

## **Tooth Care system**

Student registration number Esoft: E212527

KDU registration number: K2377609

Name: P.Sumudu Dileepa Dayarathna

Date: 12/25/2023

Contact: 0715446575 /0777319825

Driver link -

[https://drive.google.com/drive/folders/1riDNGCS\\_4nQ9nCxaGkQge9OKqVJLLxJq?usp=drive\\_link](https://drive.google.com/drive/folders/1riDNGCS_4nQ9nCxaGkQge9OKqVJLLxJq?usp=drive_link)

GitHub link

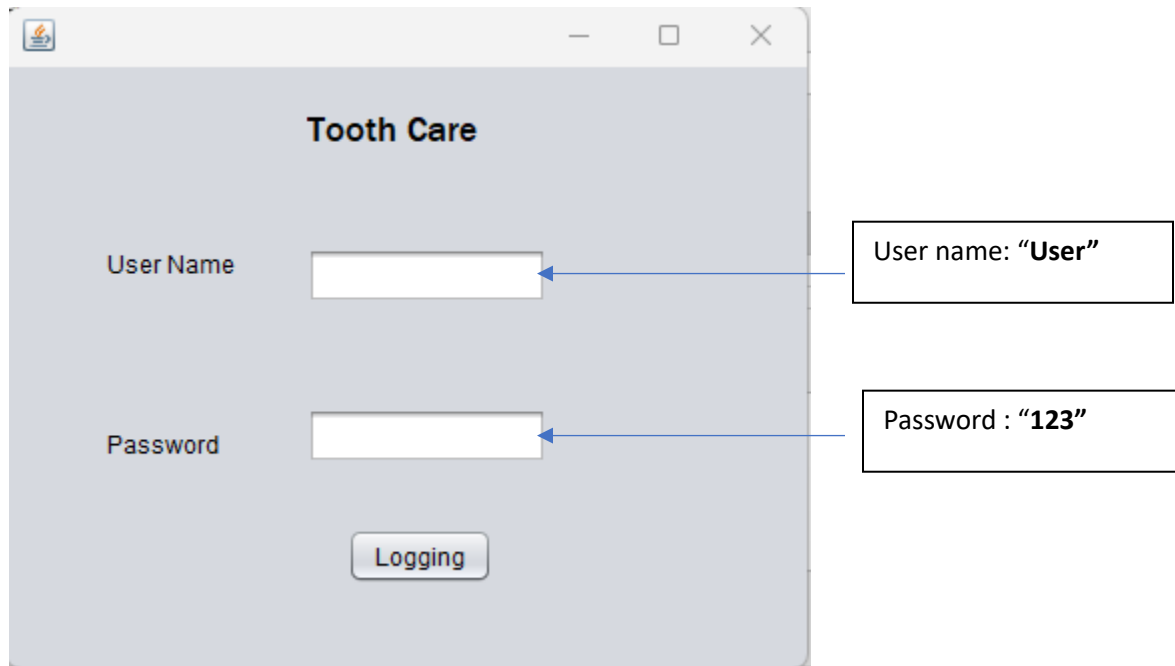
<https://github.com/Sumududd/Programming-III-Coursework--1>

## **Index**

- 1.Interfaces
- 2.Operating procedure
- 3.Code overview
- 4.Class diagram

## 1. Interfaces

### Logging Interface



The screenshot shows a window titled "Tooth Care" with a light gray background. Inside the window, there are two labels: "User Name" and "Password". Next to each label is a white text input field. Below these fields is a button labeled "Logging". To the right of the window, there are two callout boxes. The first box, labeled "User name: 'User'", has a blue arrow pointing to the "User Name" input field. The second box, labeled "Password : '123'", has a blue arrow pointing to the "Password" input field.

User Name

Password

Logging

User name: "User"

Password : "123"

Enter the default user name "– "User" and password –"123" to the logging the system

## Main Interface

The screenshot shows the 'Tooth Care' application window. It is divided into several sections: 'Enter Date or Apoiment ID', 'Search', 'Treatment', and 'Payment'. Callouts provide detailed explanations of the interface elements and their functions.

**Enter Date or Apoiment ID:**

- Date selection:** A dropdown menu showing 'Monday'.
- View the available schedule by date:** A button labeled 'View timeslot'.
- Confirmation of acceptance of appointment fee:** A checkbox labeled 'Advance payment LKR' with a value of '1000.0' and a checked box.
- Create an appointment:** A button labeled 'Make Apoiment'.
- Auto generate appointment number:** A text field showing 'Apoiment ID' with the value '4'.

**Search:**

- Filter appointment data by date:** A dropdown menu labeled 'Select Date'.
- \*Clear the appointment ID before search:** A text field labeled 'Enter Apoiment ID' with the value '2'.
- Filter appointment date by appointment ID:** A text area displaying search results: 'ApID: 2 Date:Monday Time: 7.00PM Name:Ishara Tel:071885425 Address:Athrugiriya'.
- Filtered appointment details:** A button labeled 'Search'.

**Treatment:**

- Select the treatment:** A dropdown menu labeled 'Select Treatment' with the value 'CLEANINGS'.

**Payment:**

- Get the total treatment cost:** A text field labeled 'Total bill' with the value '6000.0'.
- Generate the bill:** A button labeled 'Print Bill'.
- View the bill:** A text area displaying bill details: 'Apoiment ID :2', 'Treatmentcost LKR : 6000.0', 'Advance payment LKR : 1000.0', and 'Total amount LKR : 5000.0'.

## 2. Operating Procedure

### Make an appointment

1. Select date
2. Select generated time according selected date.
3. Enter name, Address, Telephone no, and tick the payment and make the appointment.

### View appointment

1. Select date and search appointments **or** enter appointment id and search the patient data. (When entered appointment id priorities goes to appointment id and until clean the field no view the appointment details of the relevant date.

### Add or remove treatment

1. Select treatments and add or remove to the list

### Payment

Press the “Total bill” button and view the cost only for the treatment.

Press the print bill button and view the bill details and advance payments.

## 3. Code overview

### Logging

```

22  /**
23   * This method is called from within the constructor to initialize the form.
24   * WARNING: Do NOT modify this code. The content of this method is always
25   * regenerated by the Form Editor.
26   */
27  @SuppressWarnings("unchecked")
28  Generated Code
29
75
76  private void txtUsernameActionPerformed(java.awt.event.ActionEvent evt) {
77      // TODO add your handling code here:
78  }
79
80  private void btnLoggingActionPerformed(java.awt.event.ActionEvent evt) {
81      String uname=txtUsername.getText();//User
82      String pword=txtPassword.getText();//123
83      if(uname.equals(new Object("User")) && pword.equals(new Object("123")))
84      {
85          Mainframe m1=new Mainframe();
86          m1.setVisible(true);
87          this.dispose();
88      } else {
89          JOptionPane.showMessageDialog(parentComponent,null,message:"User name or password incorrect");
90      }
91  }
92
93
94  }
95
96  /**
97

```

## Array list ,Arrays ,Linked list and major variable declarations

Code overview Array List declaration to store patient data, LinkedList declaration to store selected treatment index, Array declaration to store details of treatments from treatment class and two-dimensional array declaration to store available time range for each date.

```

13
14
15 int apointmentID=0 ; // Variable declaration for apointment id selection
16 String selecteddate; //Variable declaration for date assignn selected date
17 int selectdateindex; //Variable for date selection combo box index
18 int apointmentcount=1; // /Variable declaration for apointment ID
19 String apointmentIDforBill; //Variable declaration for get current apointment id for bill caculation
20 double tretmentcost;// Variable declararion for total tratment cost
21 double advacepayment;//Variable define for assignn the advance payment
22 double totalamount; // Variable define assignn total tratment cost
23 double sendtretmentcost; // Variable difine for assignn selected each treatment cost
24 private boolean butTotalcostAction =false; //Variable assignn for control tottal cost calculation button repeate control
25
26 ArrayList<Patient>patientdataArray=new ArrayList<>();// intiatate the array list to store patient class object
27 LinkedList<Integer>tretmentindex=new LinkedList(); // intiate linked list for save selected treatment index
28 Treatment[] treatmentarray = Treatment.values(); // create array to store all the treatment detaels retrieve from the Treatment class
29 Payment billical=new Payment();// create object for Payment class
30
31 int timearray[][]={{6,9},{6,9},{3,10},{3,10}}; //Define an array for store time range accoding to the date
32
33 public Mainframe() {
34
35     initComponents();
36     ///////////////////////////////////////////////////////////////////Initiation of the interface details
37     for(Treatment treatmentitem : Treatment.values()){ //Reasding treatment class treatment detaails
38         String treatment=treatmentitem.toString(); //convert Constant to the string
39         comboBoxTreatment.addItem(item:treatment); // Put item to treatment selection combo box
40         advancePayment.setText(s: Double.toString(d: billical.getadvancepayment())); //read advancepayment values and print in tex box
41     }
42 }

```

newpackage.Mainframe >

## Date and time selection

Array definition to the store time rage according to the date

```

Payment billical=new Payment();// create object for Payment class

int timearray[][]={{6,9},{6,9},{3,10},{3,10}}; //Define an array for store time range accoding to

```

View time according to the selected date

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    ComboBoxTime.removeAllItems();
    int index=comboBoxDate.getSelectedIndex(); // get selected date index value
    int val=timearray[index][0]; // Assing to the variable;
    int y=timearray[index][1]; // get value of the array last value of the row
    for(int i= val; i<= y;i++){ // run for loop to get find available time range

        String stringdate=String.valueOf(i);
        ComboBoxTime.addItem(stringdate+".00PM"); // Set to combo box time range relavent to the each date
        //System.out.println("Value of i is"+i);
    }
}
```

### Collect patient data and store it in an array list

Collecting customer data and create objects in patent class and save patient objects in an array list. An increment was assigned to the appointment I

```
private void jButtonMakeApoimentActionPerformed(java.awt.event.ActionEvent evt) {
    txtboxApoimnentID.setText(Integer.toString(++appointmentcount)); // appointmentID convert to string and set txtboxApoim
    if(true==checkboxAppointment.isSelected()){ // Check whether advance payment accept or not
        String name=txtboxPatientname.getText(); //read patient name and assignn to variable
        String telno=txtboxTelno.getText(); //read patient telephone number and assignn to variable
        String address=txtboxAddress.getText(); // read patient address and assignn to variable
        String time=ComboBoxTime.getSelectedItem().toString(); // Combo box item convert to string
        String date = comboBoxDate.getSelectedItem().toString(); // Combo box item convert to string

        Patient patientdatalist=new Patient( appointmentid: appointmentcount,name,telno,address,time,date); // Create object "patien
        patientdataArray.add(patientdatalist); // adding object to the array
        appointmentcount++; // appointmentcount increment by one

        txtboxPatientname.setText(""); //Clear name
        txtboxTelno.setText(""); // Clear telephone numer
        txtboxAddress.setText(""); //Clear address

        JOptionPane.showMessageDialog(parentComponent:null,message:"The appointment was successfully saved");
    }
    else { //If not advance payment accept generate a message
    }
}
```

newpackage.Mainframe > jButtonMakeApoimentActionPerformed > if (true == checkboxAppointment.isSelected())

## Patient class

```
1 package newpackage;
2
3 public class Patient{
4
5     private int apoimentid;
6     private String name;
7     private String telno;
8     private String address;
9     private String time;
10    private String date;
11
12    public Patient (int apoimentid,String name,String telno, String address,String time,String date){
13
14        this.apoimentid =apoimentid;
15        this.name=name;
16        this.telno=telno;
17        this.address=address;
18        this.time=time;
19        this.date=date;
20
21    }
22
23    public int getApoimnetID() {
24        return apoimentid;
25    }
26
27    public String getName() {
28        return name;
29    }
30    public String getTelno() {
31        return telno;
```

```
22
23    public int getApoimnetID() {
24        return apoimentid;
25    }
26
27    public String getName() {
28        return name;
29    }
30    public String getTelno() {
31        return telno;
32    }
33    public String getAddress() {
34        return address;
35    }
36    public String getTime() {
37        return time;
38    }
39    public String getDate() {
40        return date;
41    }
42
43
44
45 }
46
47 //Store patent class object in Patientdatalist array lis
48
49
50
51
52
```

## Search appointment data by name or search appointment data by appointment number.

Retrieving the object from the object array list and plot the data according to search basis on date or id number.

```

401 private void butSearchActionPerformed(java.awt.event.ActionEvent evt) {
402     billocal.clearvariable();//clear variable of payment class
403     texareaTreatmentList.setText(""); //cleat txt area
404     txtareaSearch.setText(""); //Clear the te
405     tretmentindex.clear(); // Clearing the selected treatment list
406     String ID=txtboxEnterID.getText(); //Reeading get from apoiment ID entering text feild
407     apoimentIDforBill=ID;
408
409     if (!ID.isEmpty()){ //Chacking the text feald is emty or not
410         apoimentID=Integer.parseInt(ID); // String value converter to Integer store in variable.
411     }
412
413     if(0<selectdateindex&&ID.isEmpty()){ //check whether the date has selected and did entered Apoiment ID
414
415         ////selec base on date
416         for (Patient patient : patientdataArray) { //retrieving object from patientdataArray arraylink
417             if(patient.getDate().equals(anObject.selecteddate)) // checking the dates equals to the selected date
418
419                 txtareaSearch.append("ApID: " + patient.getApoimnetID()+" "+"Date:" +patient.getDate()+" "+"Time: " + patient.getTime()+" "+"Name:" +patient
420
421             }
422         // clear text area
423     } else
424     {
425         //////////////////////////////////////
426         butTotalcostAction =true ; //activate the totalcost calculation
427
428     }
429
430

```

newpackage.Mainframe > @ butSearchActionPerformed >

Output toothcaresystem (run) #2 running... × (1 more...) 413:13 INS U

## Recording treatment details

User selected treatment details index adding to an array and retrieve treatments details from the array list accordingly to the bill calculation.



```

- }
3 private void addTreatmentActionPerformed(java.awt.event.ActionEvent evt) {
    butTotalCostAction = true;

    int i = comboBoxTreatment.getSelectedIndex(); //Treatment combo box index value assign to the variable
    treatmentindex.add(e: i); // add selected treatment to list

    Treatment treatmentfee = treatmentarray[i]; //retrieve treatment detail according to the selected treatment
    var treatmentname = treatmentfee.toString(); //array component convert to string
    texareaTreatmentlist.append(treatmentname + "\n"); // selected treatment type print text area
- }
3 private void comboBoxTreatmentActionPerformed(java.awt.event.ActionEvent evt) {

```

### Deleting treatment details from the list

User selected treatment details index removing from an array and retrieve rest of treatments details for bill calculation.

```

481 private void jButtonRemoveTreatmentActionPerformed(java.awt.event.ActionEvent evt) {
483     butTotalCostAction = true;
484
485     texareaTreatmentlist.setText(""); //Clear treatment list
486     if (!treatmentindex.isEmpty()) { // Check whether selected treatment index array is empty or not
487         treatmentindex.removeLast(); // remove last entered treatment index
488     }
489
490     for(Integer val: treatmentindex) //retrieve rest of index number
491     {
492         Treatment treatmentfee = treatmentarray[val]; //retrieve treatment detail according to the selected treatment
493         var treatmentname = treatmentfee.toString(); //array component convert to string
494         texareaTreatmentlist.append(treatmentname + "\n"); // selected treatment type print text area
495     }
496 }
497
498
499

```

## Treatment Enumeration classes

```

Source History
4 public enum Treatment {
5
6     CLEANINGS(cost: 6000.00),
7     WHITENING (cost: 8500.00),
8     FILLING (cost: 15000.00),
9     NERVE_FILING (cost: 2500.00),
10    ROOT_CANAL_THERAPY (cost: 30000.00);
11
12    static void values(int i) {
13        throw new UnsupportedOperationException(message: "Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/Generate
14    }
15
16    private final double cost;
17
18
19
20
21
22    Treatment(double cost){
23
24        this.cost=cost;
25    }
26
27
28
29    public double getcost(){
30
31        return cost;
32    }
33

```

## Total treatment cost calculation

Get the index of all treatments selected from the linked list and get the treatment details from the array containing the treatment details of the treatment enumeration class. Calculating the total cost by calling methods included in the payment class.

```

503 private void butTotalcostActionPerformed(java.awt.event.ActionEvent evt) {
504
505     if(butTotalcostAction ==true){
506
507         for(Integer val:treatmentindex) //retrive rest of index numer
508         {
509             Treatment treatmentfee=treatmentarray[val]; //retrieve treatmet detail according to the selected treatment
510             sendtretmentcost =treatmentfee.getcost() ; //array component convert to string
511
512             System.out.println("sendtretmentcost="+sendtretmentcost);
513
514             billical.addingtreatment(sendtretmentcost);// call method to get total cost of the treatment
515
516             //////////////////////////////////////
517             System.out.println("sendtretmentcost"+sendtretmentcost );
518             //////////////////////////////////////
519         }
520
521         txtboxTotalFee.setText(s: Double.toString(d: billical.getttreatmentcost())); // total cost get and set to a text box
522         //////////////////////////////////////
523         System.out.println("gettreatment total cost"+billical.getttreatmentcost() );
524         //////////////////////////////////////
525
526
527         sendtretmentcost=0;
528         tretmentcost=billical.getttreatmentcost();
529         advacepayment=billical.getadvancepayment();
530

```

## Payment Class

```

Source History
4 public class Payment{
5
6     private final double advacepayment=1000;
7
8     double totaltratmentoost;
9
10 void addingteatment(double trtretmentfeelkr){
11     totaltratmentoost= totaltratmentoost+trtretmentfeelkr;
12 }
13
14 public double getttreatementoost(){
15     return totaltratmentoost;
16 }
17
18 public double getadvacepayment(){
19     return advacepayment;
20 }
21
22 public double gettotalamount(){
23     return totaltratmentoost=totaltratmentoost-advacepayment;
24 }
25
26 public void clearvariable(){
27     totaltratmentoost=0;
28 }
29
30 }
31
32
33
34

```

## Print bill

```

36
37     butTotalcostAction =false;    //deactivate the total
38 }
39
40 private void butPrintBillActionPerformed(java.awt.event.ActionEvent evt) {
41
42
43     txtareaTreatmentlist.setText("");
44     txtareaBillprint.setText("");
45     txtareaBillprint.append(".....TOOTH CARE .....");
46     txtareaBillprint.append("Apointment ID :"+ apointmentIDforBill+"\n");
47     txtareaBillprint.append("Treatmentcost LKR : "+ tretmentcost+"\n");
48     txtareaBillprint.append("Advance payment LKR : "+ advacepayment+"\n");
49     txtareaBillprint.append("Total amount LKR : "+totalamount+"\n");
50
51
52 }
53
54 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
55
56     txtboxPatientname.setText(""); //Clear name
57     txtboxTelno.setText(""); // Clear telephone number
58

```

## Class Diagram

