

AMRITA SCHOOL OF ENGINEERING, BANGALORE

AMRITA VISHWA VIDYAPEETHAM

**BANGALORE 560 035**

**CLINIC MANAGEMENT SYSTEM**

A PROJECT REPORT

***Submitted by***

BL.EN.U4AIE19068 AISHWARYA VIRIGINENI

For the subject

19AIE111- Data Structures and Algorithms -1

Faculty Incharge

Ms. D. Radha

Asst. Prof(SG),

CSE, ASE-BLR

1. **Abstract/ Synopsis/ Description of the project**

Clinic Management System is about giving doctor’s appointment to the patients. In giving appointment, the basic details of the patient like name, age, gender, phone number will be noted. Patients will also be asked about the doctor (specialist) they want to meet. For example, the doctor can be a Cardiologist, Gynaecologist, Neurologist, etc.; When finally all the details are noted, he’ll be asked to pay the fee and then he’ll be given a token number. And after that he’ll be added into the waiting list. There is also special appointment available here, where the patient will be charged extra and is placed in the first position of the waiting list exceeding the others. The list is made using the data structure Singly Linked List. Finally, the patient on the top of the list will be called out by displaying his token no., name and the specialist on the screen available for patients.

1. **Explanation of the Data Structure and the GUI used**

**Data Structure used: *SINGLY* *LINKED LIST***

Singly Linked List:

A singly linked list is a collection of ordered set of elements. It is represented by a pointer to the first node of the linked list. The first node is called the head. If the linked list is empty, then the value of the head is NULL.  
Each node in a list consists of at least two parts:  
1) data  
2) Pointer (Or Reference) to the next node

One way chain or singly linked list can be traversed only in one direction. In other words, we can say that each node contains only next pointer, therefore we can not traverse the list in the reverse direction.

In Java , Singly Linked List can be represented as a class and a Node as a separate class. The LinkedList class contains a reference of Node class type.

Methods used from Singly linked list:

* Insert\_first()
* Insert\_last()
* Delete\_first()
* checkEmpty()
* count()

**GUI used: *SWING & AWT***

Swing:

**Java Swing** is a part of Java Foundation Classes (JFC) that is used to create window-based applications. It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java.

Unlike AWT, Java Swing provides platform-independent and lightweight components.

The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc

Awt:

**Java AWT** (Abstract Window Toolkit) is an API to develop GUI or window-based applications in java.

Java AWT components are platform-dependent i.e. components are displayed according to the view of operating system. AWT is heavyweight i.e. its components are using the resources of OS.

The java.awt [package](https://www.javatpoint.com/package) provides [classes](https://www.javatpoint.com/object-and-class-in-java) for AWT api such as [TextField](https://www.javatpoint.com/java-awt-textfield), [Label](https://www.javatpoint.com/java-awt-label), [TextArea](https://www.javatpoint.com/java-awt-textarea), RadioButton, [CheckBox](https://www.javatpoint.com/java-awt-checkbox), [Choice](https://www.javatpoint.com/java-awt-choice), [List](https://www.javatpoint.com/java-awt-list) etc

1. **Code**

**import** javax.swing.\*;

**import** javax.swing.table.DefaultTableModel;

**import** java.awt.\*;

**import** java.awt.event.\*;

**class** patient {

String name,age,gender,mobile,doc;

**int** id;

**public** patient(**int** i,String name,String age,String gender,String mobile,String doc){

**this**.age = age;

**this**.doc = doc;

**this**.name = name;

**this**.gender = gender;

**this**.mobile = mobile;

**this**.id=i;

}

}

**class** node

{

patient data;

node next,size;

node()

{

data=**null**;

next=**null**;

}

node(patient st)

{

**this**.data=st;

next=**null**;

}

}

**class** SLL

{

node head= **null**;

**int** size=0;

**void** insert\_first(patient st)

{

**if** (head==**null**)

head=**new** node(st);

**else**

{

node t=**new** node(st);

t.next=head;

head=t;

}

size++;

}

**void** insert\_last(patient st)

{

node n=**new** node(st);

node t=head;

**if**(t==**null**)

{head=n;head.next=**null**;size++;}

**else**

{

**while**(t.next!=**null**)

{

t=t.next;

}

t.next=n;

n.next=**null**;

size++;

}

}

**void** delete\_first()

{

**if**(head==**null**)

System.***out***.println("Empty list");

**else**

{head=head.next;

size--;}

}

**public** **int** checkEmpty()

{

**if**(head==**null**)

{

**return** 0;

}

**else**

**return** 1;

}

**public** **int** count()

{

**return** size;

}

**public** String[][] getData(){

String s[][] = **new** String[size][6];

**int** i = 0;

node aNode = head;

**while** (aNode != **null**) {

patient tmp = aNode.data;

s[i][0] = String.*valueOf*(tmp.id);

s[i][1] = tmp.name;

s[i][2] = String.*valueOf*(tmp.age);

s[i][3] = tmp.gender;

s[i][4] = String.*valueOf*(tmp.mobile);

s[i][5] = tmp.doc;

i += 1;

aNode = aNode.next;

}

**return** s;

}

}

**public** **class** Final\_DS **extends** JFrame **implements** ActionListener{

Container c;

JTextField txtName,txtAge,txtGender,txtMobile;

JLabel lblTitle,lblName,lblGender,lblAge,lblMobile,lblSpecialist;

JButton btnAdd,btnUpdate,btnDelete,btnPrint,btnSort,btnRefresh,btnCancel,btnDone,btnSpecial;

JTable table;

JTextArea textBox;

JComboBox cb,dc;

String data[][];

SLL l;

**public** Final\_DS(){

c = getContentPane();

c.setFont(**new** Font("Consolas", Font.***PLAIN***, 13));

c.setBackground(**new** Color(32,178,170));

setTitle(" Clinic Registration Form ");

c.setLayout(**null**);

l = **new** SLL();

// add lbl

lblTitle = **new** JLabel("ENTER PATIENT DETAILS ");

lblTitle.setFont(**new** Font ("Tahoma", Font.***BOLD*** , 18));

lblTitle.setBounds(40,20,250,20);

c.add(lblTitle);

lblName = **new** JLabel("Name");

lblName.setFont(**new** Font ("Tahoma", Font.***BOLD*** , 15));

lblName.setBounds(40,60,70,20);

c.add(lblName);

lblAge = **new** JLabel("Age");

lblAge.setFont(**new** Font ("Tahoma", Font.***BOLD*** , 15));

lblAge.setBounds(40,90,70,20);

c.add(lblAge);

lblGender = **new** JLabel("Gender");

lblGender.setFont(**new** Font ("Tahoma", Font.***BOLD*** , 15));

lblGender.setBounds(40,120,70,20);

c.add(lblGender);

lblMobile = **new** JLabel("Mobile");

lblMobile.setFont(**new** Font ("Tahoma", Font.***BOLD*** , 15));

lblMobile.setBounds(40,150,70,20);

c.add(lblMobile);

lblSpecialist = **new** JLabel("Specialist");

lblSpecialist.setFont(**new** Font ("Tahoma",Font.***BOLD*** , 15));

lblSpecialist.setBounds(40,180,80,20);

c.add(lblSpecialist);

txtName = **new** JTextField();

txtName.setBounds(140,60,120,20);

c.add(txtName);

txtAge = **new** JTextField();

txtAge.setBounds(140,90,120,20);

c.add(txtAge);

txtMobile = **new** JTextField();

txtMobile.setBounds(140,150,120,20);

c.add(txtMobile);

textBox = **new** JTextArea();

textBox.setFont(**new** Font("Consolas", Font.***BOLD***|Font.***ITALIC***, 18));

textBox.setSize(400, 200);

textBox.setLocation(550, 20);

textBox.setLineWrap(**true**);

textBox.setEditable(**false**);

c.add(textBox);

String sex[]={"Male","Female"};

cb=**new** JComboBox(sex);

cb.setBounds(140, 120, 120, 20);

c.add(cb);

String doctor[]={"Neurologist","Gynaecologist","Dermatologist","Cardiologist","Opthamologist"};

dc=**new** JComboBox(doctor);

dc.setBounds(140, 180, 120, 20);

c.add(dc);

// add btn

btnAdd = **new** JButton(" Add ");

btnAdd.setBounds(315,59,80,29);

c.add(btnAdd);

btnRefresh = **new** JButton("Refresh");

btnRefresh.setBounds(160,221,80,25);

c.add(btnRefresh);

btnCancel= **new** JButton("Exit");

btnCancel.setBounds(315,139,80,25);

c.add(btnCancel);

btnDelete= **new** JButton("Service");

btnDelete.setBounds(315,101,80,25);

c.add(btnDelete);

btnDone= **new** JButton("Done");

btnDone.setBounds(710,240,80,25);

c.add(btnDone);

btnSpecial= **new** JButton("Special");

btnSpecial.setBounds(315,179,80,25);

c.add(btnSpecial);

btnAdd.addActionListener(**this**);

btnRefresh.addActionListener(**this**);

btnCancel.addActionListener(**this**);

btnDelete.addActionListener(**this**);

btnDone.addActionListener(**this**);

btnSpecial.addActionListener(**this**);

setSize(1000, 920);

setLocationRelativeTo(**null**);

setDefaultCloseOperation(***EXIT\_ON\_CLOSE***);

setVisible(**true**);

}

**public** **void** actionPerformed(ActionEvent e){

table = **new** JTable();

Object col[] = {"Index","Name","Age","Gender","Mobile","Specialist"};

**if**(e.getSource() == btnCancel)

System.*exit*(0);

**else** **if**(e.getSource() == btnRefresh){

txtName.setText("");

txtAge.setText("");

txtMobile.setText("");

}

**else** **if**(e.getSource() == btnAdd){

**if**(txtName.getText().isEmpty()||txtAge.getText().isEmpty()||txtMobile.getText().isEmpty())

JOptionPane.*showMessageDialog*(**null**,"Please fill in all the details");

**else** **if**(isString(txtName.getText().trim()) == **false**)

JOptionPane.*showMessageDialog*(**null**,"you must use only letters in name");

**else** **if**(isCapital(txtName.getText().trim()) == **false**)

JOptionPane.*showMessageDialog*(**null**,"you must begin Name with capital Character");

**else** **if**(isString(txtMobile.getText().trim()))

JOptionPane.*showMessageDialog*(**null**,"Enter only numbers");

**else** **if**(is10Digit(txtMobile.getText().trim())==**false**)

JOptionPane.*showMessageDialog*(**null**,"Enter mobile no. with 10 digits");

**else** **if**(checkAge(txtAge.getText().trim()) == **false**)

JOptionPane.*showMessageDialog*(**null**,"You must enter real age in numbers");

**else**{

patient st = **new** patient(l.count()+1, txtName.getText(),txtAge.getText(),cb.getSelectedItem().toString(),txtMobile.getText(),dc.getSelectedItem().toString());

l.insert\_last(st);

data = l.getData();

DefaultTableModel model = **new** DefaultTableModel(data,col);

table.setModel(model);

JScrollPane scrollPane = **new** JScrollPane(table);

scrollPane.setBounds(9,300,965,565);

c.add(scrollPane);

// Change A JTable Background Color, Font Size, Font Color, Row Height

table.setBackground(Color.***LIGHT\_GRAY***);

table.setForeground(Color.***black***);

Font font = **new** Font("",1,22);

table.setFont(font);

table.setRowHeight(30);

c.setLayout(**null**);

}

}

**else** **if**(e.getSource() == btnSpecial){

**if**(txtName.getText().isEmpty()||txtAge.getText().isEmpty()||txtMobile.getText().isEmpty())

JOptionPane.*showMessageDialog*(**null**,"Please fill in all the details");

**else** **if**(isCapital(txtName.getText().trim()) == **false**)

JOptionPane.*showMessageDialog*(**null**,"you must begin Name with capital Character");

**else** **if**(isString(txtName.getText().trim()) == **false**)

JOptionPane.*showMessageDialog*(**null**,"you must use only letters");

**else** **if**(isString(txtMobile.getText().trim()))

JOptionPane.*showMessageDialog*(**null**,"Enter only numbers");

**else** **if**(is10Digit(txtMobile.getText().trim())==**false**)

JOptionPane.*showMessageDialog*(**null**,"Enter mobile no. with 10 digits");

**else** **if**(checkAge(txtAge.getText().trim()) == **false**)

JOptionPane.*showMessageDialog*(**null**,"You must enter real age in digits");

**else**{

patient st = **new** patient(l.count()+1, txtName.getText(),txtAge.getText(),cb.getSelectedItem().toString(),txtMobile.getText(),dc.getSelectedItem().toString());

l.insert\_first(st);

data = l.getData();

DefaultTableModel model = **new** DefaultTableModel(data,col);

// set the model to the table

table.setModel(model);

JScrollPane scrollPane = **new** JScrollPane(table);

scrollPane.setBounds(9,300,965,565);

c.add(scrollPane);

table.setBackground(Color.***LIGHT\_GRAY***);

table.setForeground(Color.***black***);

Font font = **new** Font("",1,22);

table.setFont(font);

table.setRowHeight(30);

c.setLayout(**null**);

}

}

**else** **if**(e.getSource() == btnDelete) {

**if**(l.checkEmpty()==0)

{

JOptionPane.*showMessageDialog*(**null**,"List is empty ");

}

**else**

{

l.delete\_first();

data=l.getData();

table = **new** JTable();

DefaultTableModel model = **new** DefaultTableModel(data,col);

// set the model to the table

table.setModel(model);

**if**(l.checkEmpty()!=0)

{

textBox.setText("Token no. " + model.getValueAt(0, 0).toString() + " " + "\n\n\n Please be ready");

Font font3 = **new** Font("",1,22);

textBox.setFont(font3);

}

JScrollPane scrollPane = **new** JScrollPane(table);

scrollPane.setBounds(9,300,965,565);

c.add(scrollPane);

// Change A JTable Background Color, Font Size, Font Color, Row Height

table.setBackground(Color.***LIGHT\_GRAY***);

table.setForeground(Color.***black***);

Font font = **new** Font("",1,22);

table.setFont(font);

table.setRowHeight(27);

}

}

**else** **if**(e.getSource() == btnDone){

**if**(l.checkEmpty()==0)

{

textBox.setText(" ");

JOptionPane.*showMessageDialog*(**null**,"List is empty!! ");

}

**else** {

data=l.getData();

table = **new** JTable();

DefaultTableModel model = **new** DefaultTableModel(data,col);

table.setModel(model);

textBox.setText(" Token no. "+model.getValueAt(0, 0).toString()+" \n '"+model.getValueAt(0, 1).toString()+

"' \n "+model.getValueAt(0, 5).toString()+" \n Please come in!!");

Font font = **new** Font("",1,22);

textBox.setFont(font);

DefaultTableModel model2 = **new** DefaultTableModel(data,col);

JScrollPane scrollPane = **new** JScrollPane(table);

scrollPane.setBounds(9,300,965,565);

c.add(scrollPane);

table.setBackground(Color.***LIGHT\_GRAY***);

table.setForeground(Color.***black***);

Font font2 = **new** Font("",1,22);

table.setFont(font2);

table.setRowHeight(30);

}

}

}

//~~~~~~~~~~~~ Constraints

**public** **boolean** is10Digit(String mobile){

**if**(mobile.length() == 10)

**return** **true**;

**return** **false**;

}

**public** **boolean** checkAge(String age){

**if**(!isNumber(age) || Integer.*parseInt*(age) > 100 || Integer.*parseInt*(age) < 6 )

**return** **false**;

**return** **true**;

}

**public** **boolean** isNumber(String id){

**for**(**int** i=0;i<id.length();i++){

**char** c=id.charAt(i);

**if**(c>'9'|| c<'0')

**return** **false**;

}

**return** **true**;

}

**public** **boolean** isString(String id){

**for**(**int** i=0;i<id.length();i++){

**char** c=id.charAt(i);

**if**(c<='9'|| c<='0')

**return** **false**;

}

**return** **true**;

}

**public** **boolean** isCapital(String Name){

**char** c=Name.charAt(0);

**if**(c>='A'&& c<='Z')

**return** **true**;

**return** **false**;

}

**public** **static** **void** main(String args[])

{

**new** Final\_DS();

}

}

1. **Snapshots of the project**

Patients added in the list by clicking on ***add button*** are in the following order :

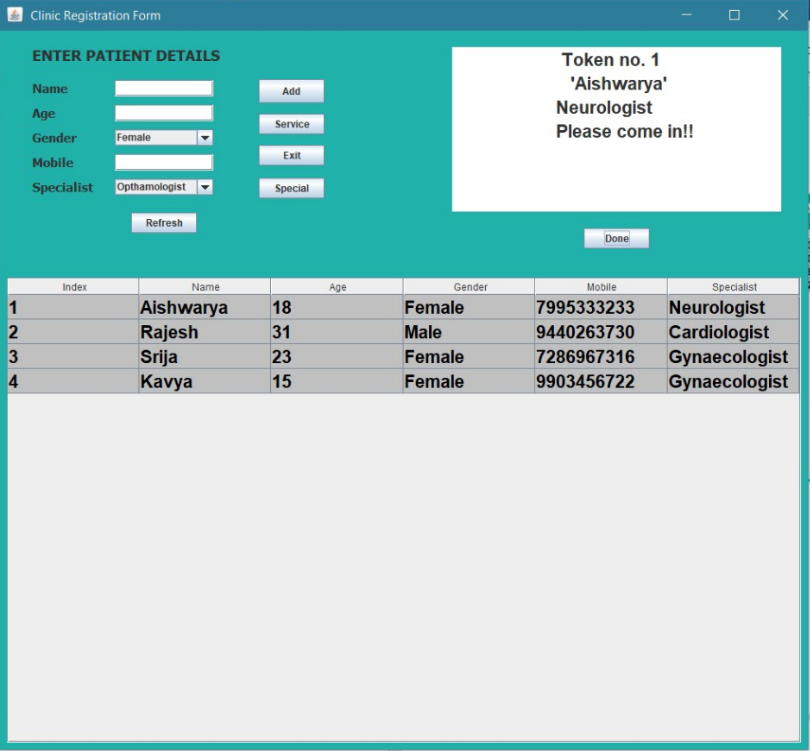
Aishwarya

Rajesh

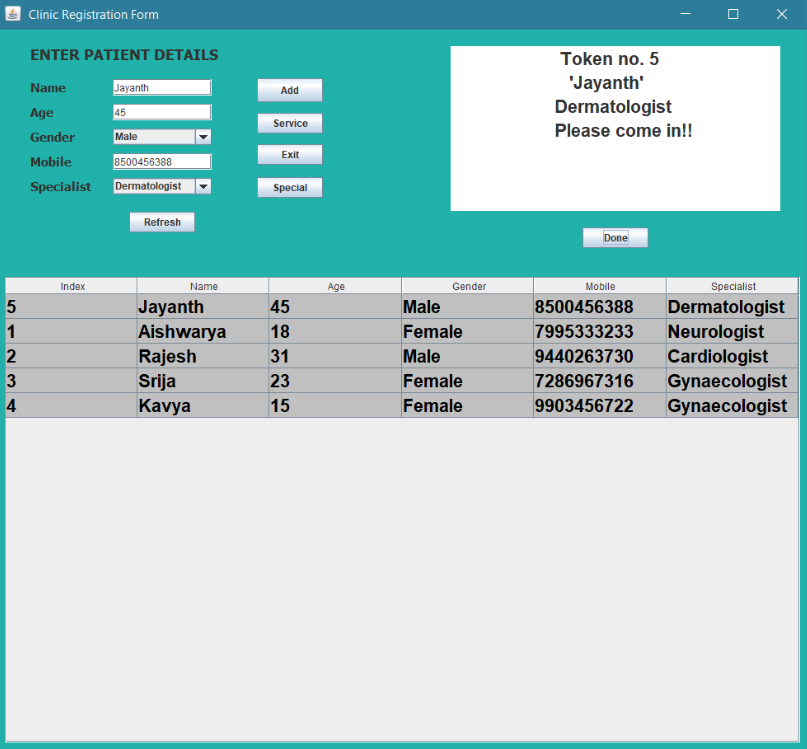
Srija

Kavya

Aishwarya is called out as she’s in the first of the list.

****

Now, Jayanth is added using special appointment using ***special button***, so moves to the first position.

****

After clicking on ***service button*** , the first patient is deleted and the next patient will be called out.

