K L UNIVERSITY

FRESHMAN ENGINEERING DEPARTMENT

A Project Based Lab Report

On

AVL Tree Application

SUBMITTED BY:

2010030071 Jyothin Movva

2010030113 N. Hemanth Srivathsav

2010030493 Vishaladitya Valluru

2010030124 P. Vivek Vardhan Reddy

UNDER THE ESTEEMED GUIDANCE OF

Dr. G. Rekha

HOD



KL UNIVERSITY

Green fields, Vaddeswaram – 522 502

Guntur Dt., AP, India.

DEPARTMENT OF BASIC ENGINEERING SCIENCES



CERTIFICATE

This is to certify that the project based laboratory report entitled “AVL Tree Application ” submitted by **Mr.Jyothin Movva(2010030071),Mr.N. Hemanth Srivathsav (2010030113), Mr.Vishaladitya Valluru(2010030493), Mr. P. Vivek Vardhan Reddy( 2010030124)** and bearing Regd. No. <REGD.NO> to the Department of Basic Engineering Sciences, KL University in partial fulfillment of the requirements for the completion of a project in “Object Oriented Programming-19CS1203” course in I B Tech II Semester, is a bonafide record of the work carried out by him/her under my supervision during the academic year 2020-21.

PROJECT SUPERVISOR HEAD OF THE DEPARTMENT

< GUIDE NAME> Dr. G. REKHA

ACKNOWLEDGEMENTS

The satisfaction that accompanies the successful completion of any task would be incomplete without mention of the people who made it possible and whose encouragement and guidance has been a source of inspiration throughout the course of the project. We express our sincere gratitude to all our Professors of Computer Science & Engineering, K L University, Hyderabad, for their precious suggestions, motivation, and cooperation for the successful completion of this project. It is our privilege.

I express the sincere gratitude to our Principal Dr. L. Koteswara Rao for his administration towards our academic growth.

I express sincere gratitude to our Coordinator and HOD-BES Dr. G. Rekha for her leadership and constant motivation provided in successful completion of our academic semester. I record it as my privilege to deeply thank for providing us the efficient faculty and facilities to make our ideas into reality.

Finally, it is pleased to acknowledge the indebtedness to all those who devoted themselves directly or indirectly to make this project report success.

Regd. No: Name:

2010030071 (Jyothin Movva)

2010030113 (N. Hemanth Srivathsav)

2010030124 (Vishaladitya Valluru)

2010030493 (P. Vivek Vardhan Reddy)

ABSTRACT

The project involves AVL tree implementation which uses Strings as data. This application involves using an AVL tree to arrange the data using single rotations and double rotations. The application has 3 types of transversal functions which we can access the data with. The traversal methods are in-order, pre-order and post-order.

The application is taking advantage of graphical user interface to take the data as input and output the data. It is also using file system to store the inputted data for future use outside the application. The application also the counts the number of occurrences.

The use of graphical user interface makes the application user-friendly and easy to use. The application is lightweight and can virtually work on any system with java idk 8+.

INDEX

|  |  |  |
| --- | --- | --- |
| S.NO | TITLE | PAGE NO |
| 1 | Abstract | 3 |
| 2 | Introduction | 5 |
| 3 | AIM | 10 |
| 4 | System Requirements | 11 |
| 5 | Class Diagram | 12 |
| 5 | Implementation | 13 |
| 6 | Output | 20 |
| 7 | Conclusion | 23 |
|  |  |  |
|  |  |  |

INTRODUCTION

Traditionally, phone directory is telephone book, telephone address book, phone book, or the white and yellow pages, is a listing of telephone subscribers in a geographical area or subscribers to services provided by the organization that publishes the directory. Its purpose is to allow the telephone number of a subscriber identified by name and address to be found.

This project is an modern implimentation of the core phone directory which uses java file class and impliments search function for more feature rich phone book. The project also impliments gui which make the features more user friendly and easy to use.

The Project:

Raj has a phone directory in which he will store his friends and family’s phone numbers. He wants to transfer all the information into a file and retrieve the details in an easy manner. The directory contains the phone numbers and name of the person. He wants to add a new person's details with phone number. So, help him by writing a Java program which retrieves the details by searching/adding the person's name and vice versa. Use GUI

AIM:

To create an gui application of the concept phone directory and add functionality that will help us with storing a new contact, searching an saved contact with giving the name and displaying the stored contacts

To impliment files concept to store and retreive data and display and use the data to help the program perfom its functionaliy.

SYSTEM REQUIREMENTS

* SOFTWARE REQUIREMENTS:

The major software requirements of the project are as follows:

Language : JAVA

Operating system: Windows XP or later.

Tools: JDK , Visual Studio Code.

* HARDWARE REQUIREMENTS:

The hardware requirements that map towards the software are as follows:

Ram : RAM 1GB+

Processor: INTEL i3+

CLASS DIAGRAM



IMPLEMENTATION

ALGORITHM

Initialise String number,name

if(new contact)

load/create file and enter the scanned data.

if(search)

call nameWithNumeber() method to load the file and retrieve the data

if(display)

display the display buttons and call readlife() method with appropriate string path

Display options – DISPLAY ALL, DISPLAY FRIENDS, DISPLAY FAMILY AND DISPLAY WORK

If(TERMINATE)

system.exit

PROGRAM

package project;

import java.awt.event.\*;

import javax.swing.\*;

import java.awt.\*;

import java.io.\*;

import java.util.\*;

import java.lang.\*;

class phoneD{

    static JLabel title = new JLabel("PHONE DIRECTORY");

    static JFrame mainframe = new JFrame("PHONE");

    static JFrame searchframe = new JFrame("SEARCH");

    static JFrame newcontactframe = new JFrame("CONTACT");

    static JFrame displayframe = new JFrame("DISPLAY");

    static JButton newContact=new JButton("NEW CONTACT");

    static JButton search=new JButton("SEARCH");

    static JButton display=new JButton("DISPLAY");

    static JButton terminate=new JButton("TERMINATE");

    static JButton save = new JButton("SAVE");

    static JButton back = new JButton("BACK");

    static JButton back2 = new JButton("BACK");

    static JButton back3 = new JButton("BACK");

    static JButton startsearchName=new JButton("SEARCH");

    static JButton displayfam=new JButton("DISPLAY FAMILY");

    static JButton displayfriend=new JButton("DISPLAY FRIENDS");

    static JButton displaywork=new JButton("DISPLAY WORK");

    static JButton displayall=new JButton("DISPLAY ALL");

    static JTextArea textarea = new JTextArea();

    static JTextField nametextarea = new JTextField();

    static JTextField numbertextarea = new JTextField();

    static JTextField emailtextarea = new JTextField();

    static JTextField searchbytextarea = new JTextField();

    static JTextField searchbynumtextarea = new JTextField();

    static JLabel entername = new JLabel("ENTER NAME");

    static JLabel eentername = new JLabel("ENTER NAME");

    static JLabel enteremail = new JLabel("ENTER E-MAIL");

    static JLabel enternumber = new JLabel("ENTER NUMBER");

    static JLabel selectgroup = new JLabel("SELECT GROUP");

    static String number = null;

    static String name = null;

    static void readFile(String path) {

        try {

            FileReader nameReader = new FileReader(path);

            BufferedReader namebufferedReader = new BufferedReader(nameReader);

            String nameline;

            while ((nameline = namebufferedReader.readLine()) != null){

                textarea.append(nameline);

                textarea.append("\n");

            }

            nameReader.close();

        }catch (IOException e){

            e.printStackTrace();

        }

    }

    static void feedFile(){

        String number = numbertextarea.getText();

        String name = nametextarea.getText();

        String email = emailtextarea.getText();

        try{

            FileWriter nWriter = new FileWriter("namesnumbers.txt", true);

            nWriter.append(name+" "+number+" "+email);

            nWriter.append("\n");

            nWriter.close();

        } catch (IOException e) {

            e.printStackTrace();

        }

    }

    static void feedFile2(String fileName){

        String number = numbertextarea.getText();

        String name = nametextarea.getText();

        String email = emailtextarea.getText();

        try {

            FileWriter nWriter = new FileWriter(fileName, true);

            nWriter.append(name+" "+number+" "+email);

            nWriter.append("\n");

            nWriter.close();

        } catch (IOException e) {

            e.printStackTrace();

        }

    }

    static void nameFinderWithNumber(String path,String searchPath){

        String key = searchbytextarea.getText();

        try{

            File namefile = new File("namesnumbers.txt");

            Scanner nameSc = new Scanner(namefile);

            String name,number,email;

            textarea.setText("");

            while(nameSc.hasNextLine()) {;

                StringTokenizer st = new StringTokenizer(nameSc.nextLine()," ");

                while (st.hasMoreTokens()){

                       name = st.nextToken();

                       number = st.nextToken();

                       email = st.nextToken();

                       System.out.println(name+number);

                       if(name.equals(key))

                       textarea.append("Name->"+name+"\n"+"Number->"+number+"\nEmail->"+email+"\n\n");

                       System.out.println("pointing to->"+name+"//"+number+"//"+email);

                }

            }

            nameSc.close();

          }

          catch(Exception e){

            e.getStackTrace();

          }

    }

    public static void main(String args[]){

        String group[] = {"NONE","FAMILY","FRIENDS","WORK"};

        final JComboBox groupbox = new JComboBox(group);

        title.setFont(new Font("Arial", Font.BOLD, 16));

        title.setSize(300, 20);

        title.setLocation(115, 30);

        entername.setFont(new Font("Arial", Font.ITALIC, 16));

        entername.setSize(300, 20);

        entername.setLocation(100,50);

        enternumber.setFont(new Font("Arial", Font.ITALIC, 16));

        enternumber.setSize(300, 20);

        enternumber.setLocation(100, 127);

        enteremail.setFont(new Font("Arial", Font.ITALIC, 16));

        enteremail.setSize(300, 20);

        enteremail.setLocation(100,214);

        selectgroup.setFont(new Font("Arial", Font.ITALIC, 16));

        selectgroup.setSize(300, 20);

        selectgroup.setLocation(100,300);

        eentername.setFont(new Font("Arial", Font.ITALIC, 16));

        eentername.setSize(300, 20);

        eentername.setLocation(100,50);

        textarea.setBounds(100,60,200,200);

        nametextarea.setBounds(100,75,200,50);

        numbertextarea.setBounds(100,150,200,50);

        emailtextarea.setBounds(100,240,200,50);

        newContact.setBounds(100,300,200,30);

        search.setBounds(100,350,200,30);

        save.setBounds(100,400,200,30);

        back.setBounds(100,600,200,30);

        back2.setBounds(100,600,200,30);

        back3.setBounds(100,600,200,30);

        display.setBounds(100,400,200,30);

        terminate.setBounds(100,600,200,30);

        searchbytextarea.setBounds(100,100,200,50);

        searchbynumtextarea.setBounds(100,100,200,50);

        startsearchName.setBounds(100,350,200,30);

        groupbox.setBounds(100,330,200,30);

        displayall.setBounds(100,250,200,30);

        displayfam.setBounds(100,300,200,30);

        displayfriend.setBounds(100,350,200,30);

        displaywork.setBounds(100,400,200,30);

        mainframe.add(title);

        mainframe.add(textarea);

        mainframe.add(newContact);

        mainframe.add(search);

        mainframe.add(display);

        mainframe.add(terminate);

        mainframe.setSize(400,700);

        mainframe.setLayout(null);

        mainframe.setVisible(true);

        mainframe.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        newcontactframe.setSize(400,700);

        newcontactframe.add(nametextarea);

        newcontactframe.add(numbertextarea);

        newcontactframe.add(emailtextarea);

        newcontactframe.add(save);

        newcontactframe.add(back2);

        newcontactframe.add(entername);

        newcontactframe.add(enternumber);

        newcontactframe.add(enteremail);

        newcontactframe.add(groupbox);

        newcontactframe.add(selectgroup);

        newcontactframe.setLayout(null);

        newcontactframe.setVisible(false);

        newcontactframe.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        searchframe.setSize(400,700);

        searchframe.add(back3);

        searchframe.add(eentername);

        searchframe.add(searchbytextarea);

        searchframe.add(startsearchName);

        searchframe.setLayout(null);

        searchframe.setVisible(false);

        searchframe.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        displayframe.add(displayfam);

        displayframe.add(displayfriend);

        displayframe.add(displaywork);

        displayframe.add(displayall);

        displayframe.add(back);

        displayframe.setSize(400,700);

        displayframe.setLayout(null);

        displayframe.setVisible(false);

        displayframe.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        terminate.addActionListener(new ActionListener(){

            public void actionPerformed(ActionEvent e){

                System.exit(1);

            }

            });

        newContact.addActionListener(new ActionListener(){

            public void actionPerformed(ActionEvent e){

                mainframe.setVisible(false);

                searchframe.setVisible(false);

                newcontactframe.setVisible(true);

                displayframe.setVisible(false);

                }

                    });

        back.addActionListener(new ActionListener(){

            public void actionPerformed(ActionEvent e){

                mainframe.setVisible(true);

                searchframe.setVisible(false);

                newcontactframe.setVisible(false);

                displayframe.setVisible(false);

                    }

                    });

        back2.addActionListener(new ActionListener(){

            public void actionPerformed(ActionEvent e){

                mainframe.setVisible(true);

                searchframe.setVisible(false);

                newcontactframe.setVisible(false);

                    }

                    });

        back3.addActionListener(new ActionListener(){

            public void actionPerformed(ActionEvent e){

                mainframe.setVisible(true);

                searchframe.setVisible(false);

                newcontactframe.setVisible(false);

                displayframe.setVisible(false);

                    }

                    });

         save.addActionListener(new ActionListener(){

            public void actionPerformed(ActionEvent e){

                    feedFile();

                    mainframe.setVisible(true);

                    searchframe.setVisible(false);

                    newcontactframe.setVisible(false);

                    displayframe.setVisible(false);

                    if(groupbox.getItemAt(groupbox.getSelectedIndex()).equals("FAMILY"))

                    feedFile2("family.txt");

                    else if(groupbox.getItemAt(groupbox.getSelectedIndex()).equals("FRIENDS"))

                    feedFile2("friends.txt");

                    else if(groupbox.getItemAt(groupbox.getSelectedIndex()).equals("WORK"))

                    feedFile2("work.txt");

                    else

                    System.out.println(groupbox.getItemAt(groupbox.getSelectedIndex()));

                    }

                    });

        search.addActionListener(new ActionListener(){

            public void actionPerformed(ActionEvent e){

                mainframe.setVisible(false);

                searchframe.setVisible(true);

                newcontactframe.setVisible(false);

                displayframe.setVisible(false);

                    }

                    });

        startsearchName.addActionListener(new ActionListener(){

            public void actionPerformed(ActionEvent e){

                mainframe.setVisible(true);

                searchframe.setVisible(false);

                newcontactframe.setVisible(false);

                displayframe.setVisible(false);

                nameFinderWithNumber("names.txt","numbers.txt");

                    }

                    });

        display.addActionListener(new ActionListener(){

            public void actionPerformed(ActionEvent e){

                mainframe.setVisible(false);

                searchframe.setVisible(false);

                newcontactframe.setVisible(false);

                displayframe.setVisible(true);

                textarea.setText("");

                }

                });

        displayall.addActionListener(new ActionListener(){

            public void actionPerformed(ActionEvent e){

                mainframe.setVisible(true);

                searchframe.setVisible(false);

                newcontactframe.setVisible(false);

                displayframe.setVisible(false);

                textarea.setText("");

                readFile("namesnumbers.txt");

                }

                });

        displayfam.addActionListener(new ActionListener(){

            public void actionPerformed(ActionEvent e){

                mainframe.setVisible(true);

                searchframe.setVisible(false);

                newcontactframe.setVisible(false);

                displayframe.setVisible(false);

                textarea.setText("");

                readFile("family.txt");

                }

                });

        displayfriend.addActionListener(new ActionListener(){

            public void actionPerformed(ActionEvent e){

                mainframe.setVisible(true);

                searchframe.setVisible(false);

                newcontactframe.setVisible(false);

                displayframe.setVisible(false);

                textarea.setText("");

                readFile("friends.txt");

                }

                });

        displaywork.addActionListener(new ActionListener(){

            public void actionPerformed(ActionEvent e){

                mainframe.setVisible(true);

                searchframe.setVisible(false);

                newcontactframe.setVisible(false);

                displayframe.setVisible(false);

                textarea.setText("");

                readFile("work.txt");

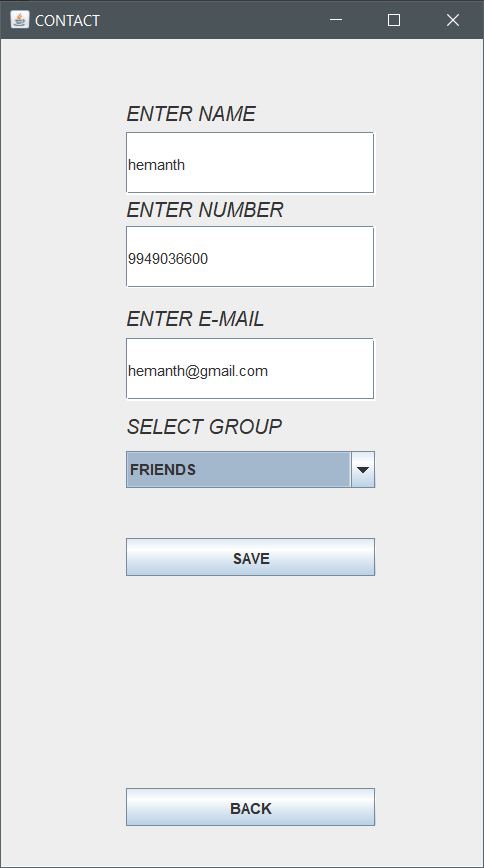
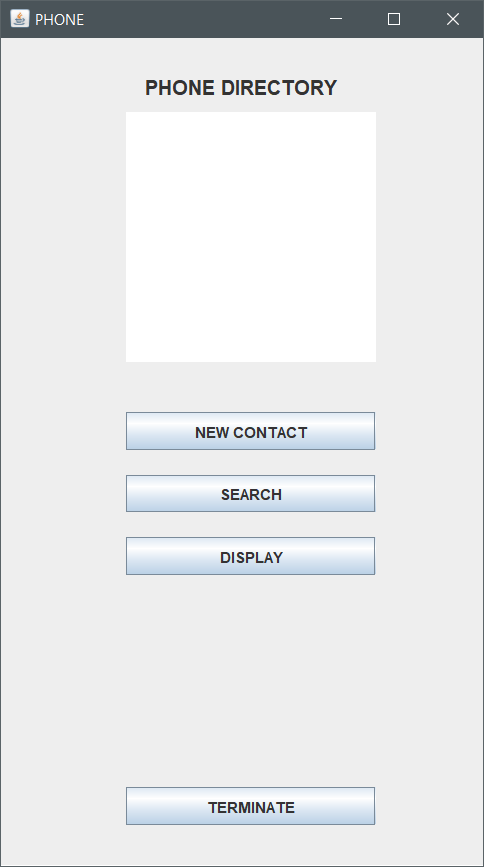
                }

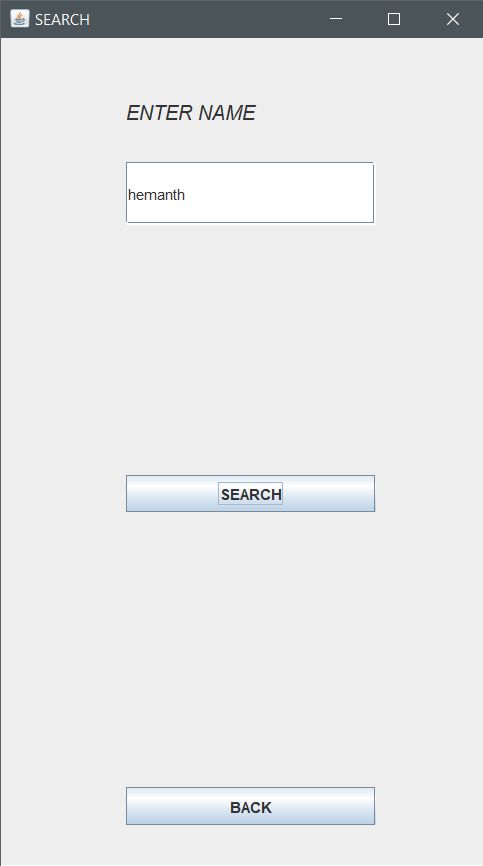
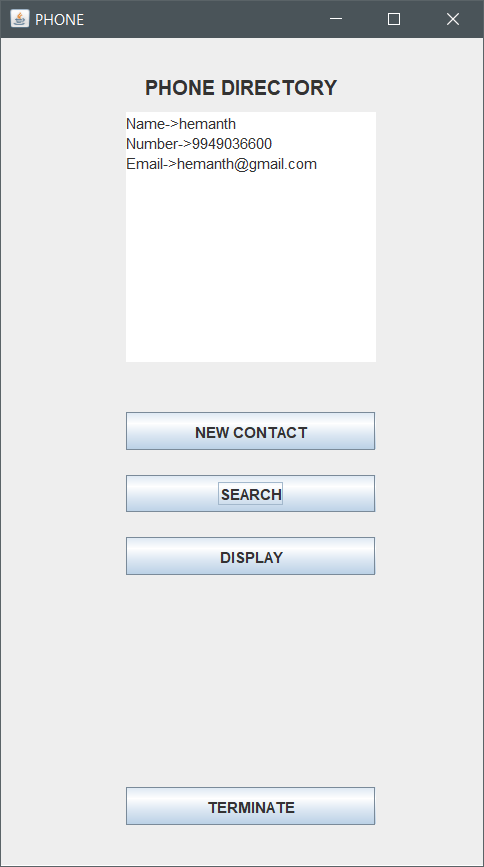
                });

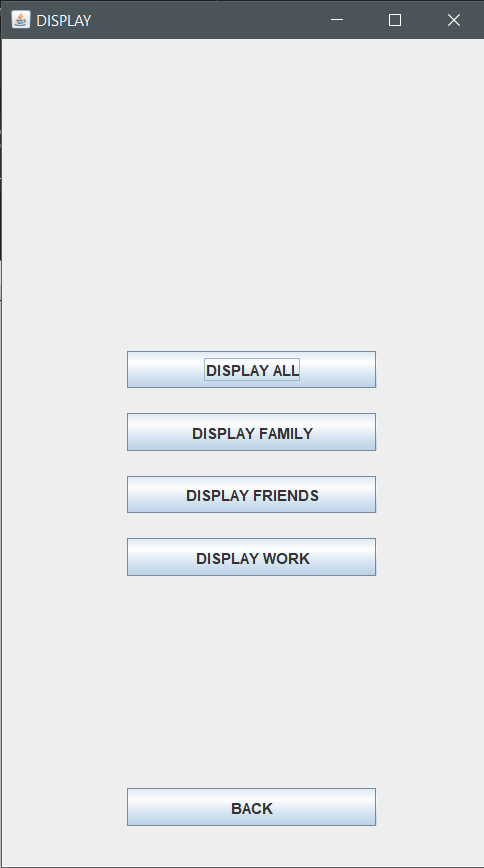
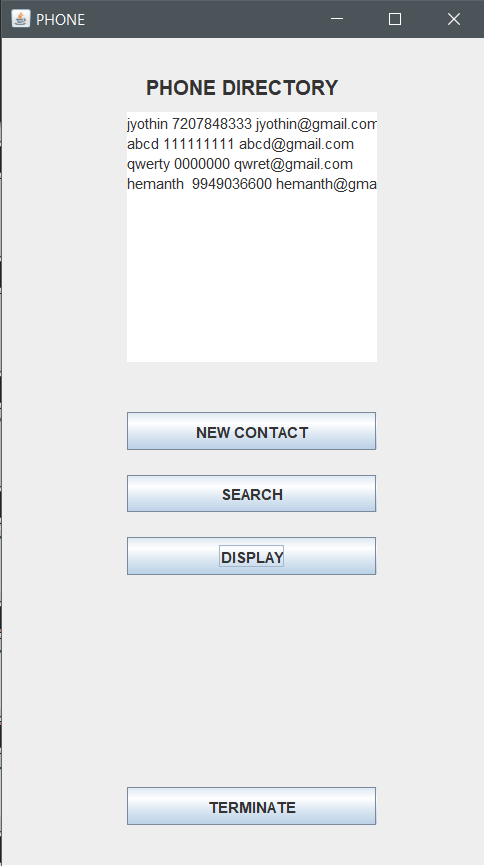
   }

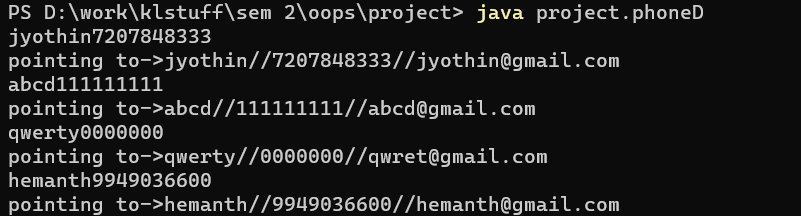
}

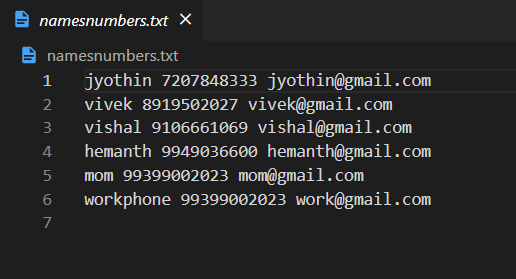
OUTPUTS

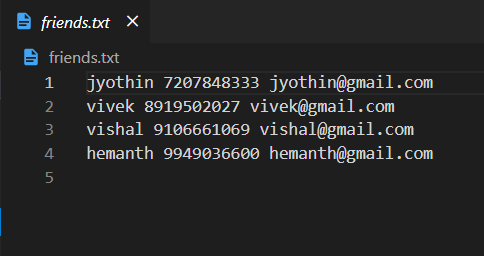


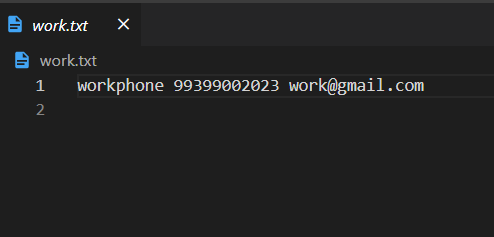


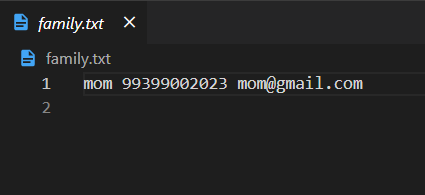












CONCLUSION

This project uses the concept of files and GUI to make the objective work. This program is managing to collect data from the user and store it in a file. It also helps us search the data by name of the contact or assigned groups.