



INFORMATICS  
INSTITUTE OF  
TECHNOLOGY

# Object Oriented Programming

## Coursework Report 2023

Seniya Lenora

w1867624

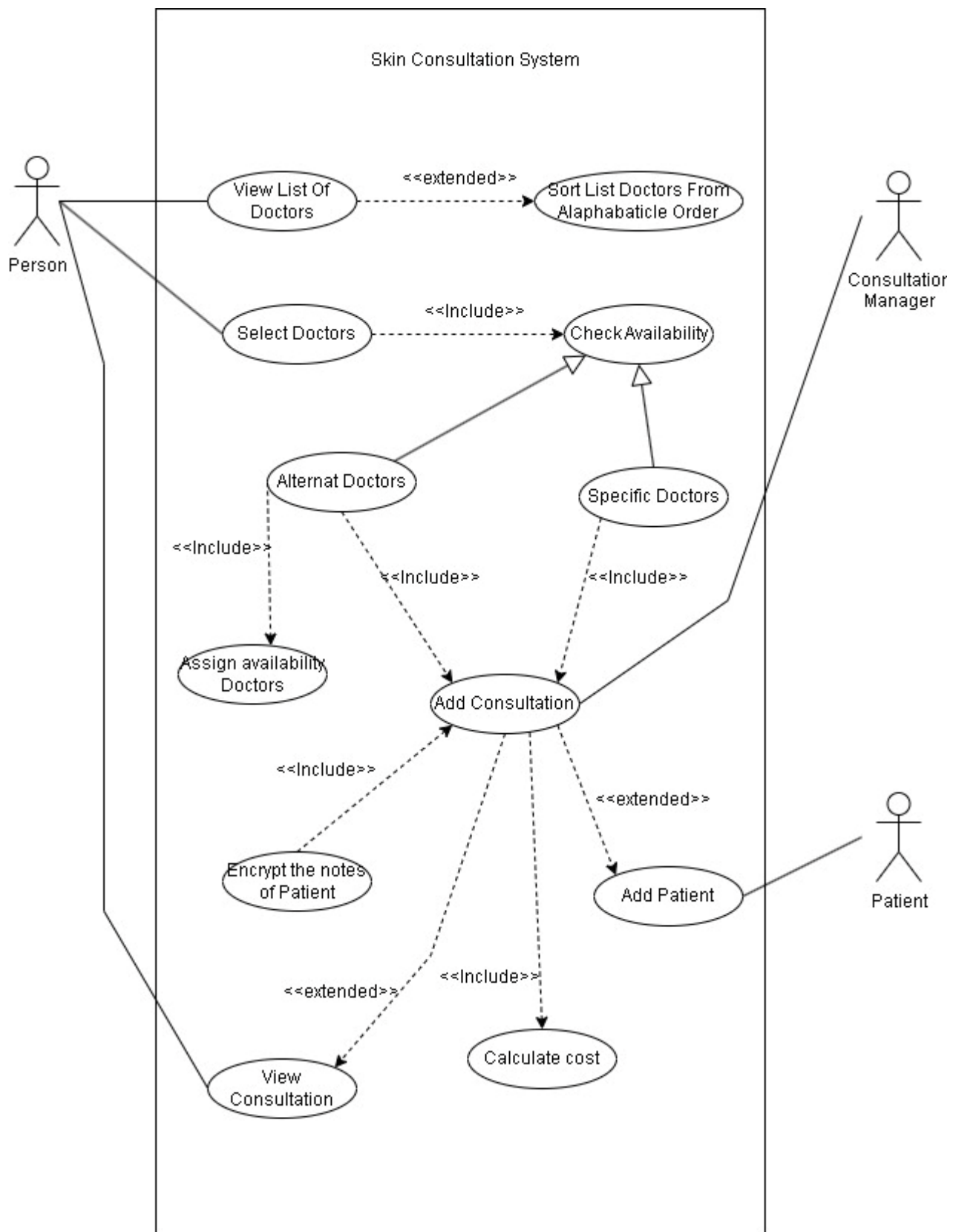
20210647

## Contents

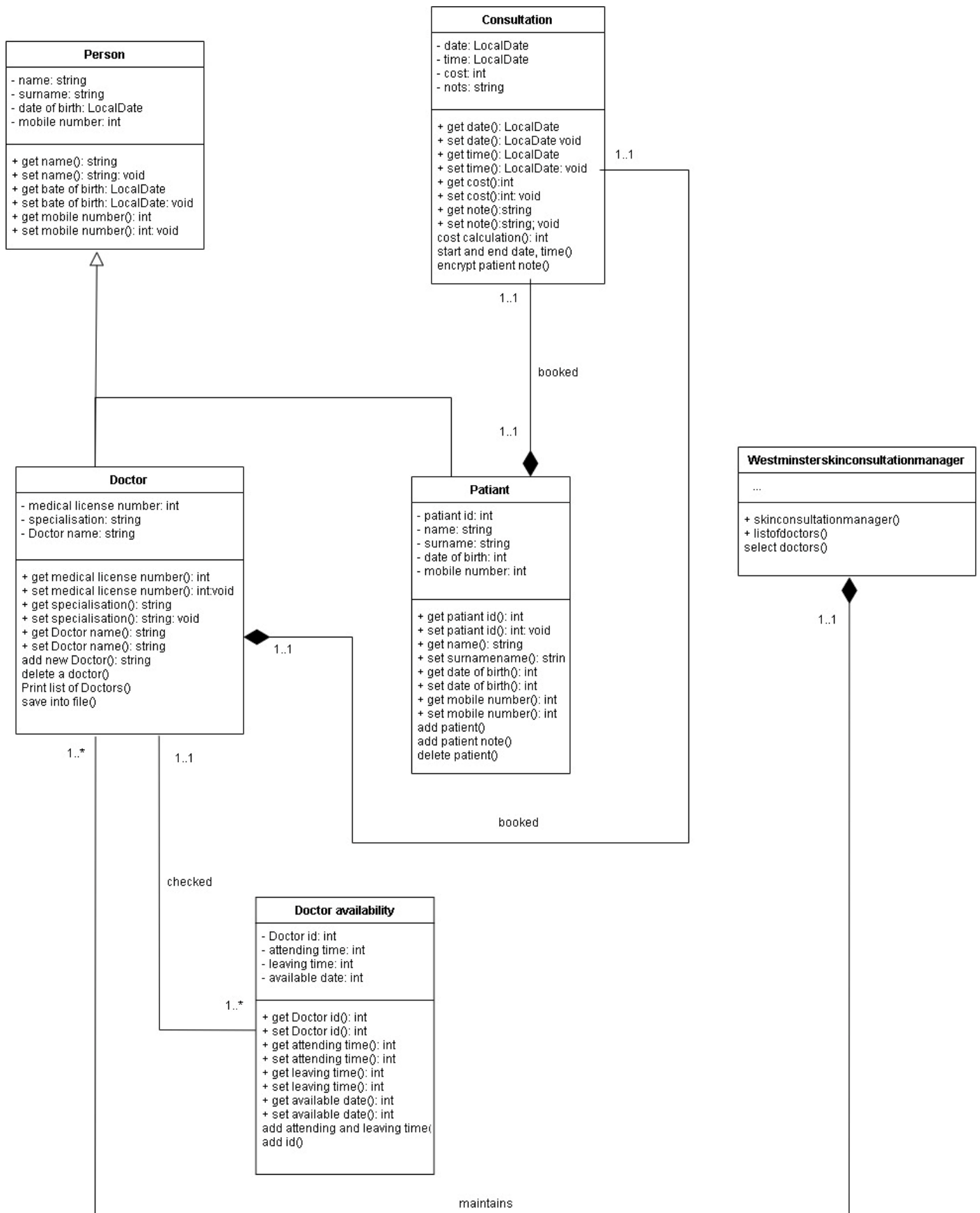
Object-Oriented Programming.....	1
Phase 01 .....	3
Use Case Diagram, .....	3
Class Diagram .....	5
Phase 02 – Course Code.....	7
Westminster Skin Consultation Manager Class, .....	7
Person, .....	14
Doctor, .....	17
Phase 03 -Source Code.....	19
Main menu, .....	19
Consultation, .....	22
Cost, .....	29
Doctor, .....	33
Doctor list, .....	38
Note encrypt, .....	41
Patient. ....	43
Persone1, .....	48
Test Plan.....	52
Reference .....	55

## **Phase 01**

**Use Case Diagram,**



## **Class Diagram**



## Phase 02 – Course Code

### Westminster Skin Consultation Manager Class,

```
package com.company;

import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Collections;
import java.util.Scanner;

import static com.company.Doctor.*;
import java.util.ArrayList;
import java.util.List;

public class WestminsterSkinConsultationManager {

    static Scanner scanner = new Scanner(System.in);
    static String reply;
    static int b = 0;
    static int i=0;
    static String lnum="";
    static String vln="";
    static String spe="";
    static int q = 0;
    static String f = "";
    static String s = "";
    static String d = "";
    static String m = "";
    static String vlnum="";
    static int TotDslots = 10;
```

```

static int TotDslots = 10;
static List<String> list=new ArrayList<String>();

public static void main(String[] args) throws IOException {

    do {
        System.out.println("\nEnter V to view all Doctor details\n" +
            "Enter A to add Doctors\n" +
            "Enter EX to Exit the Program\n" +
            "Enter D to delete Doctors\n" +
            "Enter S to store Doctor details into files\n");
        System.out.print("Enter the suitable keyword according to your need : ");
        reply = scanner.nextLine();
        switch (reply) {
            case "V":
                viewDoctors();
                break;
            case "A":
                addDoctors();
                break;
            case "EX":
                exit();
                break;
            case "D" :
                removeDoctors();
                break;
            case "S" :
                storeDoctors();
        }
    } while (true);
}

```



```

        case "A":
            addDoctors();
            break;
        case "EX":
            exit();
            break;
        case "D" :
            removeDoctors();
            break;
        case "S" :
            storeProgramData();
            break;
    }
}
while (true);
}

public static void addDoctors() {

    if (TotDslots == 0) {
        System.out.println("Maximum num of Doctors exceeded");
        scanner.nextLine();
    }
    else {
        removeDoctors();
    }
}

```

```

if (TotDslots == 0) {
    System.out.println("Maximum num of Doctors exceeded");
    scanner.nextLine();
}
else {

    Person pa=new Person();
    System.out.print("Doctor Id:");
    pa.setId(scanner.nextInt());
    System.out.print("First name:");
    pa.setF_name(scanner.next());
    System.out.print("Surname:");
    pa.setS_name(scanner.next());
    System.out.print("Date of Birth:");
    pa.setDateBirth(scanner.next());
    System.out.print("Mobile number:");
    pa.setMobnum(scanner.next());

    Doctor p1 = new Doctor();
    System.out.print("License Number:");
    getLicenseum().add(scanner.next());
    System.out.print("Specialisation:");
    getSpecial().add(scanner.next());

    Personlist.add(pa);
    TotDslots--;
}

```

```
public static void viewDoctors() {  
  
    Collections.sort(Personlist, Person.PersonComparator);  
  
    for(Person str: Personlist){  
  
        System.out.println(str+getLicensenum().get(b)+getSpecial().get(b));  
  
    }  
  
}  
  
public static void removeDoctors() {
```

```

public static void removeDoctors() {


    Scanner scanner1 = new Scanner(System.in);
    System.out.print("Enter Doctor License number: ");
    lnum = scanner1.next();
    for (i=0;i<Personlist.size();i++) {

        vln = getLicenseum().get(i);

        if (vln.equals(lnum)) {
            vlnum = getLicenseum().get(i);
            getLicenseum().remove(lnum);
            spe = getSpecial().get(i);
            getSpecial().remove(spe);
            q = Personlist.get(i).getId();
            f = Personlist.get(i).getF_name();
            s = Personlist.get(i).getS_name();
            d = Personlist.get(i).getDateBirth();
            m = Personlist.get(i).getMobnum();
            Personlist.remove(i);
        }

        System.out.print("Deleted record information: ");
        System.out.println("\n");
        System.out.println("Doctor id" + q);
    }
}

```

n.company.WestminsterSkinConsultationManager >  viewDoctors >

```

    }

    System.out.print("Deleted record information: ");
    System.out.println("\n");
    System.out.println("Doctor id:" + q);
    System.out.println("First name:" + f);
    System.out.println("Surname:" + s);
    System.out.println("Date of Birth:" + d);
    System.out.println("Mobile number:" + m);
    System.out.println("License Number:" + vlnum);
    System.out.println("Specialisation:" + spe);
}

System.out.println("Successfully removed");
}

```

```

public static void storeProgramData() throws IOException {
    File myObj = new File("filename.txt");
    myObj.createNewFile();
    System.out.println("File created: " + myObj.getName());

    FileWriter myWriter = new FileWriter("filename.txt");
    for (int i = 0; i < Personlist.size(); i++) {
        myWriter.write("Doctor id:" + Personlist.get(i).getId() + "\n");
        myWriter.write("First name:" + Personlist.get(i).getF_name() + "\n");
        myWriter.write("Surname:" + Personlist.get(i).getS_name() + "\n");
    }
}

```

h.company.WestminsterSkinConsultationManager > ● viewDoctors >

```

public static void storeProgramData() throws IOException {
    File myObj = new File("filename.txt");
    myObj.createNewFile();
    System.out.println("File created: " + myObj.getName());

    FileWriter myWriter = new FileWriter("filename.txt");
    for (int i = 0; i < Personlist.size(); i++) {
        myWriter.write("Doctor id:"+Personlist.get(i).getId() + "\n");
        myWriter.write("First name:"+Personlist.get(i).getF_name() + "\n");
        myWriter.write("Surname:"+Personlist.get(i).getS_name() + "\n");
        myWriter.write("Date of Birth:"+Personlist.get(i).getDateBirth() + "\n");
        myWriter.write("Mobile Number:"+Personlist.get(i).getMobnum() + "\n");
        myWriter.write("License number:"+getLicensesum().get(i) + "\n");
        myWriter.write("Specialisation:"+getSpecial().get(i) + "\n\n");
    }
    myWriter.close();
    System.out.println("Successfully wrote to the file.");
}

public static void exit() {
    System.out.println("Exit");
    // Terminate JVM
    System.exit(0);
}

```

**Person,**

```

package com.company;

import java.util.Comparator;
import java.util.Locale;

public class Person {

    private int id;
    private String f_name;
    private String s_name;
    private String dateBirth;
    private String mobnum;

    public static Comparator<Person> PersonComparator=new Comparator<Person>(){
        @Override
        public int compare(Person o1, Person o2) {
            String psurname1=o1.getS_name().toUpperCase();
            String psurname2=o2.getS_name().toUpperCase();
            return psurname1.compareTo(psurname2);
        }
    };

    @Override
    public String toString(){
        return id+" "+f_name+" "+s_name+" "+dateBirth+" "+mobnum;
    }

    // public String getString(){
    //     return f_name+" "+s_name+" "+dateBirth+" "+mobnum;
    // }

```

```
public int getId() {  
    return id;  
}  
  
public void setId(int id) {  
    this.id = id;  
}  
  
public String getF_name() {  
    return f_name;  
}  
  
public void setF_name(String f_name) {  
    this.f_name = f_name;  
}  
  
public String getS_name() {  
    return s_name;  
}  
  
public void setS_name(String s_name) {  
    this.s_name = s_name;  
}  
  
public String getDateBirth() {  
    return dateBirth;  
}  
  
public void setDateBirth(String dateBirth) {
```



```
}  
  
public void setS_name(String s_name) {  
    this.s_name = s_name;  
}  
  
public String getDateBirth() {  
    return dateBirth;  
}  
  
public void setDateBirth(String dateBirth) {  
    this.dateBirth = dateBirth;  
}  
  
public String getMobnum() {  
    return mobnum;  
}  
  
public void setMobnum(String mobnum) {  
    this.mobnum = mobnum;  
}  
  
}
```

**Doctor,**

```

package com.company;

import java.util.ArrayList;

public class Doctor extends Person
{
    static ArrayList<Person> Personlist=new ArrayList<>(10);
    private static ArrayList<String> licensenum=new ArrayList<>(10);
    private static ArrayList<String> Special=new ArrayList<>(10);

    public static ArrayList<String> getLicensenum() {
        return licensenum;
    }

    public static void setLicensenum(ArrayList<String> licensenum) {
        Doctor.licensenum = licensenum;
    }

    public static ArrayList<String> getSpecial() {
        return Special;
    }

    public static void setSpecial(ArrayList<String> special) {
        Doctor.Special = special;
    }
}

```

```

    public static ArrayList<String> getLicensenum() {
        return licensenum;
    }

    public static void setLicensenum(ArrayList<String> licensenum) {
        Doctor.licensenum = licensenum;
    }

    public static ArrayList<String> getSpecial() {
        return Special;
    }

    public static void setSpecial(ArrayList<String> special) {
        Special = special;
    }
}

```

# Phase 03 -Source Code

## Main menu,

```
package Phase3;

/**
 *
 * @author seni
 */
public class Mainmenu extends javax.swing.JFrame {

    /**
     * Creates new form Mainmenu
     */
    public Mainmenu() {
        initComponents();
        this.setSize(800, 850);
        this.setResizable(false);
        this.setBounds(600, 100, 800,850);
    }

    @SuppressWarnings("unchecked")
    Generated Code

    private void exitBtnActionPerformed(java.awt.event.ActionEvent evt) {

        System.exit(0);
    }
}
```

```
private void exitBtnActionPerformed(java.awt.event.ActionEvent evt) {  
    System.exit(0);  
}  
  
private void adddoctorBtnActionPerformed(java.awt.event.ActionEvent evt) {  
    Doctor d=new Doctor();  
    d.setVisible(true);  
}  
  
private void addpatientBtnActionPerformed(java.awt.event.ActionEvent evt) {  
    Patient p=new Patient();  
    p.setVisible(true);  
}  
  
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    Consultation k=new Consultation();  
    k.setVisible(true);  
}  
  
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {  
    Cost co=new Cost();  
    co.setVisible(true);  
}
```

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    Consultation k=new Consultation();  
    k.setVisible(true);  
}
```

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {  
    Cost co=new Cost();  
    co.setVisible(true);  
}
```

```
/**  
 * @param args the command line arguments  
 */  
public static void main(String args[]) {  
    /* Set the Nimbus look and feel */  
    Look and feel setting code (optional)  
  
    /* Create and display the form */  
    java.awt.EventQueue.invokeLater(new Runnable() {  
        public void run() {  
            new Mainmenu().setVisible(true);  
        }  
    });  
}
```

```
// Variables declaration - do not modify  
private javax.swing.JButton adddoctorBtn;
```

```

/* Set the Nimbus look and feel */
Look and feel setting code (optional)

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new Mainmenu().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton adddoctorBtn;
private javax.swing.JButton addpatientBtn;
private javax.swing.JButton exitBtn;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JPanel jPanel1;
// End of variables declaration
}

```

**Consultation,**

```

package Phase3;

import java.io.FileWriter;
import java.io.IOException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

/**
 *
 * @author seni
 */
public class Consultation extends javax.swing.JFrame {

    static String rf,
    static int x=0;
    static int y=0;
    static String c1=" ";

    /**
     * Creates new form Consultation
     */
    public Consultation() {

        initComponents();

        this.setResizable(false);
    }
}

```

```

    this.setResizable(false);

    this.setBounds(550, 50, 900, 1000);
}

```

```

@SuppressWarnings("unchecked")

```

Generated Code

```

private void addBtnActionPerformed(java.awt.event.ActionEvent evt) {

```

```

    String a=" ";
    String b=" ";
    String d=" ";
    String j=" ";
    String f=" ";
    String h=" ";

```

```

    a=textDoctorn.getText();
    b=textDateConsult.getText();
    d=textConsultationtime.getText();
    f=textNotearea.getText();

```

```

    Noteencrypt c=new Noteencrypt();
    String g= textNotearea.getText();
    String n=c.edisplay(g);

```



```

String n=c.display(g);
textEncrypt.setText(n);

// Patient p1=new Patient();
textPatientName.setText(rf);
textCost.setText(ct);

try
{
    h=ct;
    j=rf;

    // File myObj = new File("filename.txt");
    // myObj.createNewFile();
    // System.out.println("File created: " + myObj.getName());

    FileWriter myWriter = new FileWriter("Consultation.txt",true);

    myWriter.write("Doctor Name:"+a+ "\n");
    myWriter.write("Date of Consultation:"+b+ "\n");
    myWriter.write("Time of Consultation:"+d+ "\n");
    myWriter.write("Patient Name:"+j+ "\n");
    myWriter.write("Patient Note:"+f+ "\n");
    myWriter.write("Cost For Consultation:"+h+ "\n");
}

```

```

myWriter.close();
System.out.println("Successfully wrote to the file.");
}
catch(IOException e)
{
    JOptionPane.showMessageDialog(null,e.getMessage(),"Error",JOptionPane.INFORMATION_MESSAGE);
}

}

private void clearBtnActionPerformed(java.awt.event.ActionEvent evt) {
    textDoctorn.setText("");
    textDateConsult.setText("");
    textConsultationtime.setText("");
    textPtientName.setText("");
    textNotearea.setText("");
    textEncrypt.setText("");
    textCost.setText("");
}

private void closeBtnActionPerformed(java.awt.event.ActionEvent evt) {
    Mainmenu m=new Mainmenu();
    m.setVisible(true);
}

```

```

Mainmenu m=new Mainmenu();
m.setVisible(true);
}

private void doctordetailsBtnActionPerformed(java.awt.event.ActionEvent evt) {

    Doctorlist dl=new Doctorlist();
    dl.setVisible(true);
}

private void patientdetailsBtnActionPerformed(java.awt.event.ActionEvent evt) {

//    Patient p=new Patient();
//    p.setVisible(true);

    if(x==0){

        Patient p1=new Patient();
        p1.setVisible(true);
    }

    x=x+1;
}

```

```

private void costcalBtnActionPerformed(java.awt.event.ActionEvent evt) {

    if(y==0){

        Cost p2=new Cost();
        p2.setVisible(true);
    }

    y=y+1;
}

private void textDoctornActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void textDateConsultActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void textConsultationtimeActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void textPtientNameActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```



```
package Phase3;

/**
 *
 * @author seni
 */
public class Cost extends javax.swing.JFrame {

    static int amount=0;
    static String co=" ";

    /**
     * Creates new form Cost
     */
    public Cost() {
        initComponents();
        this.setBounds(600, 100, 800,850);
    }

    @SuppressWarnings("unchecked")
    Generated Code

    private void textAddhoursActionPerformed(java.awt.event.ActionEvent evt) {
```

```
}
```

```
private void calculateBtnActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    int cr=0;
```

```
    if(comboRate.getSelectedIndex()==0){
```

```
        cr=10;
```

```
    }
```

```
    if(comboRate.getSelectedIndex()==1){
```

```
        cr=15;
```

```
    }
```

```
    if(comboRate.getSelectedIndex()==2){
```

```
        cr=20;
```

```
    }
```

```
    if(comboRate.getSelectedIndex()==3){
```

```
        cr=25;
```

```
    }
```

```
    int ah = Integer.parseInt(textAddhours.getText());
```

```
    amount=cr*ah;
```

```
    textfinalcost.setText(amount+" ");
```

```
}
```

```

private void addBtnActionPerformed(java.awt.event.ActionEvent evt) {

    // co=Integer.parseInt(textfinalcost.getText());
    co=textfinalcost.getText();
    Consultation.c1=co;
    Consultation con=new Consultation();
    con.revalidate();
    con.repaint();

}

private void textfinalcostActionPerformed(java.awt.event.ActionEvent evt) {

}

private void closeBtnActionPerformed(java.awt.event.ActionEvent evt) {
    Consultation co=new Consultation();
    co.setVisible(true);
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    Look and feel setting code (optional)

    /* Create and display the form */

```



```

    * @param args the command line arguments
    */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        Look and feel setting code (optional)

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Cost().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton addBtn;
    private javax.swing.JButton calculateBtn;
    private javax.swing.JButton closeBtn;
    private javax.swing.JComboBox<String> comboRate;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JTextField textAddhours;
    private javax.swing.JTextField textfinalcost;
    // End of variables declaration
}

```

**Doctor,**

```

package Phase3;

import java.util.ArrayList;

/**
 *
 * @author seni
 */
public class Doctor extends javax.swing.JFrame {

    /**
     * Creates new form Doctor
     */

    String did="";
    String dname="";
    String dsurname="";
    String ddateofbirth="";
    String dmobnum="";
    String dspecialisation="";
    String dlnum="";

    public Doctor() {

        initComponents();
        this.setSize(800, 850);
        this.setTitle("Doctor Form");
    }
}

```

```
this.setSize(800, 850);  
this.setResizable(false);  
this.setBounds(600, 100, 800, 850);
```

```
}
```

```
public static ArrayList<Person1> doctorList=new ArrayList<>(5);
```

```
@SuppressWarnings("unchecked")
```

```
Generated Code
```

```
private void addBtnActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    did=textId.getText();  
    dname=textName.getText();  
    dsurname=textSurname.getText();  
    ddateofbirth=textDateofbirth.getText();  
    dmobnum=textMobnum.getText();  
    dspecialisation=textSpecialisation.getText();  
    dlnum=textLnum.getText();
```

```
    Person1 p=new Person1();  
    p.setId(did);  
    p.setF_name(dname);  
    p.setS_name(dsurname);  
    p.setDateBirth(ddateofbirth);  
    p.setMobnum(dmobnum);  
    p.setDspecialisation(dspecialisation);  
    p.setDlnum(dlnum);
```

```
p.setDnum(dnum);
```

```
doctorList.add(p);
```

```
}
```

```
private void clearBtnActionPerformed(java.awt.event.ActionEvent evt) {
```

```
textId.setText(" ");
```

```
textName.setText(" ");
```

```
textSurname.setText(" ");
```

```
textDateofbirth.setText(" ");
```

```
textMobnum.setText(" ");
```

```
textSpecialisation.setText(" ");
```

```
textLnum.setText(" ");
```

```
}
```

```
private void doctorBtnActionPerformed(java.awt.event.ActionEvent evt) {
```

```
Doctorlist dl=new Doctorlist();
```

```
dl.setVisible(true);
```

```
}
```

```
private void textLnumActionPerformed(java.awt.event.ActionEvent evt) {
```

```
,
```

```

    }

    private void textLnumActionPerformed(java.awt.event.ActionEvent evt) {

    }

    private void closeBtnActionPerformed(java.awt.event.ActionEvent evt) {
        Mainmenu m=new Mainmenu();
        m.setVisible(true);
    }

    private void textNameActionPerformed(java.awt.event.ActionEvent evt) {

    }

    private void textIdActionPerformed(java.awt.event.ActionEvent evt) {

    }

```

```

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    Look and feel setting code (optional)

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {

```

```

java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new Doctor().setVisible(true);
    }
});
}

```

// Variables declaration - do not modify

```

private javax.swing.JButton addBtn;
private javax.swing.JButton clearBtn;
private javax.swing.JButton closeBtn;
private javax.swing.JButton doctorBtn;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JTextField textDateofbirth;
private javax.swing.JTextField textId;
private javax.swing.JTextField textLnum;
private javax.swing.JTextField textMobnum;
private javax.swing.JTextField textName;
private javax.swing.JTextField textSpecialisation;
private javax.swing.JTextField textSurname;

```

se3.Doctor > textIdActionPerformed >

**Doctor list,**

```

package Phase3;

import java.util.Collections;
import static Phase3.Doctor.doctorList;

/**
 *
 * @author seni
 */
public class Doctorlist extends javax.swing.JFrame {

    static String gh;

    /**
     * Creates new form Doctorlist
     */
    public Doctorlist() {
        initComponents();
        this.setBounds(600, 100, 800, 850);
    }

    @SuppressWarnings("unchecked")
    Generated Code

    private void formWindowOpened(java.awt.event.WindowEvent evt) {
        for(int a = 0; a<5;a++){

```

```

        Doctor.doctorList.get(a).getF_name()+" "+
        Doctor.doctorList.get(a).getS_name()+" "+
        Doctor.doctorList.get(a).getDateBirth()+" "+
        Doctor.doctorList.get(a).getMobnum()+" "+
        Doctor.doctorList.get(a).getDlnum()+" "+
        Doctor.doctorList.get(a).getDspecialisation());
    }
}

private void sortBtnActionPerformed(java.awt.event.ActionEvent evt) {

    Collections.sort(doctorList,Person1.PersonComparator);

    for(Person1 str1:doctorList){
        Person1 p2=new Person1();
        // String s=p2.toString();
        comboDoctor.addItem(p2.getString());
        System.out.print(str1);
    }
}

private void comboDoctorActionPerformed(java.awt.event.ActionEvent evt) {

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    Doctor doc=new Doctor();
    // String s=doc.toString();
}

```

se3.Doctorlist > comboDoctorActionPerformed >



```

        Doctor doc=new Doctor();
        doc.setVisible(true);
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        Look and feel setting code (optional)

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Doctorlist().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JComboBox<String> comboDoctor;
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JButton sortBtn;
    // End of variables declaration
}

```

**Note encrypt,**

```

package Phase3;

import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;

/**
 *
 * @author seni
 */
public class Noteencrypt {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {

        Noteencrypt p=new Noteencrypt();

    }

    public String edisplay(String s){

        String ps="";

        try{

            // Generate object of KeyGenerator & SecretKey
            KeyGenerator keyGenerator=KeyGenerator.getInstance("DES");

```

```

// Generate object of KeyGenerator & SecretKey
KeyGenerator keygenerator =KeyGenerator.getInstance("DES");
SecretKey myDesKey=keygenerator.generateKey();

//Creating object of Cipher

Cipher desCipher;
desCipher= Cipher.getInstance("DES");

//Creating Byte Array to store string byte[ ] text = " No body can see me,".getBytes("UTF8");

byte[] text = s.getBytes("UTF8");

//Encrypting text
desCipher.init(Cipher.ENCRYPT_MODE,myDesKey);
byte[] Noteencrypt= desCipher.doFinal(text);

//Converting encrypted byte array to string
ps= new String( Noteencrypt);

}

catch (Exception e){
System.out.println("Exception");
}

```

**Patient.**

```

package Phase3;

import java.io.File;
import java.io.FileWriter;
import javax.swing.JOptionPane;

/**
 *
 * @author seni
 */
public class Patient extends javax.swing.JFrame {

    /**
     * Creates new form Patient
     */

    String vid="";
    String vname="";
    String vsurname="";
    String vdateofbirth="";
    String vmobnum="";
    static String df,

    public Patient() {
        initComponents();
        this.setSize(800,850);
        this.setResizable(false);
    }

```

```
    this.setResizable(false);  
    this.setBounds(600, 100, 800,850);  
}
```

```
@SuppressWarnings("unchecked")
```

```
Generated Code
```

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    jTextField1.setText(" ");  
    jTextField2.setText(" ");  
    jTextField3.setText(" ");  
    jTextField4.setText(" ");  
    jTextField5.setText(" ");  
}
```

```
private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {  
}
```

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    vid=jTextField1.getText();  
    vname=jTextField2.getText();  
    vsurname=jTextField3.getText();  
    vdateofbirth=jTextField4.getText();  
    vmobnum=jTextField5.getText();  
  
    try
```

```

// File myObj = new File("filename.txt");
// myObj.createNewFile();
// System.out.println("File created: " + myObj.getName());

FileWriter myWriter = new FileWriter("Patient.txt",true);

myWriter.write("Patient id:"+vid+ "\n");
myWriter.write("First name:"+vname+ "\n");
myWriter.write("Surname:"+vsurname+ "\n");
myWriter.write("Date of Birth:"+vdateofbirth+ "\n");
myWriter.write("Mobile Number"+vmobnum+ "\n");


myWriter.close();
System.out.println("Successfully wrote to the file.");
}
catch(Exception e)
{
    JOptionPane.showMessageDialog(null,e.getMessage(),"Error",JOptionPane.INFORMATION_MESSAGE);
}

df=jTextField2.getText();

Consultation rf=df;
Consultation d=new Consultation();
d.revalidate();
}

```

```

    }

    private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {

    }

    private void closeBtnActionPerformed(java.awt.event.ActionEvent evt) {
        Mainmenu m=new Mainmenu();
        m.setVisible(true);
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        Look and feel setting code (optional)
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Patient().setVisible(true);
            }
        });
    }
}

```

```

    }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton closeBtn;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JPanel jPanel1;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField3;
private javax.swing.JTextField jTextField4;
private javax.swing.JTextField jTextField5;
// End of variables declaration
}

```

**Persone1,**



```
package Phase3;
```

```
import java.util.ArrayList;
```

```
import java.util.Comparator;
```

```
public class Person1 {
```

```
    private String id;
```

```
    private String f_name;
```

```
    private String s_name;
```

```
    private String dateBirth;
```

```
    private String mobnum;
```

```
    private String dspecialisation;
```

```
    private String dlnum;
```

```
    public static Comparator<Person1> PersonComparator=new Comparator<Person1>(){
```

```
        @Override
```

```
        public int compare(Person1 o1, Person1 o2) {
```

```
            String psurname1=o1.getS_name().toUpperCase();
```

```
            String psurname2=o2.getS_name().toUpperCase();
```

```
            return psurname1.compareTo(psurname2);
```

```
        };
```

```
        @Override
```

```
        public String toString(){
```

```
            return id+" "+f_name+" "+s_name+" "+dateBirth+" "+mobnum+" "+dspecialisation+" "+dlnum;
```

```
        return id+" "+f_name+" "+s_name+" "+dateBirth+" "+mobnum+" "+dspecialisation+" "+dlnum;
    }

    public String getString(){
        return toString();
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getF_name() {
        return f_name;
    }

    public void setF_name(String f_name) {
        this.f_name = f_name;
    }

    public String getS_name() {
        return s_name;
    }
}
```

```

    }

    public void setS_name(String s_name) {
        this.s_name = s_name;
    }

    public String getDateBirth() {
        return dateBirth;
    }

    public void setDateBirth(String dateBirth) {
        this.dateBirth = dateBirth;
    }

    public String getMobnum() {
        return mobnum;
    }

    public void setMobnum(String mobnum) {
        this.mobnum = mobnum;
    }

    /**
     * @return the dspecialisation
     */
    public String getDspecialisation() {
        return dspecialisation;
    }

```

```

    * @param dspecialisation the dspecialisation to set
    */
    public void setDspecialisation(String dspecialisation) {
        this.dspecialisation = dspecialisation;
    }

    /**
     * @return the dlnum
     */
    public String getDlnum() {
        return dlnum;
    }

    /**
     * @param dlnum the dlnum to set
     */
    public void setDlnum(String dlnum) {
        this.dlnum = dlnum;
    }
}

```

# Test Plan

Have you submitted the video with the demonstration of your system?

☒ Yes

☐ No

## Phase 1 – Design and classes implementation

Task	Did you attempt the task?	Student's comments (To which extent you implemented the task? Have you encountered any problems or issue?)
Design a UML Use Case Diagram of your system (submitted in a separate file).	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	Designed a UML Use Case Diagram and submitted it in a separate file.
Design a UML Class Diagram of your system (submitted in a separate file).	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	Designed a UML Class Diagram and submitted it in a separate file.
Implementation Class Person	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	Implemented a Class Person
Implementation Class Doctor	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	Implemented a Class Doctor
Implementation Class Patient	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	Implemented a Class Patient
Implementation Class Consultation	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	Implemented a Class Consultation
Implementation Interface WestminsterSkinConsultationManager	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	Implemented the Interface WestminsterSkinConsultationManager

--	--	--

## **Phase 2 – Console menu implementation**

<b>Task</b>	<b>Did you attempt the task?</b>	<b>Student's comments</b> (To which extent you implemented the task? Have you encountered any problems or issue?)
Add a doctor in the system with all the relative information (max 10 doctors)	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	Add 10 doctors in the system with all the relative information .
Delete a doctor from the system selecting the medical licence number. Display a message to confirm he/she has been removed and the total number of doctors in the centres.	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	Delete a doctor from the system by selecting the medical license number. Display a message to confirm he/she has been removed and the total number of doctors in the centres.
Print on the screen the list the doctors in the centre with all the relative information. The list should be ordered alphabetically.	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	Print on the screen the list of the doctors in the centre with all the relative information. The list is ordered alphabetically.
Save in a file entered by the user so far. The user should be able to load back the information running a new instance of the application.	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	Save in a file entered by the user. The user is able to load back the information by running a new instance of the application.

## **Phase 3 – GUI Implementation**

<b>Task</b>	<b>Did you attempt the task?</b>	<b>Student's comments</b> (To which extent you implemented the task? Have you encountered any problems or issue?)
Doctor list visualisation. Sorting alphabetically.	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	The doctor list visualizes and sorted alphabetically.
The user can select a doctor and add a consultation.	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	The user can select a doctor and add a consultation.

In the consultation the user can add all the patient details.	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	In the consultation, the user can add all the patient details.
The user can select the date/time of the consultation considering that a doctor cannot have more than one consultation at the time.	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	The user can select the date/time of the consultation considering that a doctor cannot have more than one consultation at a time.
The user can enter and save the cost for the consultation. (£25 per hour and only the first one £15).	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	The user can enter and save the cost for the consultation. (£25 per hour and only the first one £15).
The user can add some notes (text information or images). This information has been encrypted.	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	The user can add some notes (text information or images). This information has been encrypted

#### **Phase 4 – Testing and system validation**

<b>Task</b>	<b>Did you attempt the task?</b>	<b>Student's comments</b> (To which extent you implemented the task? Have you encountered any problems or issue?)
Test plan. (Submitted in a separate file).	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	Test plan. (Submitted in a separate file).
Implementation of an automated unit test for each scenario in the console menu.	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	Implementation of an automated unit test for each scenario in the console menu.
Error Handling across all the code, input validation and code quality.	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	Error Handling across all the code, input validation and code quality.

# Reference

<https://www.w3schools.com/java/default.asp>