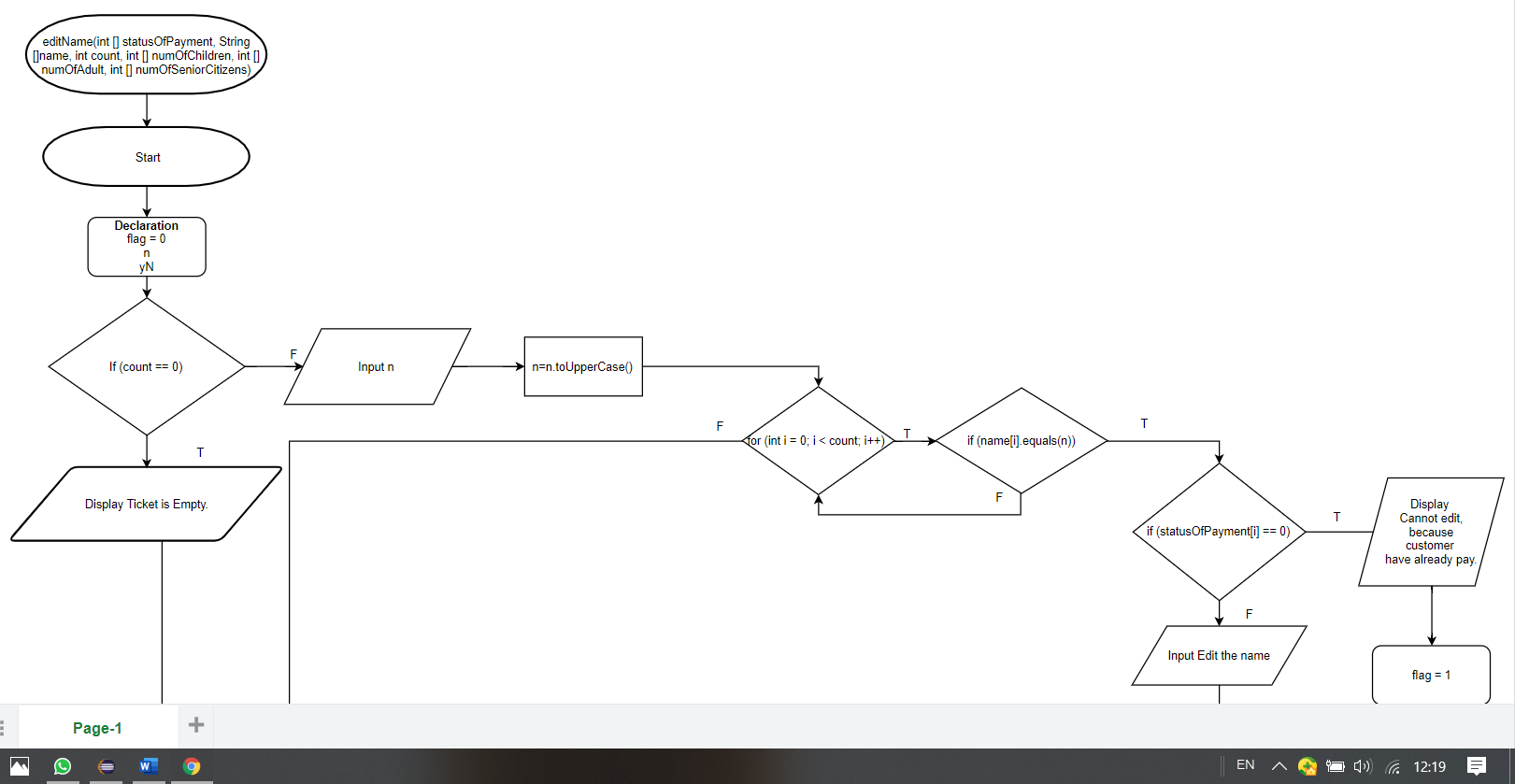
END

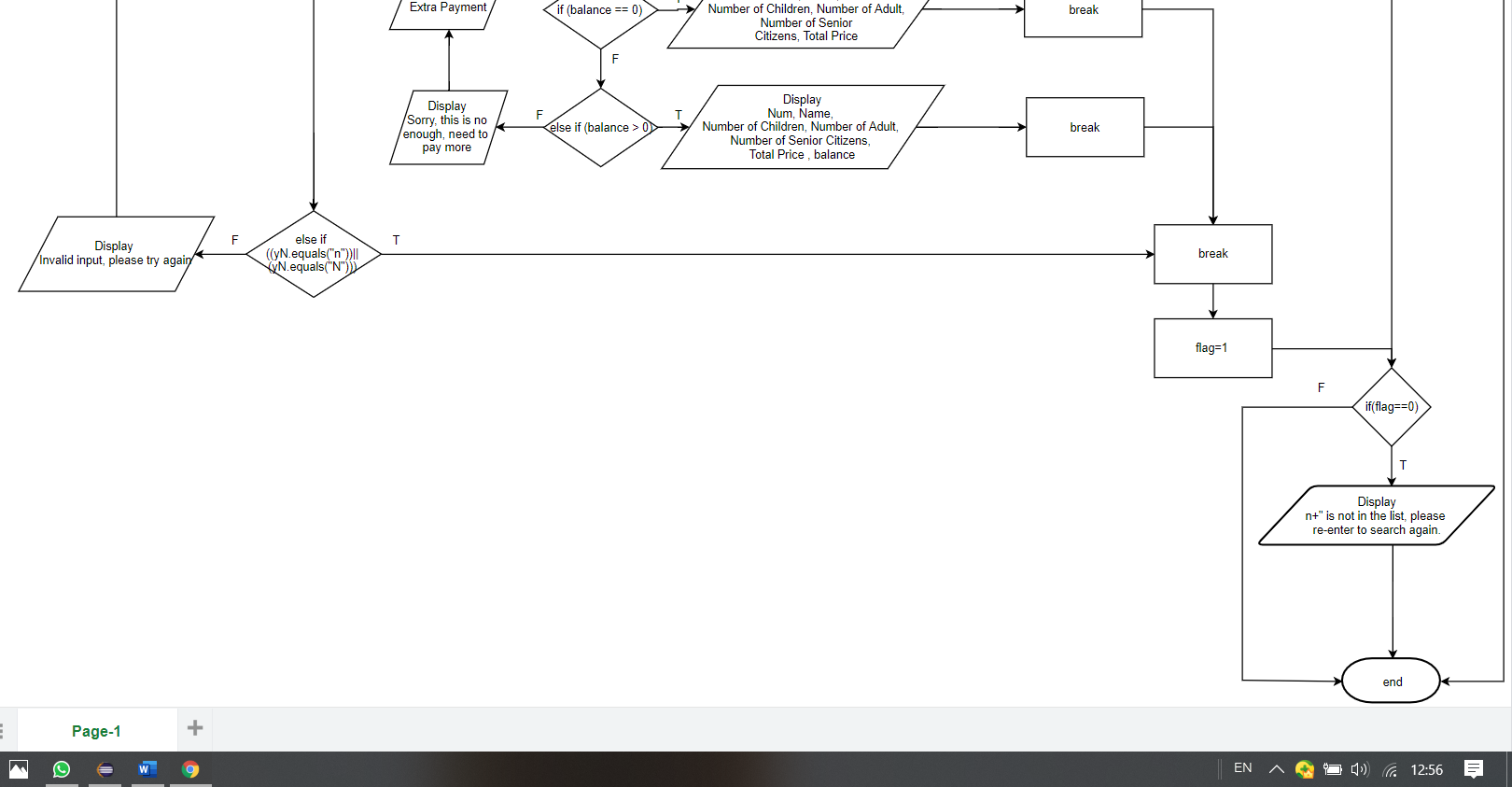
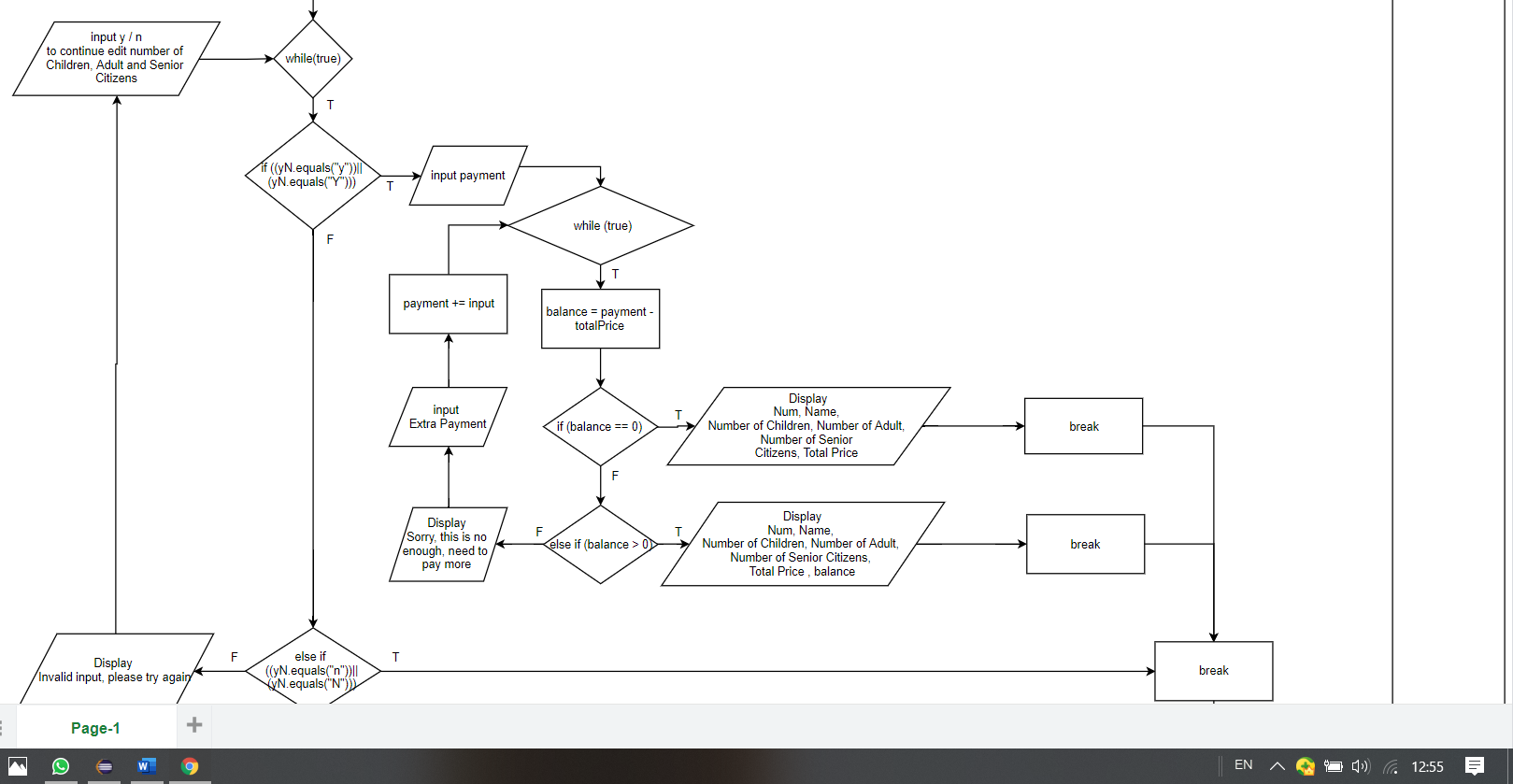
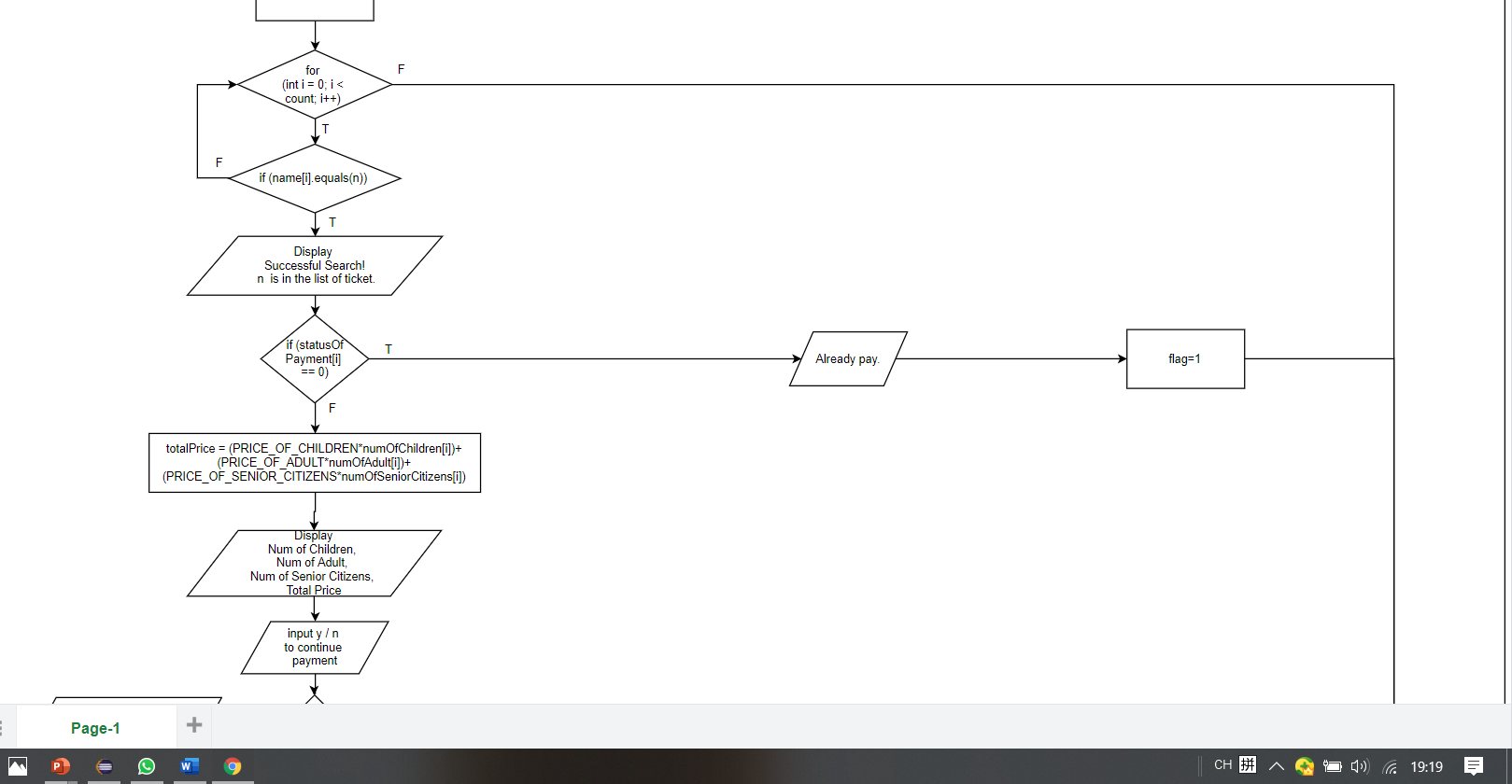
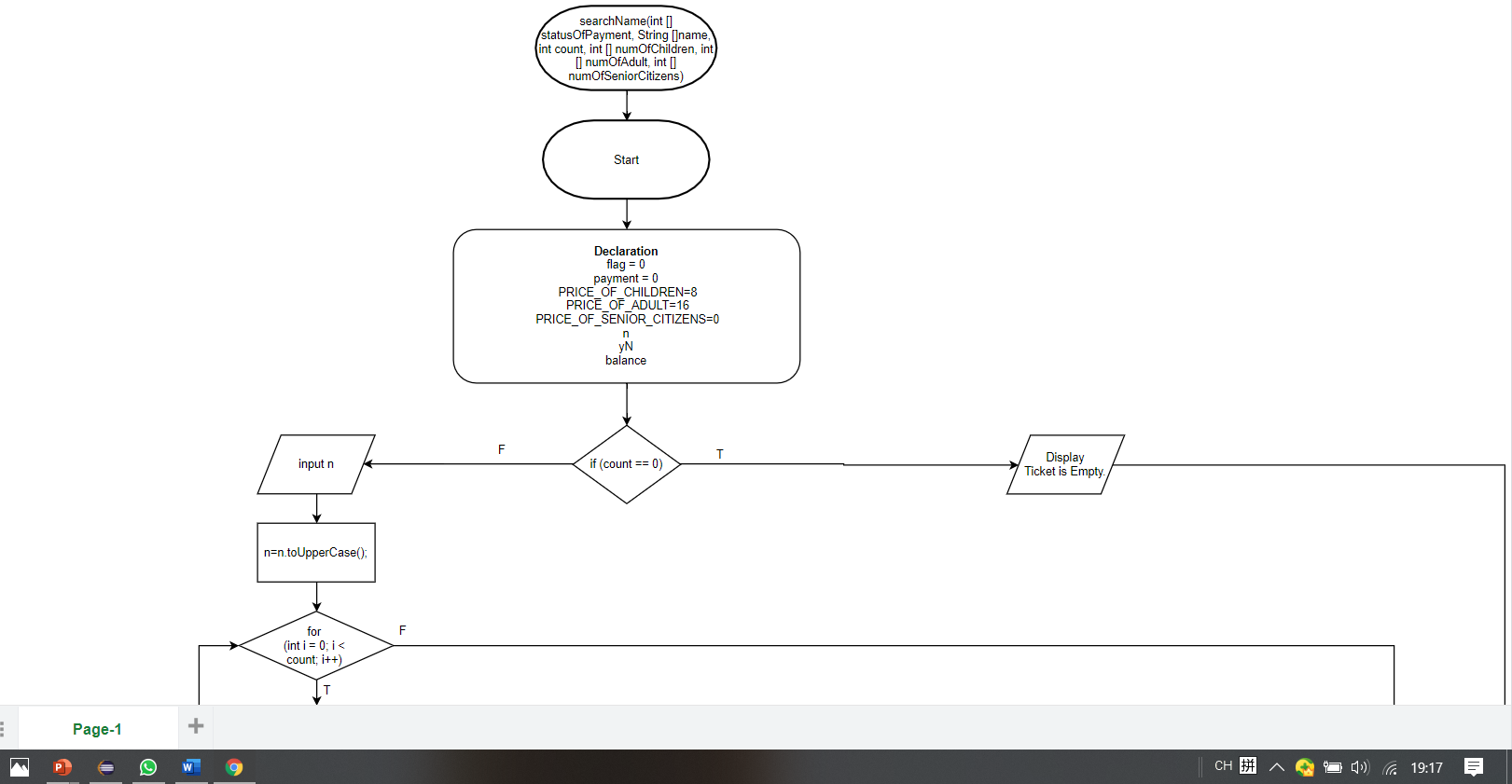
**Flowchart**

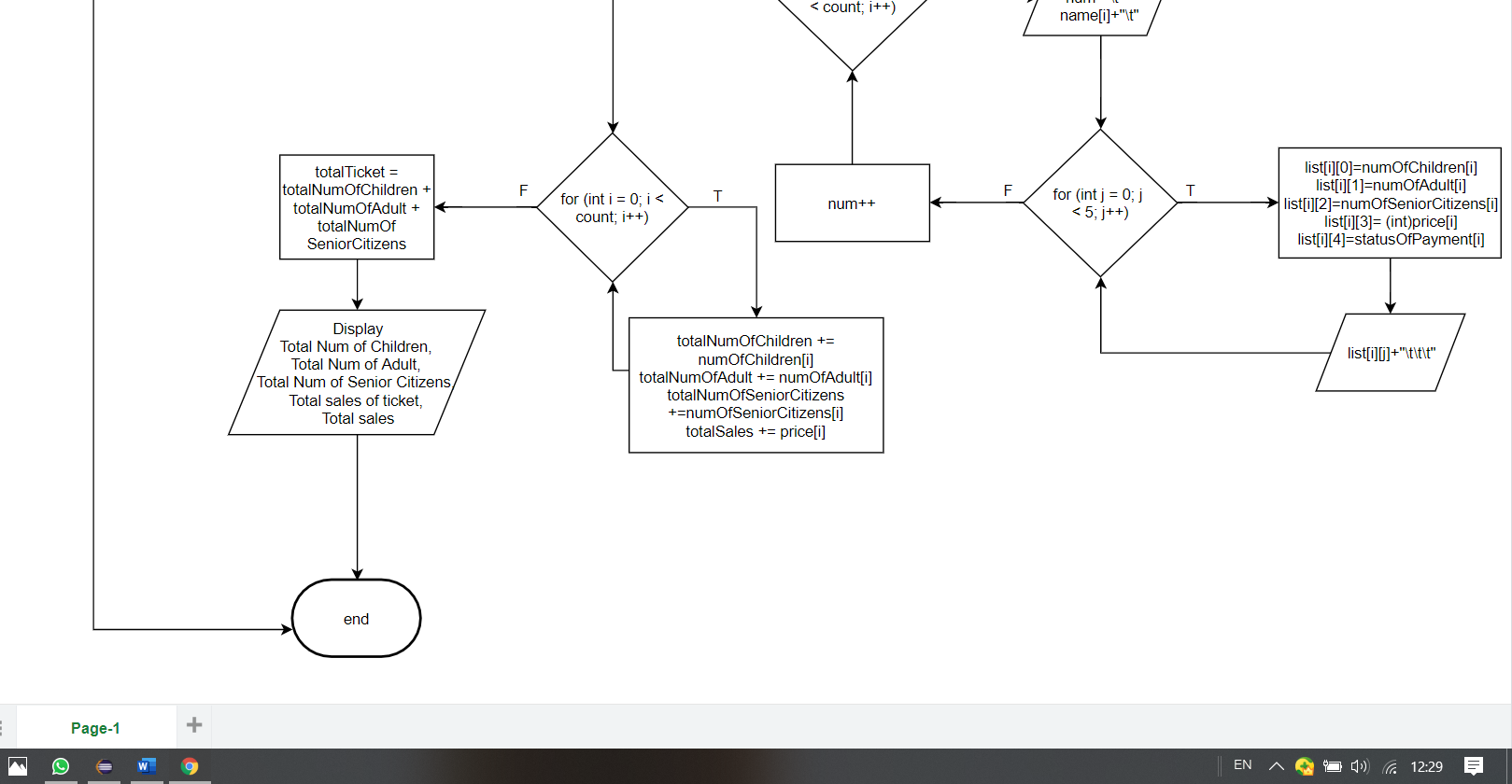
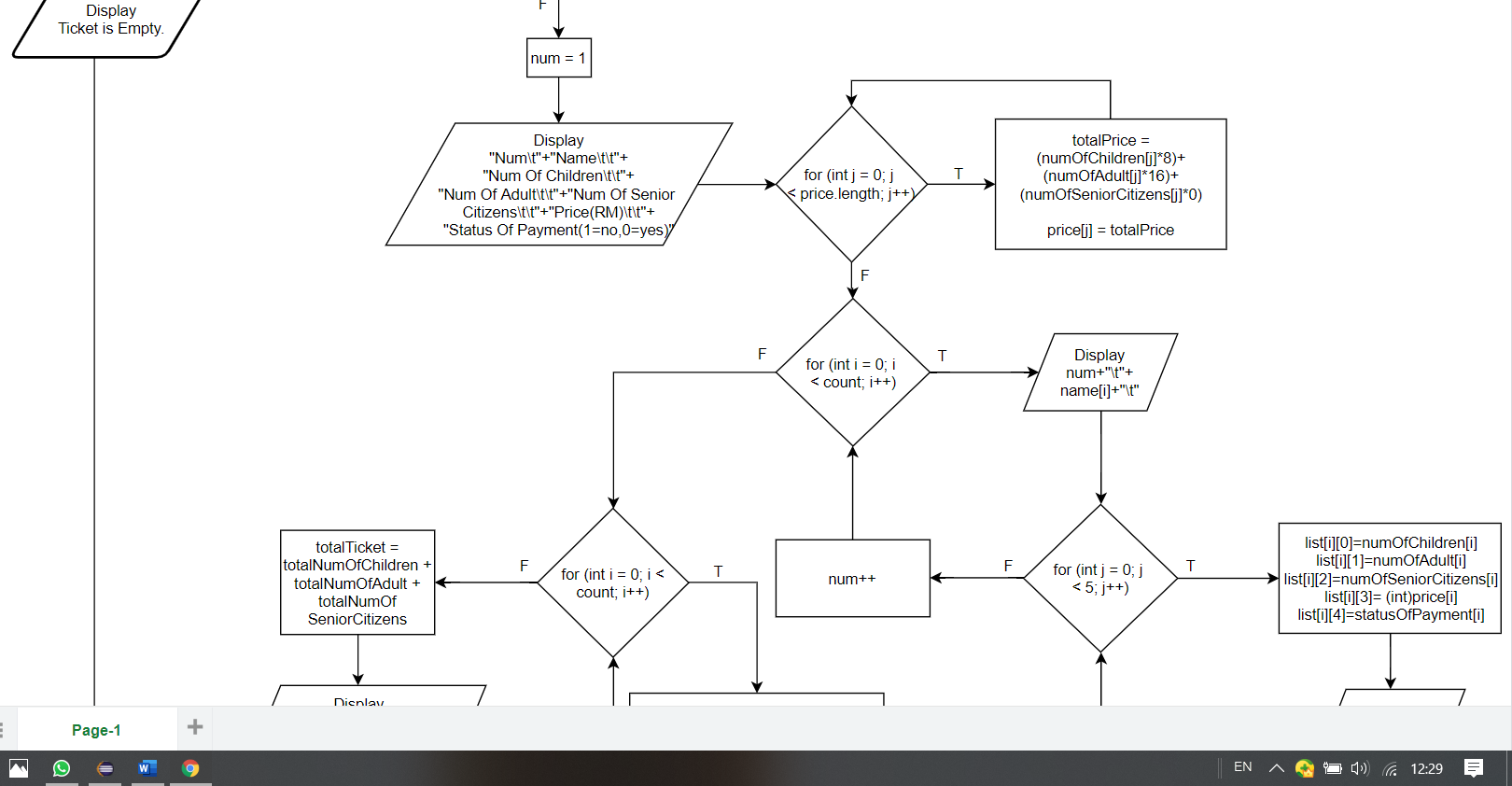
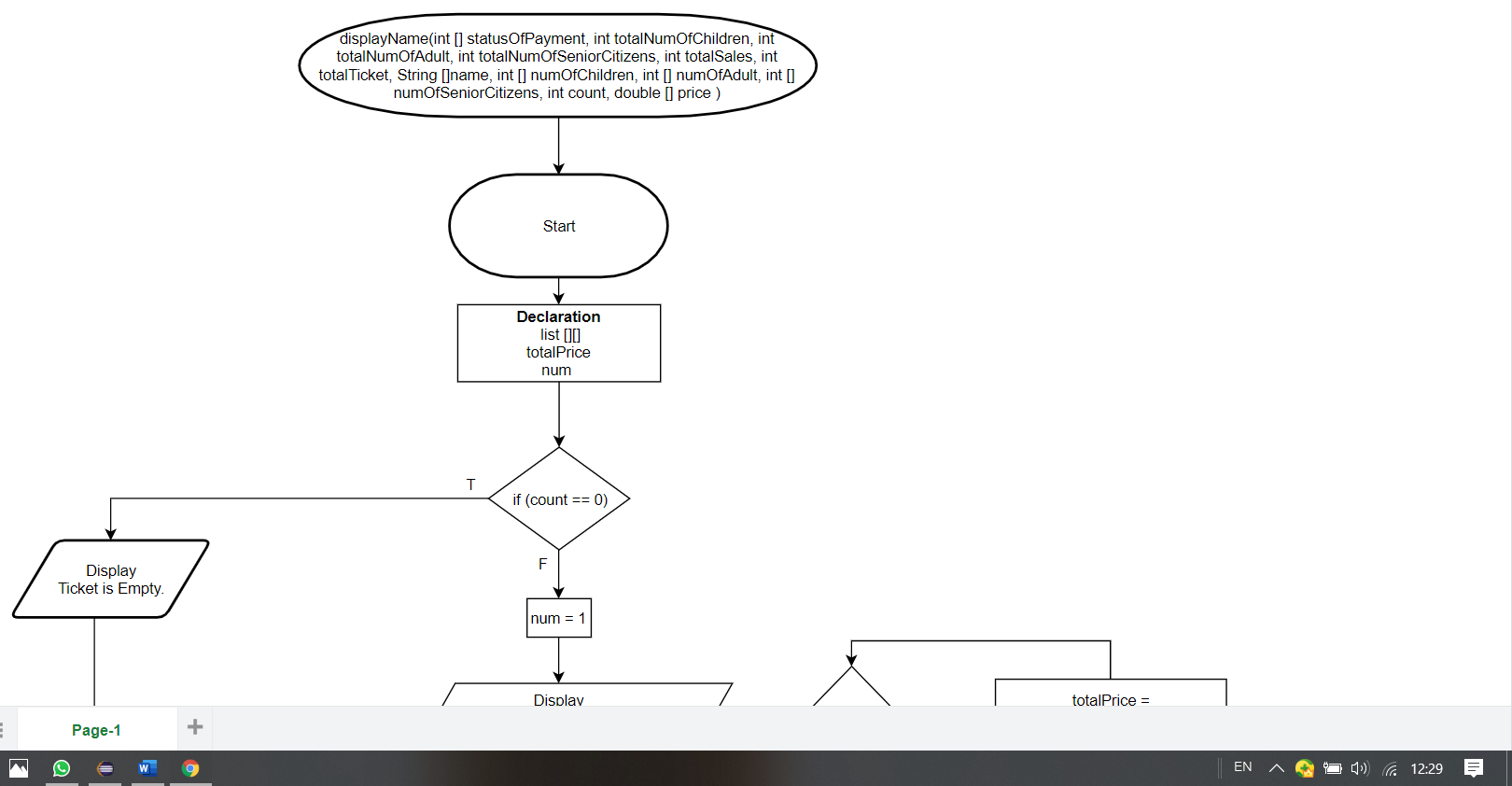


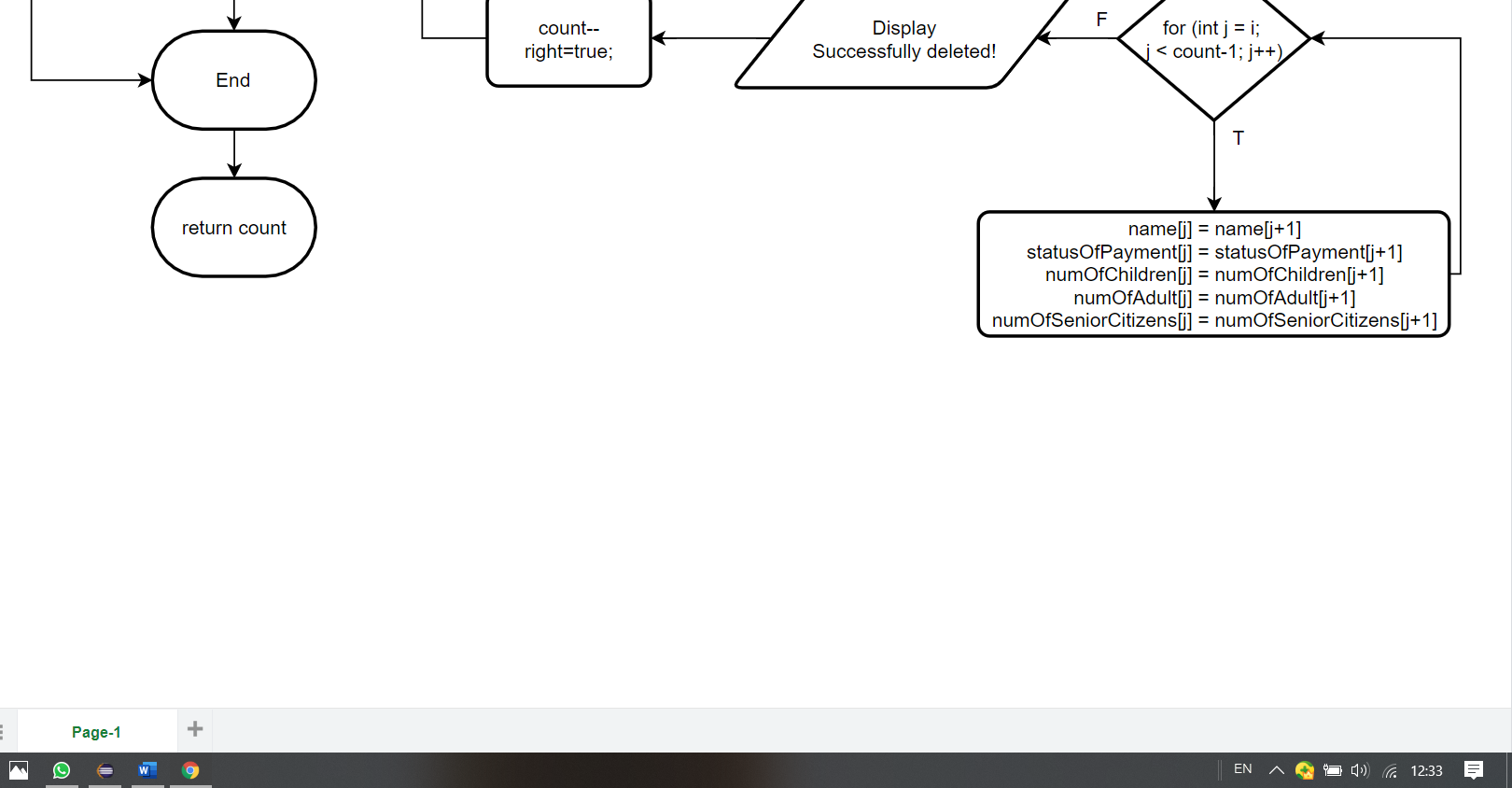
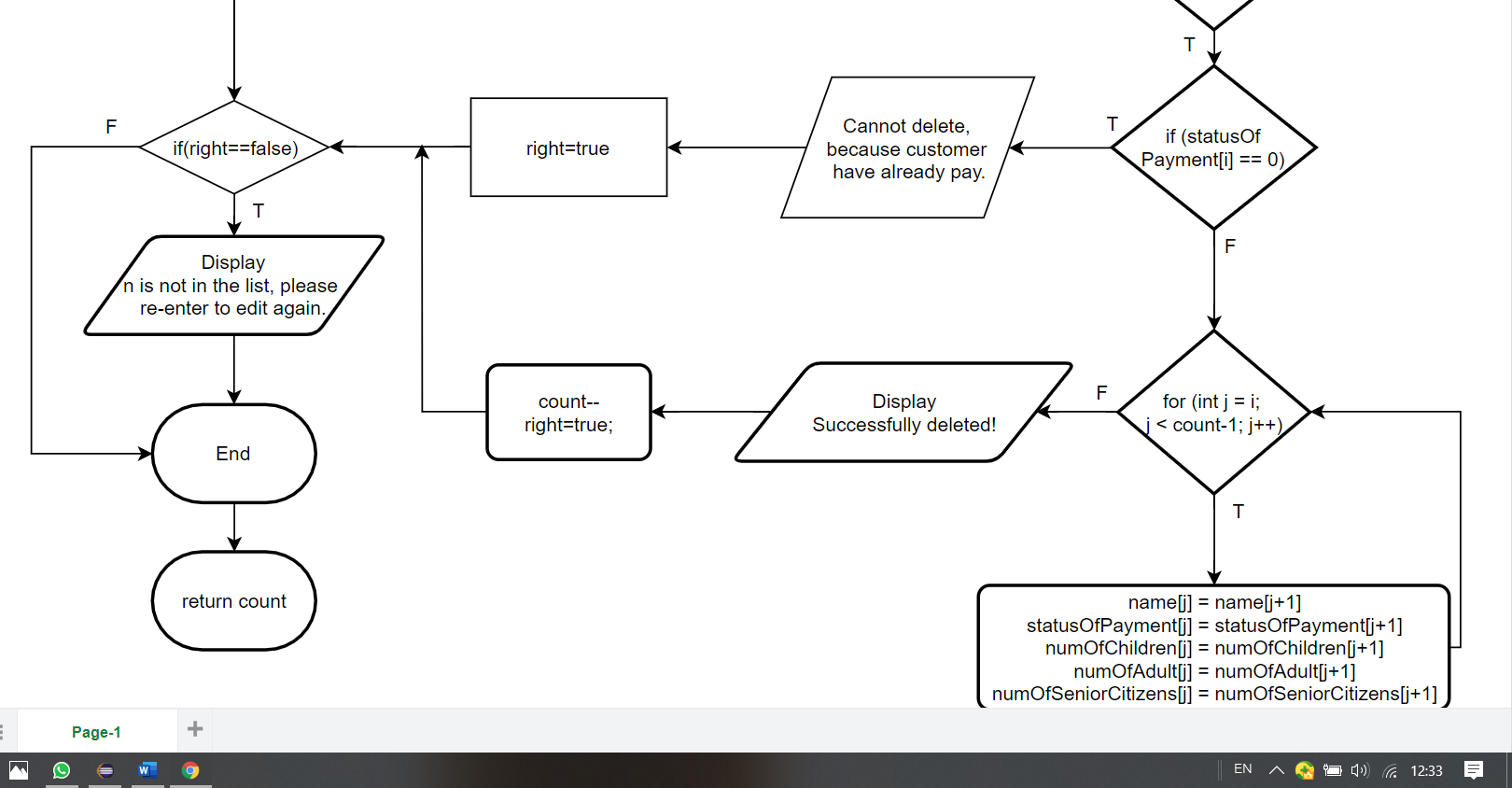
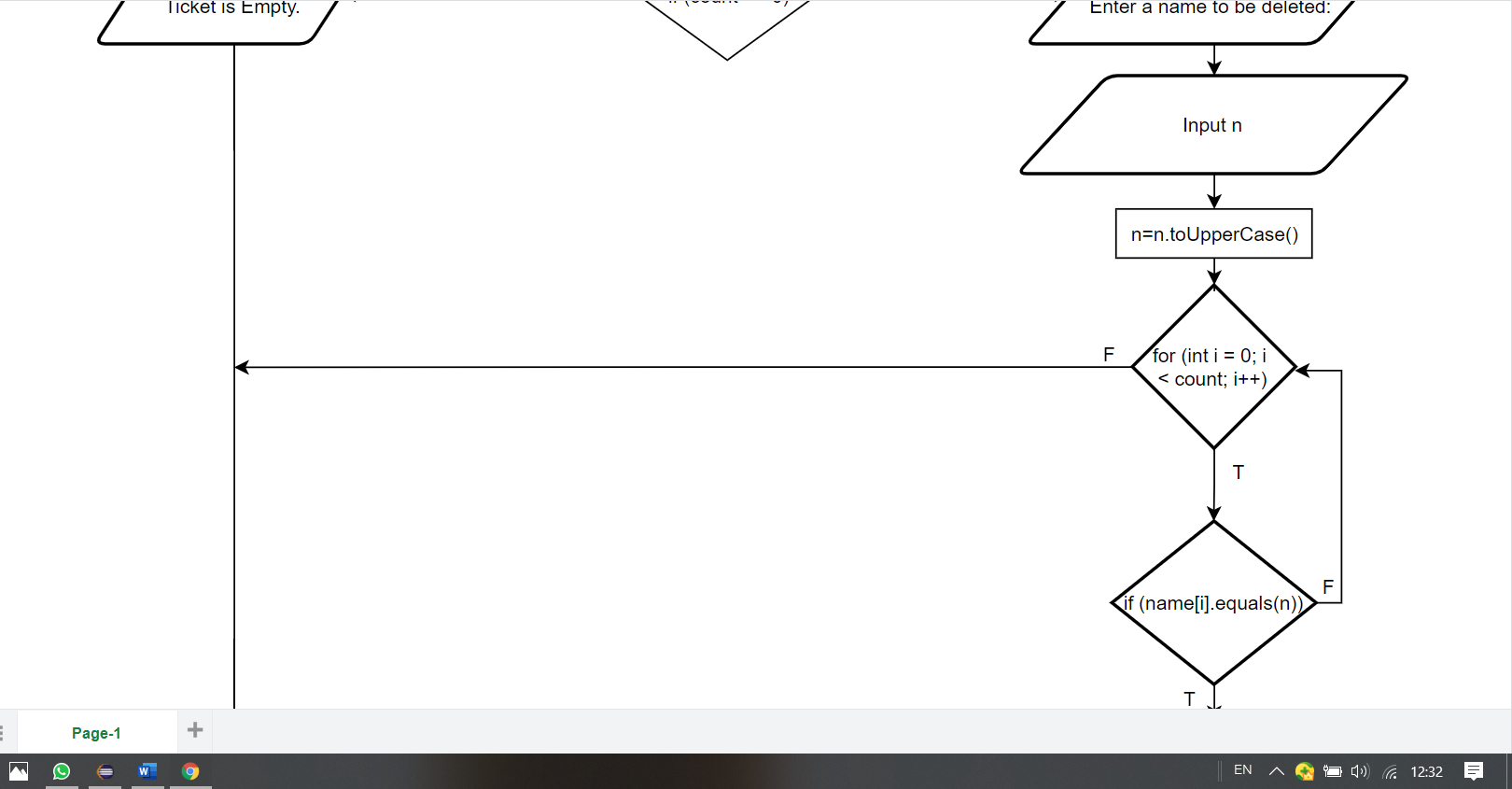
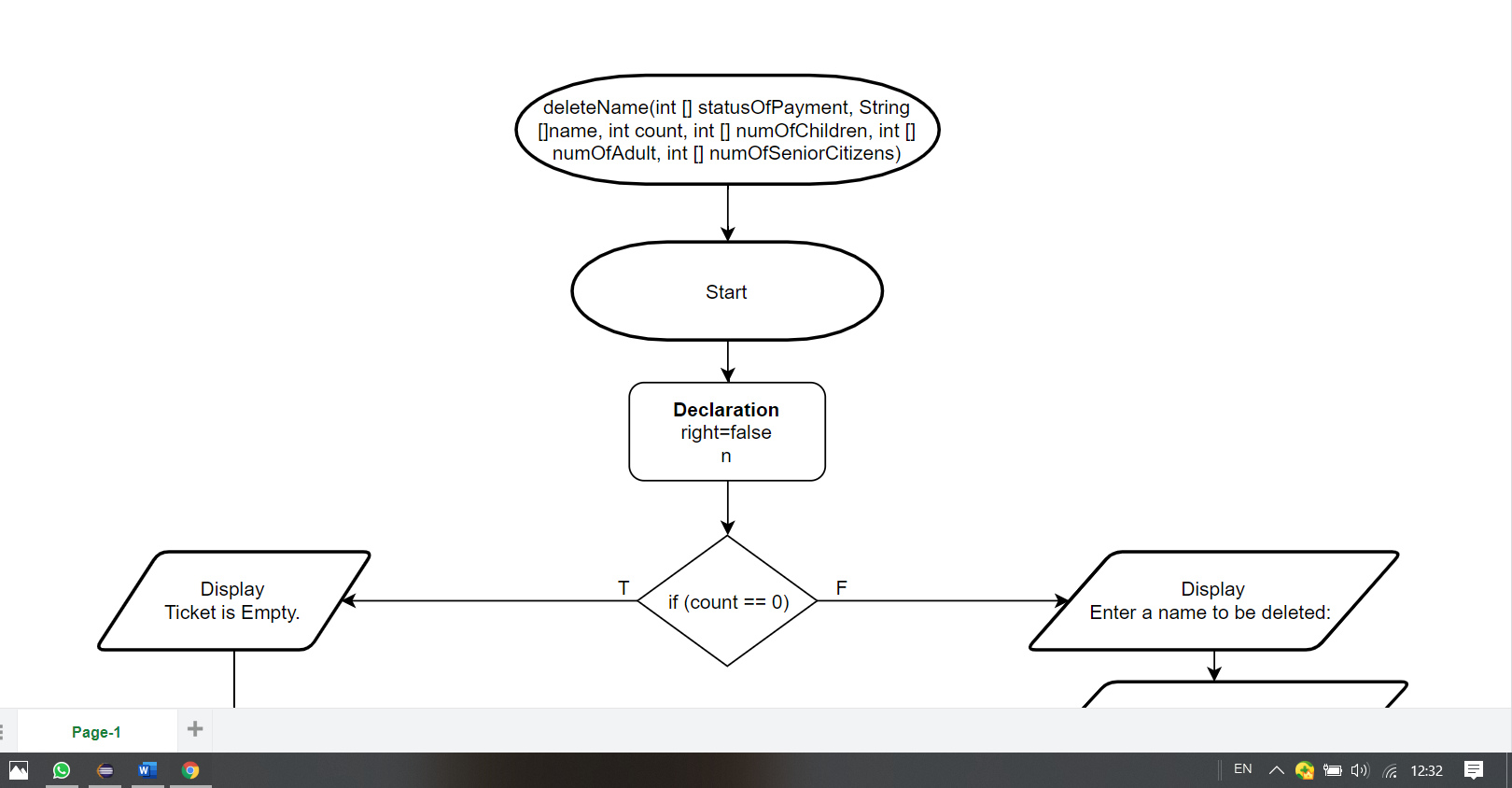












**Source Code**

import java.util.Scanner;  
  
public class FinalProject {  
 static Scanner input = new Scanner(System.in);  
 public static void main(String[] args) {  
   
 //Declare variables  
 int choice;  
 int count = 0;  
 String [] name = new String [50];  
 int [] numOfChildren = new int [50];  
 int [] numOfAdult = new int [50];  
 int [] numOfSeniorCitizens = new int [50];  
 double [] price = new double [50];  
 int [] statusOfPayment = new int [50];  
 int totalNumOfChildren=0;  
 int totalNumOfAdult=0;  
 int totalNumOfSeniorCitizens=0;  
 int totalTicket=0;  
 int totalSales=0;  
   
 do {  
   
 menu(); //call menu  
 choice = input.nextInt();  
 input.nextLine();//free buffer  
   
 switch(choice) {  
   
 case 0: System.exit(0);//exit  
 case 1: count = addName(statusOfPayment, name, count, numOfChildren, numOfAdult, numOfSeniorCitizens, price);//call addNameAndAge  
 break;  
 case 2: editName( statusOfPayment, name, count, numOfChildren, numOfAdult, numOfSeniorCitizens);//call editName  
 break;  
 case 3: searchName(statusOfPayment, name, count, numOfChildren, numOfAdult, numOfSeniorCitizens);//call searchName  
 break;  
 case 4: displayName( statusOfPayment, totalNumOfChildren, totalNumOfAdult, totalNumOfSeniorCitizens, totalSales, totalTicket, name, numOfChildren, numOfAdult, numOfSeniorCitizens, count, price);//call displayName  
 break;  
 case 5: count = deleteName(statusOfPayment, name, count, numOfChildren, numOfAdult, numOfSeniorCitizens);//call deleteName  
 break;  
 default: System.out.println("Error!!Invalid");  
 }//end switch  
   
  
 System.out.println("\n");  
 }while(true);  
}  
 // create method menu  
 public static void menu() {  
   
 System.out.println("\*\*\*\*ZOO TAIPING\*\*\*\*\nAwesome Wildlife Encounters");  
 System.out.println(":::::::: MENU ::::::::"  
 +"\n1. Add Name"  
 +"\n2. Edit information"  
 +"\n3. Search and pay"  
 +"\n4. Display All"  
 +"\n5. Delete Name"  
 +"\n0. EXIT");  
 System.out.println("Enter your choice (1-5) or 0 to Exit ");  
 System.out.print("Please Select : ");  
  
 }//end menu()  
   
   
   
 // create method addName  
 public static int addName(int [] statusOfPayment, String []name, int count, int [] numOfChildren, int [] numOfAdult, int [] numOfSeniorCitizens, double [] price) {  
 if (count == name.length) {//array is full  
 System.out.println("List of ticket is full.");  
 }  
 else {//array not full  
 System.out.print("Enter Name \t\t: ");  
 name[count] = input.nextLine();  
 name[count]=name[count].toUpperCase();  
 System.out.print("Enter Num of Children \t: ");  
 numOfChildren [count] = input.nextInt();  
 System.out.print("Enter Num of Adult \t: ");  
 numOfAdult [count] = input.nextInt();  
 System.out.print("Enter Num of Senior Citizens \t: ");  
 numOfSeniorCitizens [count] = input.nextInt();  
 statusOfPayment[count] = 1;  
  
 count ++;  
   
 }  
 return count;  
 }//end addName()   
   
   
   
 // create method editName  
 public static void editName(int [] statusOfPayment, String []name, int count, int [] numOfChildren, int [] numOfAdult, int [] numOfSeniorCitizens) {  
 int flag=0;  
 if (count == 0) {//array is empty  
 System.out.println("List of ticket is Empty. Please add a name first.");  
 }  
 else {//array not empty  
 System.out.print("Enter the name want to edit: ");  
 String n = input.nextLine();  
 n=n.toUpperCase();  
 for (int i = 0; i < count; i++) {  
 if (name[i].equals(n)) {  
 if (statusOfPayment[i] == 0) {  
 System.out.println("Cannot edit, because customer have already pay.");  
 flag=1;  
 }  
 else {  
 System.out.print("Edit the name " + name[i] +" to : ");  
 name[i] = input.nextLine();  
 name[i]=name[i].toUpperCase();  
 System.out.println();  
 System.out.println("Want to edit the number of Children, Adult and Senior Citizens");  
 System.out.print("Y / N :");  
 String yN = input.next();  
 System.out.println();  
 do {  
 if ((yN.equals("y"))||(yN.equals("Y"))) {  
 System.out.print("Edit the number of Children from " + numOfChildren[i] +" to : ");  
 numOfChildren[i] = input.nextInt();  
 System.out.print("Edit the number of Adult from " + numOfAdult[i] +" to : ");  
 numOfAdult[i] = input.nextInt();  
 System.out.print("Edit the number of Senior Citizens from " + numOfSeniorCitizens[i] +" to : ");  
 numOfSeniorCitizens[i] = input.nextInt();  
 break;  
 }  
 else if ((yN.equals("n"))||(yN.equals("N"))){  
 break;  
 }  
 else {  
 System.out.println("Invalid input, please try again");  
 System.out.println("Want to edit the number of Children, Adult and Senior Citizens");  
 System.out.print("Y / N :");  
 yN = input.next();  
 }  
 } while (true);  
   
 System.out.println("Successfully edit!");  
 flag=1;  
   
   
 }  
 }  
 }  
 if(flag==0)  
 System.out.println(n+" is not in the list, please re-enter to edit again.");  
 }  
   
 }// end edit name()  
   
   
   
 // create method searchName  
 public static void searchName(int [] statusOfPayment, String []name, int count, int [] numOfChildren, int [] numOfAdult, int [] numOfSeniorCitizens) {  
 int flag=0;  
 int payment = 0;  
 final int PRICE\_OF\_CHILDREN=8;  
 final int PRICE\_OF\_ADULT=16;  
 final int PRICE\_OF\_SENIOR\_CITIZENS=0;  
   
   
 if (count == 0) {//array is empty  
 System.out.println("List of ticket is Empty. Please add a name first.");  
 }  
 else {//array not empty  
 System.out.print("Enter a search name: ");  
 String n = input.nextLine();  
 n=n.toUpperCase();  
 for (int i = 0; i < count; i++) {  
 if (name[i].equals(n)) {  
 System.out.println("Successful Search! "+ n +" is in the list of ticket.");  
 if (statusOfPayment[i] == 0) {  
 System.out.println("Already paid.");  
 flag=1;  
 }  
 else {  
 double totalPrice = (PRICE\_OF\_CHILDREN\*numOfChildren[i])+(PRICE\_OF\_ADULT\*numOfAdult[i])+(PRICE\_OF\_SENIOR\_CITIZENS\*numOfSeniorCitizens[i]);  
 System.out.println("Num of Children :"+numOfChildren[i]);  
 System.out.println("Num of Adult :"+numOfAdult[i]);  
 System.out.println("Num of Senior Citizens :"+numOfSeniorCitizens[i]);  
 System.out.printf("Total Price :RM%.2f",totalPrice);  
 System.out.println("\n");  
 System.out.println("Want to pay now");  
 System.out.println("Y /N :");  
 String yN = input.next();  
   
 do {  
 if ((yN.equals("Y"))||(yN.equals("y"))) {  
 System.out.print("\n\nCustomer Payment :RM");  
 payment = input.nextInt();  
 statusOfPayment[i] = 0;  
   
 while(true){  
 double balance = payment - totalPrice;  
 if (balance == 0) {  
 System.out.println("\nCustomer's receipt");  
 System.out.println();  
 System.out.println("==================================================");  
 System.out.println(" ZOO TAIPING ");  
 System.out.println("==================================================");  
 System.out.println("Num :"+(i+1));  
 System.out.println("Name :"+n);  
 System.out.println("Number of Children (RM8/person) :"+numOfChildren[i]);  
 System.out.println("Number of Adult (RM16/person):"+numOfAdult[i]);  
 System.out.println("Number of Senior Citizens (Free):"+numOfSeniorCitizens[i]);  
 System.out.printf("\n\nTotal Price :RM%.2f",totalPrice);  
 System.out.println("\n\*\*\*\*Thank you, Please come again\*\*\*\*");  
 break;  
 }  
 else if (balance > 0) {  
 System.out.println("\nCustomer's receipt");  
 System.out.println();  
 System.out.println("==================================================");  
 System.out.println(" ZOO TAIPING ");  
 System.out.println("==================================================");  
 System.out.println("Num :"+(i+1));  
 System.out.println("Name :"+n);  
 System.out.println("Number of Children (RM8/person) :"+numOfChildren[i]);  
 System.out.println("Number of Adult (RM16/person):"+numOfAdult[i]);  
 System.out.println("Number of Senior Citizens (Free):"+numOfSeniorCitizens[i]);  
 System.out.printf("\n\nTotal Price :RM%.2f",totalPrice);  
 System.out.printf("\nThis your balance :RM%.2f",balance);  
 System.out.println("\n===========Thank you, Please come again===========");  
 break;  
 }  
 else {  
 System.out.println("Sorry, this is not enough,you need to pay more");  
 System.out.print("Extra Payment:");  
 payment += input.nextInt();  
 }  
 }//end for  
 break;  
 }  
 else if ((yN.equals("N"))||(yN.equals("n"))) {  
 break;  
 }  
 else {  
 System.out.println("Invalid input, please try again");  
 System.out.println("Want to pay now");  
 System.out.print("Y / N :");  
 yN = input.next();  
 }  
 } while (true);  
 } flag=1;  
 }   
 }  
 if(flag==0)  
 System.out.println(n+" is not in the list, please re-enter to search again.");   
 }  
 }// end searchName()  
   
   
   
 // create method displayName  
 public static void displayName(int [] statusOfPayment, int totalNumOfChildren, int totalNumOfAdult, int totalNumOfSeniorCitizens, int totalSales, int totalTicket, String []name, int [] numOfChildren, int [] numOfAdult, int [] numOfSeniorCitizens, int count, double [] price ) {   
 int [][]list = new int [50][5];  
   
 if (count == 0) { //array is empty  
 System.out.print("List of ticket is Empty. Please add a name first.");  
 }  
 else {  
 int num = 1;  
 System.out.println("Num\t"+"Name\t\t\t"+"Num Of Children\t\t"+"Num Of Adult\t\t"+"Num Of Senior Citizens\t\t"+"Price(RM)\t\t"+"Status Of Payment(1=no,0=yes)");  
   
 for (int j = 0; j < price.length; j++) {  
 int totalPrice = (numOfChildren[j]\*8)+(numOfAdult[j]\*16)+(numOfSeniorCitizens[j]\*0);  
 price[j] = totalPrice;  
 }  
 for (int i = 0; i < count; i++) {  
 System.out.print(num+"\t"+name[i]+"\t\t\t");  
   
 for (int j = 0; j < 5; j++) {  
 list[i][0]=numOfChildren[i];  
 list[i][1]=numOfAdult[i];  
 list[i][2]=numOfSeniorCitizens[i];  
 list[i][3]= (int)price[i];  
 list[i][4]=statusOfPayment[i];  
 System.out.print(list[i][j]+"\t\t\t\t\t\t\t");  
 }  
 System.out.println();  
 num++;  
 }  
   
 for (int i = 0; i < count; i++) {  
 totalNumOfChildren += numOfChildren[i];  
 totalNumOfAdult += numOfAdult[i];  
 totalNumOfSeniorCitizens += numOfSeniorCitizens[i];  
 totalSales += price[i];  
   
 }  
   
 System.out.println("\n");  
 totalTicket = totalNumOfChildren + totalNumOfAdult + totalNumOfSeniorCitizens;  
 System.out.println("Total Number of Children is "+totalNumOfChildren+" ("+(totalNumOfChildren\*8)+")");  
 System.out.println("Total Number of Adult is "+totalNumOfAdult+" ("+(totalNumOfAdult\*16)+")");   
 System.out.println("Total Number of Senior Citizens is "+totalNumOfSeniorCitizens+" (0)");  
 System.out.println("Total sales of ticket is "+ totalTicket);  
 System.out.printf("Total sales is RM%.2f",(double)totalSales);  
 System.out.println();  
 }  
 }//end displayName()  
   
  
   
 // create method deleteName  
 public static int deleteName(int [] statusOfPayment, String []name, int count, int [] numOfChildren, int [] numOfAdult, int [] numOfSeniorCitizens) {  
 boolean right=false;  
 if (count == 0) {//array is empty  
 System.out.print("List of ticket is Empty. Please add a name first.");  
 }  
 else {//array not empty   
 System.out.print("Enter a name to be deleted: " );  
 String n = input.nextLine();  
 n=n.toUpperCase();  
 for (int i = 0; i < count; i++) {  
 if (name[i].equals(n)) {  
 if (statusOfPayment[i] == 0) {  
 System.out.println("Cannot be deleted, customer have already paid.");  
 right=true;  
 }  
 else {  
 for (int j = i; j < count-1; j++) {  
 name[j] = name[j+1];  
 statusOfPayment[j] = statusOfPayment[j+1];  
 numOfChildren[j] = numOfChildren[j+1];  
 numOfAdult[j] = numOfAdult[j+1];  
 numOfSeniorCitizens[j] = numOfSeniorCitizens[j+1];  
 }  
 System.out.println("Successfully deleted!");  
 count--;  
 right=true;  
   
 }  
 }  
 }  
 if(right==false)  
 System.out.println(n+" is not in the list, please re-enter to delete.");   
 }  
   
 return count;  
 }//end deleteName()  
   
}