

# **ADDRESS DIARY**



An

Object-Oriented Programming through Java Course Project Report

in partial fulfilment of the degree

## **Bachelor of Technology in Computer Science & Engineering**

By

N.BHARATH CHANDRA

HTNo (2003A52097)

CH.AKSHAY RAO

HTNo (2103A52L01)

Submitted to

**Mr. Bala Krishna**

Assistant Proffessor



**SR**  
UNIVERSITY



## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

### **CERTIFICATE**

This is to certify that the **Object Oriented Programming through Java - Course Project** Report entitled “**ADDRESS DIARY**” is a record of bonafide work carried out by the student **N.Bharath Chandra and CH.Akshay Rao** bearing Roll No(s) **2003A52097 and 2103A52L01** during the academic year 2021-2022 in partial fulfillment of the award of the degree of ***Bachelor of Technology*** in **Computer Science & Engineering** by the SR University, Warangal.

**Mr.Balakrishana**  
Assistant professor

**Head of the Department**

## **Table of Contents**

<b>Contents</b>	<b>Page.no</b>
1. OBJECTIVE OF THE PROJECT	05
2. DEFINITIONS OF THE ELEMENTS USED IN THE PROJECT	06
3. DESIGN	
3.1. SCREENS	07
4. IMPLEMENTATION	
4.1. CODE	09
5. RESULT SCREENS	28
6. CONCLUSION	32

## **ABSTRACT**

This is an Address Book application built by using Java's Native GUI Framework called Swing, using swing and other packages available in java we were able to build an application that people can use to save their contacts in a Graphical User Interface that is not hard to understand but is so intuitive that even a 5 year old can use it to save and retrieve contacts, we also had to use java file stream api's to allow the data to be stored inside the client instead of using a dbms system so that we don't need to have internet access to use our app. The data stored in the files is stored according to Java's format and so normal people can't access the data. User defined exception is thrown if the user tries to add an already allotted contact. Exception handling is properly done to deal with any kind of unexpected exception.

## **1.OBJECTIVE OF THE PROJECT:**

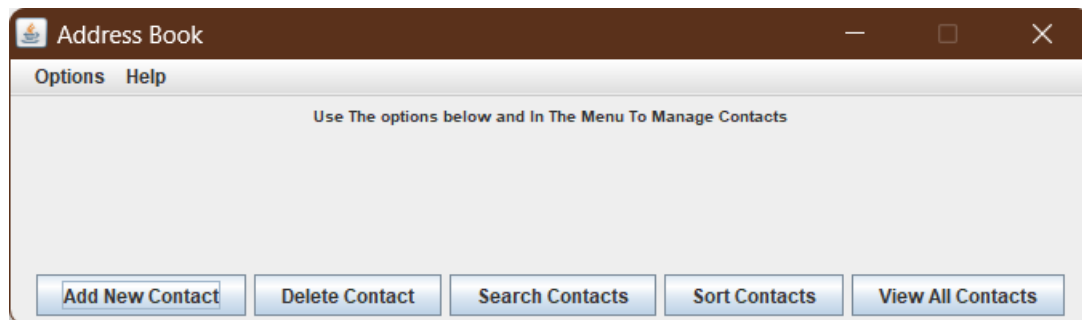
The main goal of this project is to create app for use in a user's address books. The system should be as adaptable as possible, allowing it to be used in a variety of people. To learn about the various contacts that people have used. We need to figure out UI is better to implement, and then build an app that will help in achieving that goal. We need to look up how an Address book is built, use our own knowledge, or enquire on internet forums. The more varied your information sources are, the better the resulting method and, potentially, your details are available close to you. This project aims to make record updating, maintenance, and searching more user-friendly. will be able to retrieve the correct information from the files..

## **2.DEFINITIONS OF THE ELEMENTS USED IN THE PROJECT:**

1. **Classes & Objects :** : A Class is like an object constructor, or a "blueprint" for creating objects. An object is called an instance of a class
2. **Inheritance :** Inheritance in Java is a concept that acquires the properties from one class to other classes.
3. **File handling with objects :** File handling in Java is defined as reading and writing data to a file. The particular file class from the package called java.io allows us to handle and work with different formats of files.
4. **Swing:** Swing is a framework used to implement Native User Interfaces for use in Desktop Environments, this tool has many features and was built using the awt package from java, so this inherits many of those properties. **Implementing Interface :** The implements keyword is used to implement an interface . The interface keyword is used to declare a special type of class that only contains abstract methods. To access the interface methods, the interface must be "implemented" by another class with the implements keyword .
5. **User Defined Exception and Exception handling :** we can create our own exceptions that are derived classes of the Exception class. Creating our own Exception is known as custom exception or user-defined exception. Basically, Java custom exceptions are used to customize the exception according to user need. The Exception Handling in Java is one of the powerful mechanism *to* handle the runtime errors

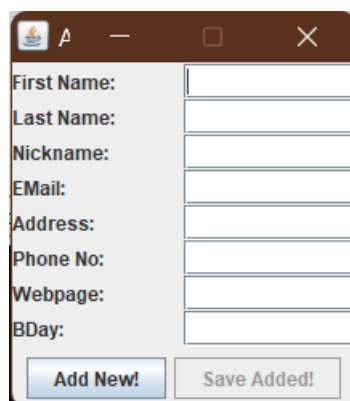
## 3.Design

### 3.1 Test Screens



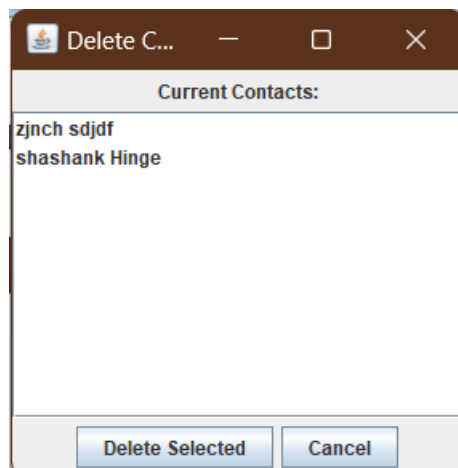
**fig 3.1.1**

The figure states the home screen which shows many options for what you can do to your address book.

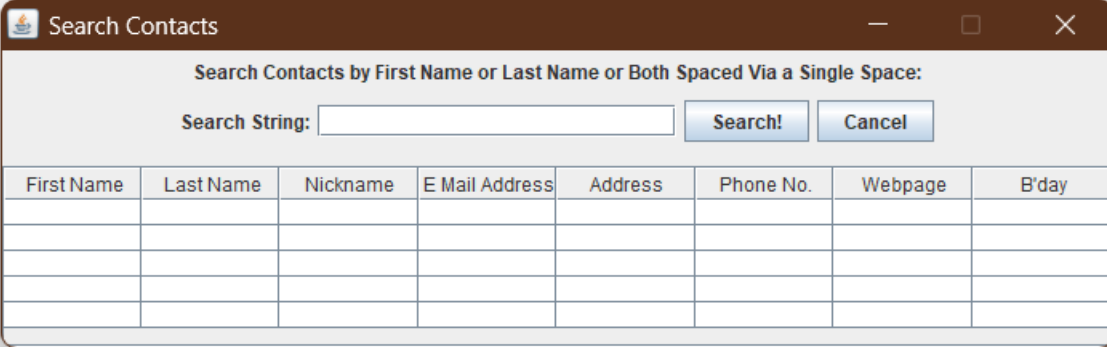


**fig 3.1.2**

The above window can be used to add a new non existing contact.



The above window can be used to delete a contact.



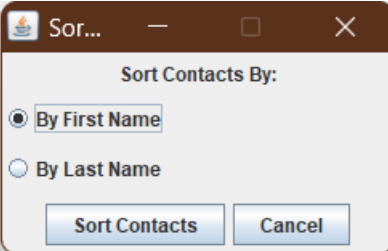
Search Contacts by First Name or Last Name or Both Spaced Via a Single Space:

Search String:

First Name	Last Name	Nickname	E Mail Address	Address	Phone No.	Webpage	B'day

fig 3.1.3

- This window is used to search for a contact in the address book



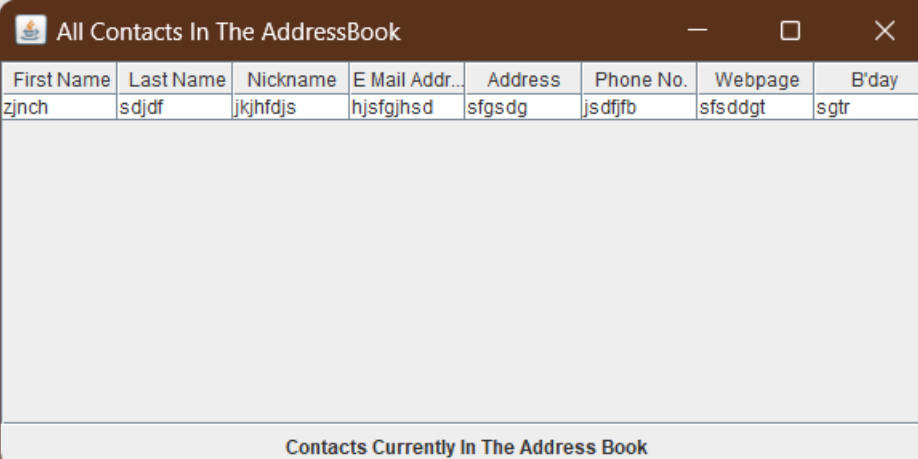
Sort Contacts By:

☒ By First Name

☐ By Last Name

fig 3.1.4

- This window is used to sort the contacts based on their last or first names.



First Name	Last Name	Nickname	E Mail Addr...	Address	Phone No.	Webpage	B'day
zjnc	sdjdf	jkjhfdjs	hjsfgjhds	sfgsdg	jsdfjfb	sfsddgt	sgtr

Contacts Currently In The Address Book

- This window shows all the contacts in the address book.



## 4.Implementation

### 4.1 Code

J NewClass.java 9+ ×

src > J NewClass.java > AddressBook > screenSize

```
1  import java.io.*;
2  import java.awt.*;
3  import java.awt.event.*;
4  import javax.swing.*;
5  import javax.swing.event.*;
6  import java.util.*;
7
8  class AddressBook implements ActionListener
9  {
10     JPanel topPanel,bottomPanel;
11     JScrollPane scrollPane;
12     JFrame frame;
13     JMenuBar menubar = new JMenuBar(); ;
14     JMenu menu = new JMenu();
15     JMenuItem menuItem;
16     Toolkit kit = Toolkit.getDefaultToolkit();
17     Dimension screenSize = kit.getScreenSize();
18     int screenHeight = screenSize.height;
19     int screenWidth = screenSize.width;
20     Image img = kit.getImage(filename: "images/icon.JPG");
21     AddressBook()
22     {
23         frame = new JFrame(title: "Address Book");
24         frame.setSize(width: 680,height: 200);
25         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
26         frame.setLocation(screenWidth/4, screenHeight/4);
27         frame.setIconImage(img);
28         addWidgets();
29         frame.show();
30     }
```

```
34  ✓ public void addWidgets()
35
36     {
37         menubar.add(menu);
38         menu = new JMenu("Options");
39         menuItem = new JMenuItem("Add New Contact");
40         menu.add(menuItem);
41         menuItem.addActionListener(this);
42         menuItem = new JMenuItem("Delete Contact");
43         menu.add(menuItem);
```

```

43     menu.add(menuItem);
44     menuItem.addActionListener(this);
45     menuItem = new JMenuItem("Search Contacts");
46     menu.add(menuItem);
47     menuItem.addActionListener(this);
48     menuItem = new JMenuItem("Sort Contacts");
49     menu.add(menuItem);
50     menuItem.addActionListener(this);
51     menuItem = new JMenuItem("View All Contacts");
52     menu.add(menuItem);
53     menuItem.addActionListener(this);
54     menuItem = new JMenuItem("Backup Contacts");
55     menu.add(menuItem);
56     menuItem.addActionListener(this);
57     menubar.add(menu);
58     menu = new JMenu("Help");
59     menuItem = new JMenuItem("Help Contents");
60     menu.add(menuItem);
61     menuItem.addActionListener(this);
62     menuItem = new JMenuItem("About");
63     menu.add(menuItem);
64     menuItem.addActionListener(this);
65     menubar.add(menu);
66     frame.setJMenuBar(menubar);
67     JPanel topPanel = new JPanel();
68     JPanel bottomPanel = new JPanel();
69     //Add Buttons To Bottom Panel
70     JButton AddNew = new JButton("Add New Contact");
71     JButton DeleteContact = new JButton("Delete Contact");
72     JButton SearchContacts = new JButton("Search Contacts");

```

```

73     JButton SortContacts = new JButton("Sort Contacts");
74     JButton ViewContactList = new JButton("View All Contacts");
75     JLabel label = new JLabel("<HTML><FONT FACE = COSMIC SANS SIZE = 2><B>Use  

76     The options below and In The Menu To Manage Contacts");
77     //Add Action Listeners
78     AddNew.addActionListener(this);
79     DeleteContact.addActionListener(this);
80     SearchContacts.addActionListener(this);
81     SortContacts.addActionListener(this);
82     ViewContactList.addActionListener(this);
83     topPanel.add(label);
84     bottomPanel.add(AddNew);
85     bottomPanel.add(DeleteContact);
86     bottomPanel.add(SearchContacts);
87     bottomPanel.add(SortContacts);
88     bottomPanel.add(ViewContactList);
89     frame.getContentPane().add(topPanel, BorderLayout.NORTH);
90     frame.getContentPane().add(bottomPanel, BorderLayout.SOUTH);
91     frame.setResizable(false);
92 }
93

```

```

94
Run | Debug
95 public static void main(String args[])
96 {
97     AddressBook add = new AddressBook();
98
99 }
100 OperationHandler oh = new OperationHandler();
101 public void actionPerformed(ActionEvent ae)
102 {
103     if(ae.getActionCommand() == "Add New Contact")
104     {
105         oh.AddNew();
106     }
107     else if(ae.getActionCommand() == "Search Contacts")
108     {
109         oh.SearchContacts();
110     }
111     else if(ae.getActionCommand() == "Sort Contacts")
112     {
113         oh.SortContacts();
114     }
115     else if(ae.getActionCommand() == "Delete Contact")
116     {
117         oh.DeleteContact();
118     }
119
120
121 }

```

```

122
123     else if(ae.getActionCommand() == "View All Contacts")
124     {
125
126         oh.ViewAllContacts();
127     }
128
129
130     else if(ae.getActionCommand() == "About")
131     {
132         JOptionPane.showMessageDialog(frame, "About Address Book:  

133         Created By Harpreet Arora Computer Engg, IET-DAVV, Indore.  

134         Website: http://www.harpreetarora.tk",
135         "About Address Book", JOptionPane.INFORMATION_MESSAGE);
136

```

```

137     else if(ae.getActionCommand() == "Help Contents")
138     {
139         try
140         {
141             oh.showHelp();
142         }
143         catch(IOException e)
144         {
145         }
146     }
147
148     else if(ae.getActionCommand() == "Backup Contacts")
149     {
150         JFileChooser chooser = new JFileChooser();
151         chooser.setCurrentDirectory(new File("."));
152         chooser.setMultiSelectionEnabled(false);
153         chooser.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);
154         chooser.showSaveDialog(frame);
155         FileOutputStream bfout = null;
156         FileInputStream bfin = null;
157         String filename=null;
158         int p;
159         try
160         {
161             filename = chooser.getSelectedFile().getPath();
162         }
163         catch(Exception e)
164         {
165         }

```

```

---
166         try
167         {
168             bfout = new FileOutputStream(filename + "data.dat");
169         }
170         catch(Exception e)
171         {
172         }
173     }
174
175     try
176     {
177         bfin = new FileInputStream("data.dat");
178     }
179     catch(Exception e)
180     {
181     }
182
183
184     try
185     {
186         do
187         {
188             p = bfin.read();
189             if(p!=-1)
190                 bfout.write(p);
191             }while(p!=-1);

```

```

191         }
192         catch(Exception e)
193         {
194
195         }
196
197     }
198
199 }
200
201
202
203 }
204
205

```

```

206 class Contact implements Serializable
207 {
208     private String FName;
209     private String LName;
210     private String Nname;
211     private String EMail;
212     private String Address;
213     private String PhoneNo;
214     private String Webpage;
215     private String Bday;
216

```

```

216
217     public void setDetails(String fname, String lname, String nname,
218 String email, String address, String phone, String web, String bday)
219     {
220         FName = fname;
221         LName = lname;
222         Nname = nname;
223         EMail = email;
224         Address = address;
225         PhoneNo = phone;
226         Webpage = web;
227         Bday = bday;
228     }
229
230
231     public String getFName()
232     {
233         return FName;
234     }
235

```

```

235
236     public String getLName()
237     {
238         |     return LName;
239     }
240
241     public String getNname()
242     {
243         |     return Nname;
244     }
245
246     public String getEmail()
247     {
248         |     return EMail;
249     }
250
251     public String getAddress()
252     {
253         |     return Address;
254     }
255
256     public String getPhoneNo()
257     {
258         |     return PhoneNo;
259     }
260

```

```

260
261     public String getWebpage()
262     {
263         |     return Webpage;
264     }
265
266     public String getBday()
267     {
268         |     return Bday;
269     }
270
271
272 }
273

```

```

275 class OperationHandler implements ListSelectionListener, ActionListener, Runnable
276 {
277     JFrame newFrame;
278     JTextField txtFirstName;
279     JTextField txtLastName;
280     JTextField txtNickname;
281     JTextField txtEmail;
282     JTextField txtAddress;
283     JTextField txtPhoneNo;
284     JTextField txtWebpage;
285     JTextField txtB'Day;
286     JButton BtnSaveAdded;
287
288     Vector v = new Vector(initialCapacity: 10, capacityIncrement: 3);
289     int i=0, k=0;
290     Toolkit kit = Toolkit.getDefaultToolkit();
291     Dimension screenSize = kit.getScreenSize();
292     int screenHeight = screenSize.height;
293     int screenWidth = screenSize.width;
294
295     Image img = kit.getImage(filename: "images/icon.JPG");
296
297     FileInputStream fis;
298     ObjectInputStream ois;
299
300     JList list;
301     DefaultListModel listModel;
302     ListSelectionModel listSelectionModel;
303
304     JRadioButton byfname, bylname;
305
306     Thread t;
307
308     JTable searchTable;
309
310     JTextField txtSearch;
311
312     String columnNames[] = { "First Name",
313                               "Last Name",
314                               "Nickname",
315                               "E Mail Address",
316                               "Address",
317                               "Phone No.",
318                               "Webpage",
319                               "B'day"
320                               };
321
322     Object data[][]= new Object[5][8];
323

```

```

324 ✓      OperationHandler()
325          {
326
327 ✓          try {
328              fis = new FileInputStream(name: "data.dat");
329              ois = new ObjectInputStream(fis);
330              v = (Vector) ois.readObject();
331              ois.close();
332          }
333
334          catch(Exception e)
335 ✓          {
336
337          }
338
339      }
340
341
342 ✓      public void run()
343          {
344
345          try
346 ✓          {
347              FileOutputStream fos = new FileOutputStream(name: "data.dat");
348              ObjectOutputStream oos = new ObjectOutputStream(fos);
349              oos.writeObject(v);
350              oos.flush();
351              oos.close();
352
353          }

```

```

352          }
353      }
354      catch(Exception e)
355      {
356          JOptionPane.showMessageDialog(newFrame,
357              message: "Error Opening Data File: Could Not Save Contents.",
358              title: "Error Opening Data File",JOptionPane.INFORMATION_MESSAGE);
359      }
360
361  }

```



```

362     public void AddNew()
363     {
364         newFrame = new JFrame(title: "Add New");
365         newFrame.setSize(width: 220, height: 250);
366         newFrame.setResizable(resizable: false);
367         newFrame.setIconImage(img);
368         JLabel lblFirstName = new JLabel(text: "First Name: ");
369         JLabel lblLastName = new JLabel(text: "Last Name: ");
370         JLabel lblNickname = new JLabel(text: "Nickname: ");
371         JLabel lblEmail = new JLabel(text: "EMail: ");
372         JLabel lblAddress = new JLabel(text: "Address: ");
373         JLabel lblPhoneNo = new JLabel(text: "Phone No: ");
374         JLabel lblWebpage = new JLabel(text: "Webpage: ");
375         JLabel lblBDay = new JLabel(text: "BDay: ");
376         JLabel lblEmpty1 = new JLabel(text: "");
377         JLabel lblEmpty2 = new JLabel(text: "");
378
379         txtFirstName = new JTextField(columns: 10);
380         txtLastName = new JTextField(columns: 10);
381         txtNickname = new JTextField(columns: 10);
382         txtEmail = new JTextField(columns: 10);
383         txtAddress = new JTextField(columns: 10);
384         txtPhoneNo = new JTextField(columns: 10);
385         txtWebpage = new JTextField(columns: 10);
386         txtBDay = new JTextField(columns: 10);
387
388         JButton BtnAdd = new JButton(text: "Add New!");
389         BtnSaveAdded = new JButton(text: "Save Added!");
390
391
392
393
394
395
396         JPanel centerPane = new JPanel();
397         JPanel bottomPane = new JPanel();
398

```

```

398
399         centerPane.add(lblFirstName);
400         centerPane.add(txtFirstName);
401         centerPane.add(lblLastName);
402         centerPane.add(txtLastName);
403         centerPane.add(lblNickname);
404         centerPane.add(txtNickname);
405         centerPane.add(lblEmail);
406         centerPane.add(txtEmail);
407         centerPane.add(lblAddress);
408         centerPane.add(txtAddress);
409         centerPane.add(lblPhoneNo);
410         centerPane.add(txtPhoneNo);
411         centerPane.add(lblWebpage);
412         centerPane.add(txtWebpage);
413         centerPane.add(lblBDay);
414         centerPane.add(txtBDay);
415         bottomPane.add(BtnAdd);
416         bottomPane.add(BtnSaveAdded);
417
418         centerPane.setLayout(new GridLayout(rows: 0, cols: 2));
419
420
421
422     newFrame.getContentPane().add(centerPane, BorderLayout.CENTER);
423
424     newFrame.getContentPane().add(bottomPane, BorderLayout.SOUTH);
425     newFrame.setLocation(screenWidth/4, screenHeight/4);
426     newFrame.show();

```

---

```

426         newFrame.show();
427
428     }
429
430

```

```

431     public void SearchContacts()
432     {
433         newFrame = new JFrame("Search Contacts");
434         newFrame.setSize(700,220);
435         newFrame.setLocation(screenWidth/4, screenHeight/4);
436         newFrame.setIconImage(img);
437         newFrame.setResizable(false);
438
439         JPanel topPane = new JPanel();
440         JLabel label1 = new JLabel("Search Contacts by First Name or
441         Last Name or Both Spaced Via a Single Space:");
442         topPane.add(label1);
443
444         JPanel centerPane = new JPanel();
445         JLabel label2 = new JLabel("Search String:");
446         txtSearch = new JTextField(20);
447         JButton btnSearch = new JButton("Search!");
448         btnSearch.addActionListener(this);
449         JButton btnCancel = new JButton("Cancel");
450         btnCancel.addActionListener(this);
451         centerPane.add(label2);
452         centerPane.add(txtSearch);
453         centerPane.add(btnSearch);
454         centerPane.add(btnCancel);
455
456         searchTable = new JTable(data,columnNames);
457         JScrollPane scrollPane = new JScrollPane(searchTable);
458         searchTable.setPreferredScrollableViewportSize(new Dimension(500, 90));
459

```

```

459
460     newFrame.getContentPane().add(scrollPane,BorderLayout.SOUTH);
461
462     newFrame.getContentPane().add(topPane, BorderLayout.NORTH);
463     newFrame.getContentPane().add(centerPane,BorderLayout.CENTER);
464     newFrame.show();
465 }
466
467

```

```

467
468     public void SortContacts()
469     {
470         newFrame = new JFrame(title: "Sort Contacts");
471         newFrame.setSize(width: 250,height: 160);
472         newFrame.setLocation(screenWidth/4, screenHeight/4);
473         newFrame.setIconImage(img);
474         newFrame.setResizable(resizable: false);
475
476         byfname = new JRadioButton(text: "By First Name");
477         byfname.setActionCommand(actionCommand: "By First Name");
478         byfname.setSelected(b: true);
479
480         bylname = new JRadioButton(text: "By Last Name");
481         bylname.setActionCommand(actionCommand: "By Last Name");

```

```

481 bylname.setActionCommand(actionCommand: "By Last Name");
482
483 ButtonGroup group = new ButtonGroup();
484 group.add(byfname);
485 group.add(bylname);
486
487 JPanel topPane = new JPanel();
488 JLabel label = new JLabel(text: "Sort Contacts By:");
489 topPane.add(label);
490
491 JPanel pane = new JPanel(new GridLayout(rows: 0,cols: 1));
492 pane.add(byfname);
493 pane.add(bylname);
494
495 JPanel bottomPane = new JPanel();
496 JButton sortBttn = new JButton(text: "Sort Contacts");
497 JButton cancelBttn = new JButton(text: "Cancel");
498 bottomPane.add(sortBttn);
499 bottomPane.add(cancelBttn);
500 sortBttn.addActionListener(this);
501 cancelBttn.addActionListener(this);
502
503 newFrame.getContentPane().add(topPane,BorderLayout.NORTH);
504 newFrame.getContentPane().add(pane, BorderLayout.CENTER);
505 newFrame.getContentPane().add(bottomPane,BorderLayout.SOUTH);
506
507 newFrame.show();
508
509 }

```

```

510
511
512 ✓ public void DeleteContact()
513 {
514     String fname, lname;
515     newFrame = new JFrame(title: "Delete Contact");
516     newFrame.setSize(width: 300,height: 300);
517     newFrame.setLocation(screenWidth/4, screenHeight/4);
518     newFrame.setIconImage(img);
519
520     JPanel centerPane = new JPanel();
521     listModel = new DefaultListModel();
522
523     Contact contact = new Contact();
524
525     for(int l=0;l<v.size();l++)
526     {
527         contact = (Contact) v.elementAt(l);
528
529         fname = contact.getFName();
530         lname = contact.getLName();
531         listModel.addElement(fname + " " + lname);
532     }
533
534

```

```

535
536         list = new JList(listModel);
537
538     list.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
539     listSelectionModel = list.getSelectionModel();
540     listSelectionModel.addListSelectionListener(this);
541
542     JScrollPane listScrollPane = new JScrollPane(list);
543
544     JPanel topPane = new JPanel();
545     JLabel label = new JLabel(text: "Current Contacts:");
546     topPane.add(label);
547
548     JPanel bottomPane = new JPanel();
549
550     JButton btnDelete = new JButton(text: "Delete Selected");
551     bottomPane.add(btnDelete);
552     btnDelete.addActionListener(this);
553
554     JButton btnCancel = new JButton(text: "Cancel");
555     bottomPane.add(btnCancel);
556     btnCancel.addActionListener(this);
557
558     newFrame.getContentPane().add(topPane, BorderLayout.NORTH);
559     newFrame.getContentPane().add(listScrollPane, BorderLayout.CENTER);
560     newFrame.getContentPane().add(bottomPane, BorderLayout.SOUTH);
561
562     newFrame.show();

```

```

562     newFrame.show();
563
564 }
565
566
567 public void ViewAllContacts()
568 {
569
570     newFrame = new JFrame(title: "All Contacts In The AddressBook");
571     newFrame.setSize(width: 600, height: 300);
572     newFrame.setIconImage(img);
573
574     Contact con = new Contact();
575
576
577     String columnNames[] = { "First Name",
578                             "Last Name",
579                             "Nickname",
580                             "E Mail Address",
581                             "Address",
582                             "Phone No.",
583                             "Webpage",
584                             "B'day"
585     };
586
587
588
589     Object data[][]= new Object[v.size()][8];
590
591

```

```

592         for(int j=0;j<v.size();j++)
593         {
594
595             con = (Contact) v.elementAt(k);
596
597             data[j][0] = con.getFName();
598             data[j][1] = con.getLName();
599             data[j][2] = con.getNname();
600             data[j][3] = con.getEmail();
601             data[j][4] = con.getAddress();
602             data[j][5] = con.getPhoneNo();
603             data[j][6] = con.getWebpage();
604             data[j][7] = con.getBday();
605
606             k++;
607         }
608     }
609     k=0;
610
611     JTable abtable = new JTable(data,columnNames);
612     JScrollPane scrollPane = new JScrollPane(abtable);
613     abtable.setPreferredScrollableViewportSize(new Dimension(width: 500, height: 370));
614
615     JPanel pane = new JPanel();
616     JLabel label = new JLabel(text: "Contacts Currently In The Address Book");
617     pane.add(label);
618
619
620     newFrame.getContentPane().add(pane,BorderLayout.SOUTH);
621     newFrame.getContentPane().add(scrollPane,BorderLayout.CENTER);
622     newFrame.setLocation(screenWidth/4,screenHeight/4);
623     newFrame.show();
624
625
626
627
628     }
629
630
631     public void showHelp() throws IOException
632     {
633         String title = "Help Contents";
634         String data = "";
635         FileInputStream fishhelp = null;
636         int i;
637
638         newFrame = new JFrame(title);
639         newFrame.setSize(width: 401, height: 400);
640         newFrame.setResizable(resizable: false);
641         newFrame.setLocation(screenWidth/4, screenHeight/4);
642         newFrame.setIconImage(img);
643
644         JTextArea textArea = new JTextArea(rows: 5, columns: 20);
645         textArea.setLineWrap(wrap: true);
646         textArea.setEditable(b: false);
647

```

```

647
648         try
649         {
650             fishelp= new FileInputStream(name: "help/help.txt");
651         }
652         catch(Exception e)
653         {
654             JOptionPane.showMessageDialog(newFrame, message: "Help File Not Found.",
655             title: "Help File Not Found", JOptionPane.INFORMATION_MESSAGE);
656         }
657     do
658     {
659         i = fishelp.read();
660         if(i!=1)
661             data = data + (char) i;
662     } while(i!=-1);
663
664     fishelp.close();
665
666     textArea.setText(data);
667
668     JPanel bottomPane = new JPanel();
669     JButton button = new JButton(text: "Ok");
670     bottomPane.add(button);
671     button.addActionListener(this);
672
673     JPanel topPane = new JPanel();
674     JLabel label = new JLabel(title);
675     topPane.add(label);

```

```

675         topPane.add(label);
676
677         JScrollPane scrollPane = new JScrollPane(textArea);
678
679
680     newFrame.getContentPane().add(topPane,BorderLayout.NORTH);
681     newFrame.getContentPane().add(scrollPane);
682
683     newFrame.getContentPane().add(bottomPane,BorderLayout.SOUTH);
684
685     newFrame.show();
686
687 }
688
689
690
691 public void actionPerformed(ActionEvent ae)
692 {
693
694
695     if(ae.getActionCommand() == "Add New!")
696     {

```

```

696 {
697
698     if(txtFirstName.getText().equals(anObject: "") &&
699 txtLastName.getText().equals(anObject: "") && txtNickname.getText().equals(anObject: "") &&
700 txtEmail.getText().equals(anObject: "") && txtAddress.getText().equals(anObject: "") &&
701 txtPhoneNo.getText().equals(anObject: "") && txtWebpage.getText().equals(anObject: "") &&
702 txtBDay.getText().equals(anObject: ""))
703 {
704
705 JOptionPane.showMessageDialog(newFrame,
706 message: "Entries Empty! Fill in the required entries to save Contact.", title: "Entries Empty",
707 JOptionPane.INFORMATION_MESSAGE);
708 }
709
710 else
711 {
712     Contact contact = new Contact();
713
714 contact.setDetails(txtFirstName.getText(),txtLastName.getText(),txtNickname.getText(),
715 txtEmail.getText(),txtAddress.getText(),txtPhoneNo.getText(),txtWebpage.getText(),txtBDay.getText());
716 v.addElement(contact);
717 txtFirstName.setText(t: "");
718 txtLastName.setText(t: "");
719 txtNickname.setText(t: "");
720 txtEmail.setText(t: "");
721 txtAddress.setText(t: "");
722 txtPhoneNo.setText(t: "");
723 txtWebpage.setText(t: "");
724 txtBDay.setText(t: "");
725 if(BtnSaveAdded.isEnabled() == false)
726     if(BtnSaveAdded.isEnabled() == false)
727         BtnSaveAdded.setEnabled(b: true);
728 }
729 else if(ae.getActionCommand() == "Save Added!")
730 {
731
732     saveVector();
733     newFrame.setVisible(b: false);
734
735 }
736
737 else if(ae.getActionCommand() == "Ok")
738 {
739     newFrame.setVisible(b: false);
740
741 }
742 else if(ae.getActionCommand() == "Delete Selected")
743 {
744
745     int index;
746     try
747     {
748
749         index = list.getSelectedIndex();
750
751         if(index!=-1)
752         {
753

```



```

753 JOptionPane.showMessageDialog(newFrame, "Select a Contact first to delete it.",
754 "Select a Contact First", JOptionPane.INFORMATION_MESSAGE);
755 }
756 else
757 {
758     int n =
759     JOptionPane.showConfirmDialog(newFrame, "Are you sure you want to delete
760 the selected Contact?",
761 "Are you sure?", JOptionPane.YES_NO_OPTION);
762     if (n == JOptionPane.YES_OPTION)
763     {
764         listModel.remove(index);
765         v.removeElementAt(index);
766         saveVector();
767         newFrame.show();
768     }
769     else if (n ==JOptionPane.NO_OPTION)
770     {
771     }
772 }
773 }
774 }
775 }
776 catch(Exception e)
777 {
778 }
779 }
780 }
781 else if(ae.getActionCommand() == "Cancel")
782 {
783     newFrame.setVisible(false);
784 }
785 else if(ae.getActionCommand() == "Search!")
786 {
787     String SearchStr;
788     SearchStr = txtSearch.getText();
789     boolean flag=false;
790     Contact con = new Contact();
791     int c=0;
792     for(int t=0;t<5;t++)
793     {
794         data[t][0] = "";
795         data[t][1] = "";
796         data[t][2] = "";
797         data[t][3] = "";
798         data[t][4] = "";
799         data[t][5] = "";
800         data[t][6] = "";
801         data[t][7] = "";
802     }
803     for(int t=0;t<v.size();t++)
804     {
805         con = (Contact) v.elementAt(t);

```

```

809 if(SearchStr.equalsIgnoreCase(con.getFName()) || SearchStr.equalsIgnoreCase(con.getLName())
810 || SearchStr.equalsIgnoreCase(con.getFName() + " " + con.getLName()))
811 {
812     flag=true;
813
814     data[c][0] = con.getFName();
815     data[c][1] = con.getLName();
816     data[c][2] = con.getNname();
817     data[c][3] = con.getEmail();
818     data[c][4] = con.getAddress();
819     data[c][5] = con.getPhoneNo();
820     data[c][6] = con.getWebpage();
821     data[c][7] = con.getBday();
822     searchTable = new JTable(data,columnNames);
823     newFrame.setSize(700,221);
824     newFrame.setSize(700,220);
825     if(c<5)
826         c++;
827 }
828
829 if(flag)
830 {
831     JOptionPane.showMessageDialog(newFrame,"Contact Found!", "Search Result!",
832     JOptionPane.INFORMATION_MESSAGE);
833 }
834 else
835 {
836     JOptionPane.showMessageDialog(newFrame, "No Such Contact Found!",
837     "Search Result!", JOptionPane.INFORMATION_MESSAGE);
838 }
839
840
841 }
842 else if(ae.getActionCommand() == "Sort Contacts")
843 {
844
845     if(byfname.isSelected())
846     {
847         Contact contact1 = new Contact();
848         Contact contact2 = new Contact();
849         Contact temp = new Contact();
850         int l,m;
851
852         for(l=0;l<v.size()-1;l++)
853         {
854             for(m=l+1;m<v.size();m++)
855             {
856                 contact1 = (Contact) v.elementAt(l);
857                 contact2 = (Contact) v.elementAt(m);
858
859
860                 if(contact1.getFName().compareTo(contact2.getFName()) > 0)
861                 {
862                     temp = (Contact)v.elementAt(m);
863                     v.setElementAt(v.elementAt(l),m);
864                     v.setElementAt(temp,l);
865                 }
866             }
867         }
868     }
869 }

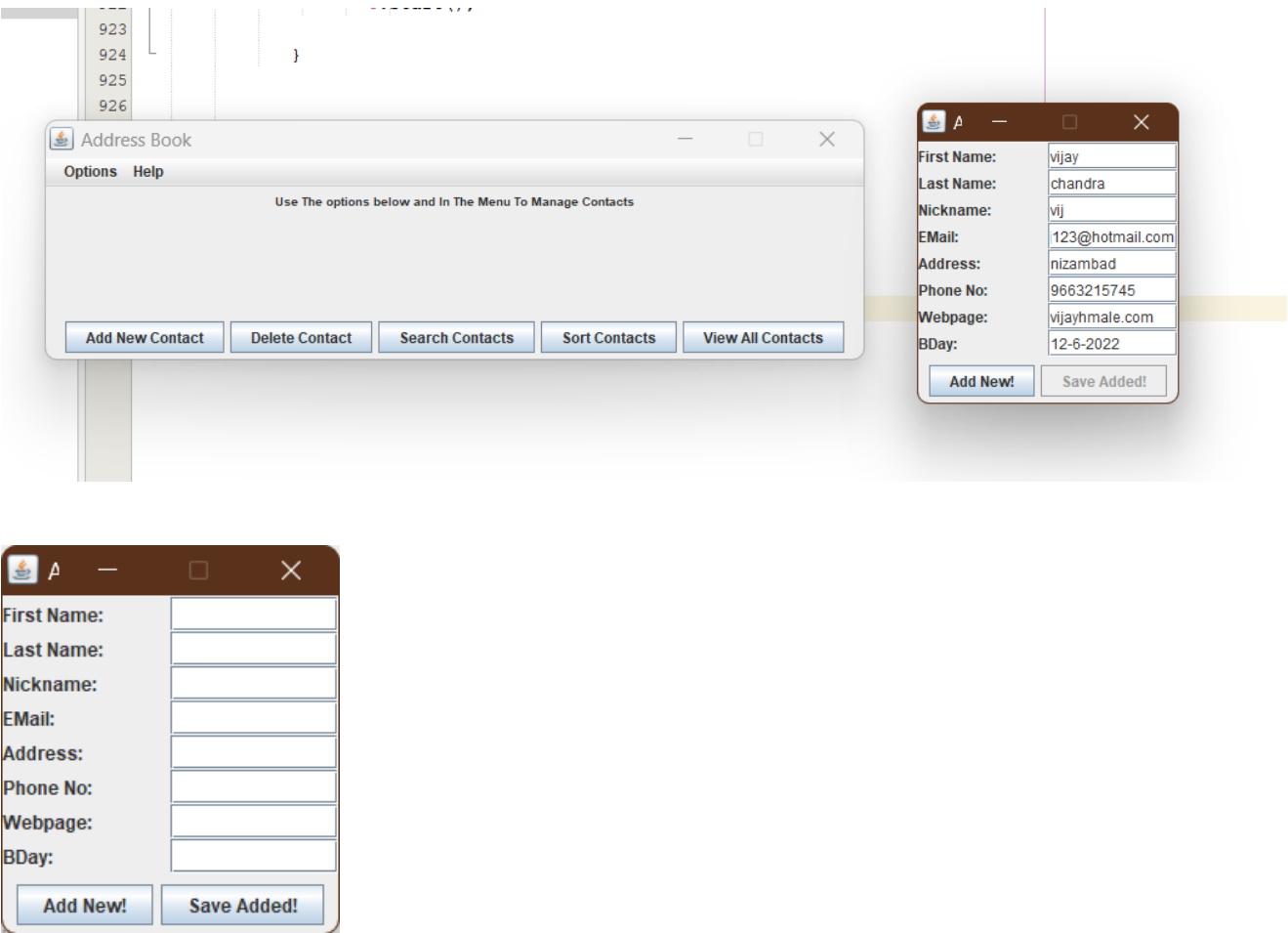
```

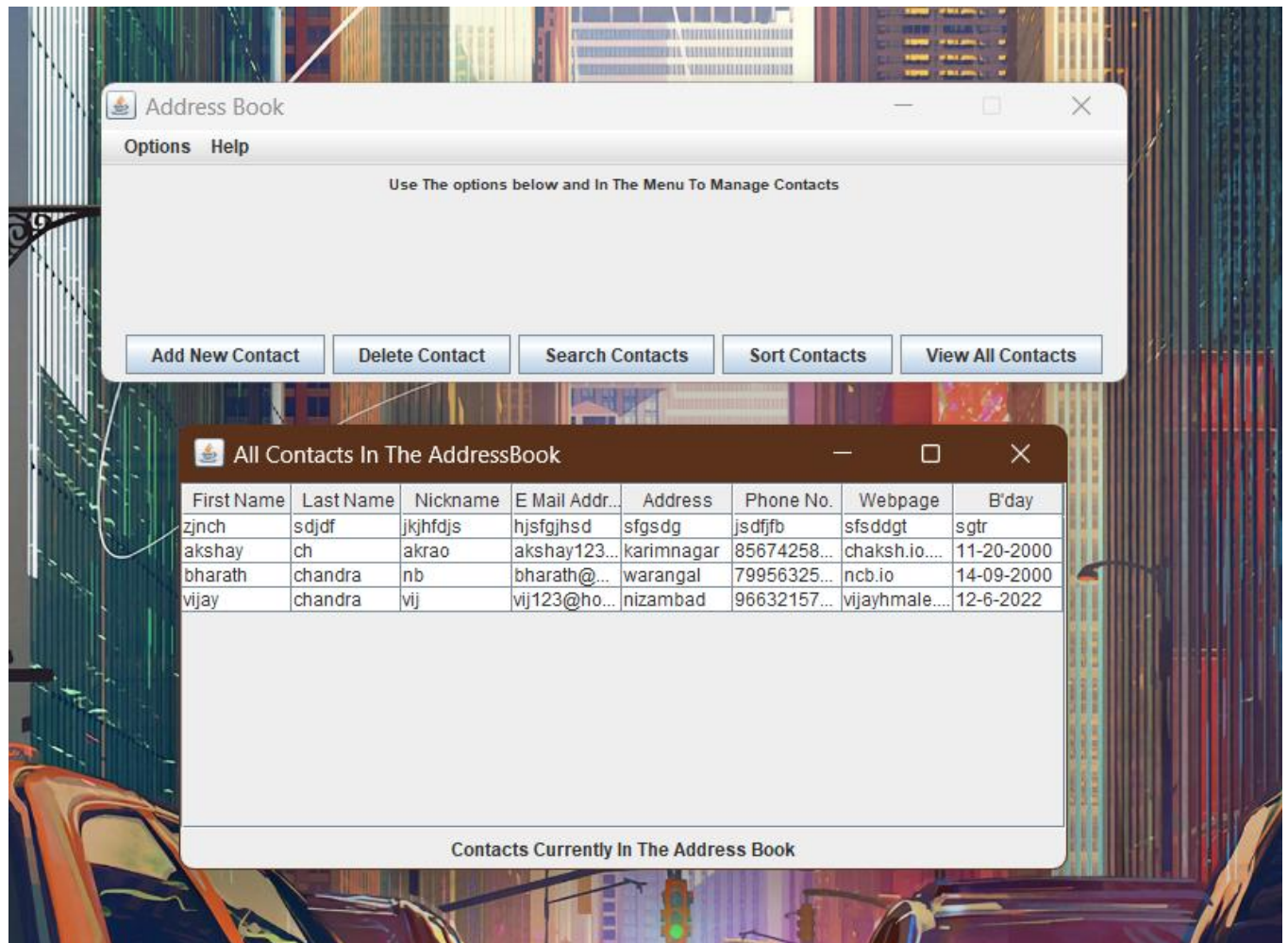
```

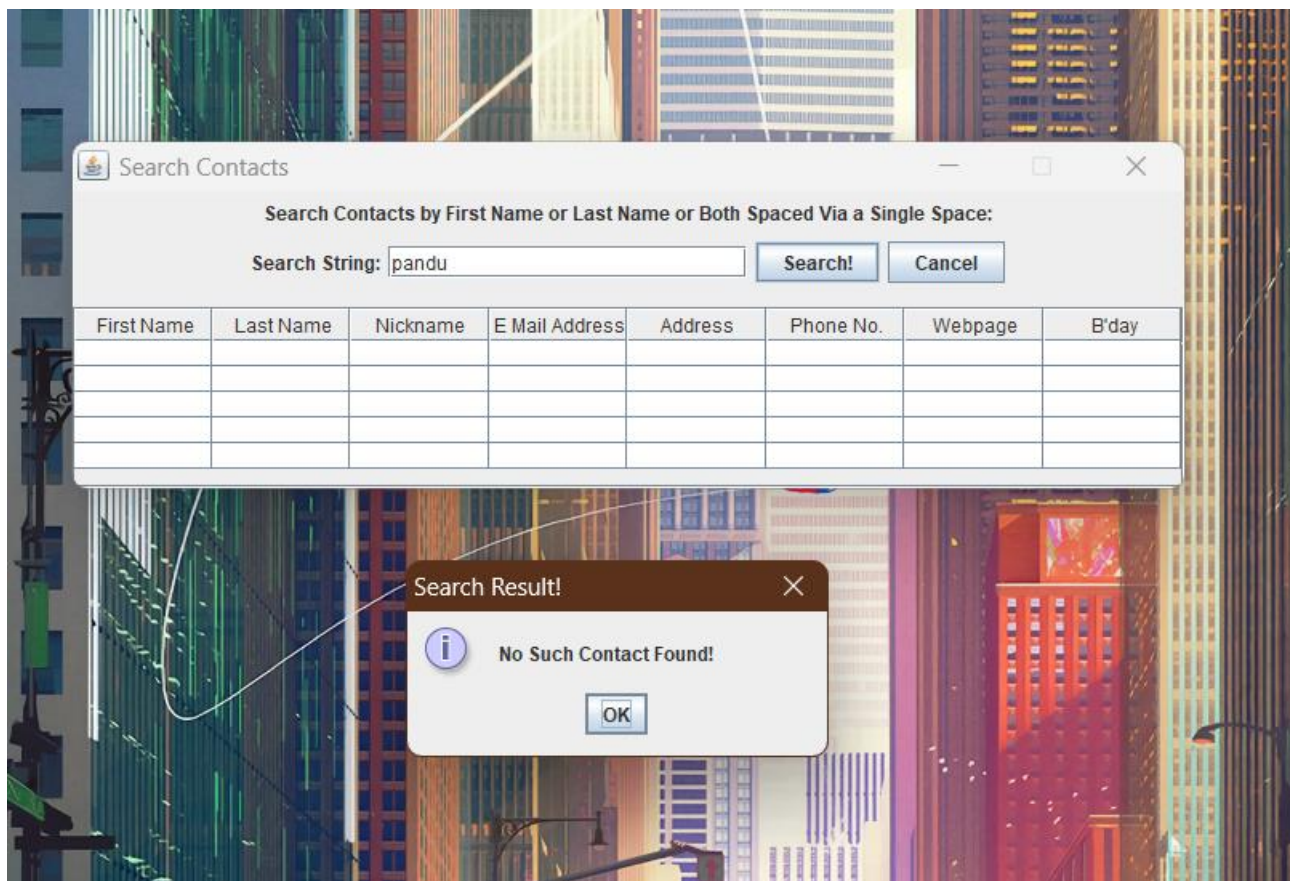
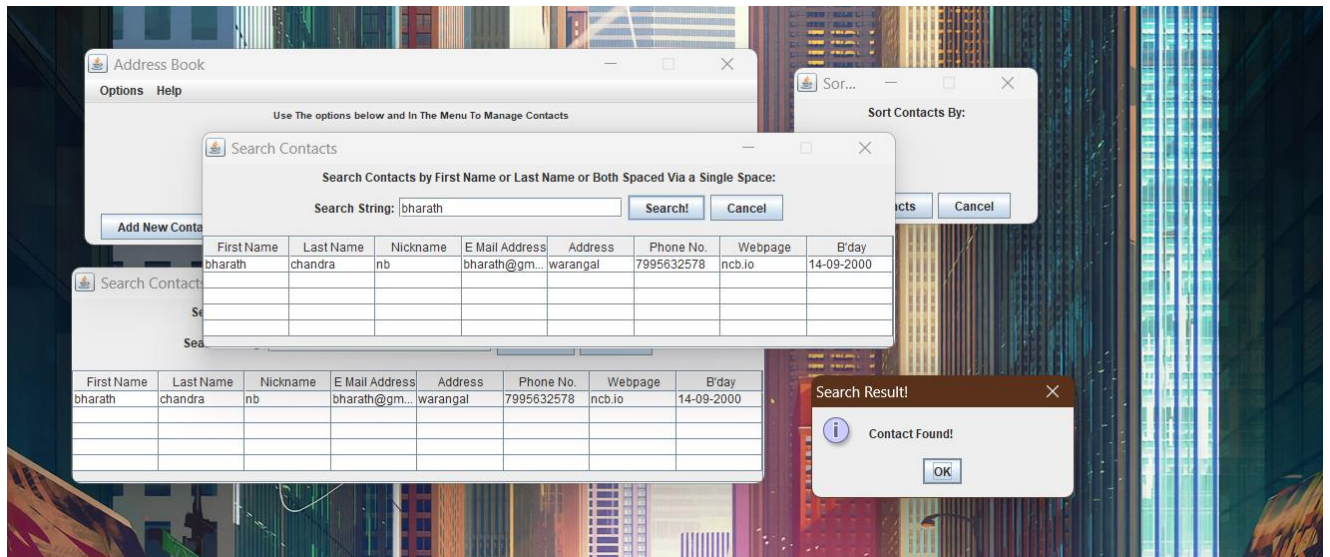
870         saveVector();
871     }
872     else
873     {
874
875         Contact contact1 = new Contact();
876         Contact contact2 = new Contact();
877         Contact temp = new Contact();
878         int l,m;
879         for(l=0;l<v.size()-1;l++)
880         {
881             for(m=l+1;m<v.size();m++)
882             {
883                 contact1 = (Contact) v.elementAt(l);
884                 contact2 = (Contact) v.elementAt(m);
885                 if(contact1.getLName().compareTo(contact2.getLName()) > 0)
886                 {
887                     temp = (Contact)v.elementAt(m);
888                     v.setElementAt(v.elementAt(l),m);
889                     v.setElementAt(temp,l);
890                 }
891             }
892         }
893         saveVector();
894     }
895     newFrame.setVisible(false);
896 }
897 }
898
899
900
901
902
903
904
905
906
907
908 public void saveVector()
909 {
910     t = new Thread(this, name: "Save Vector Thread");
911     t.start();
912 }
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

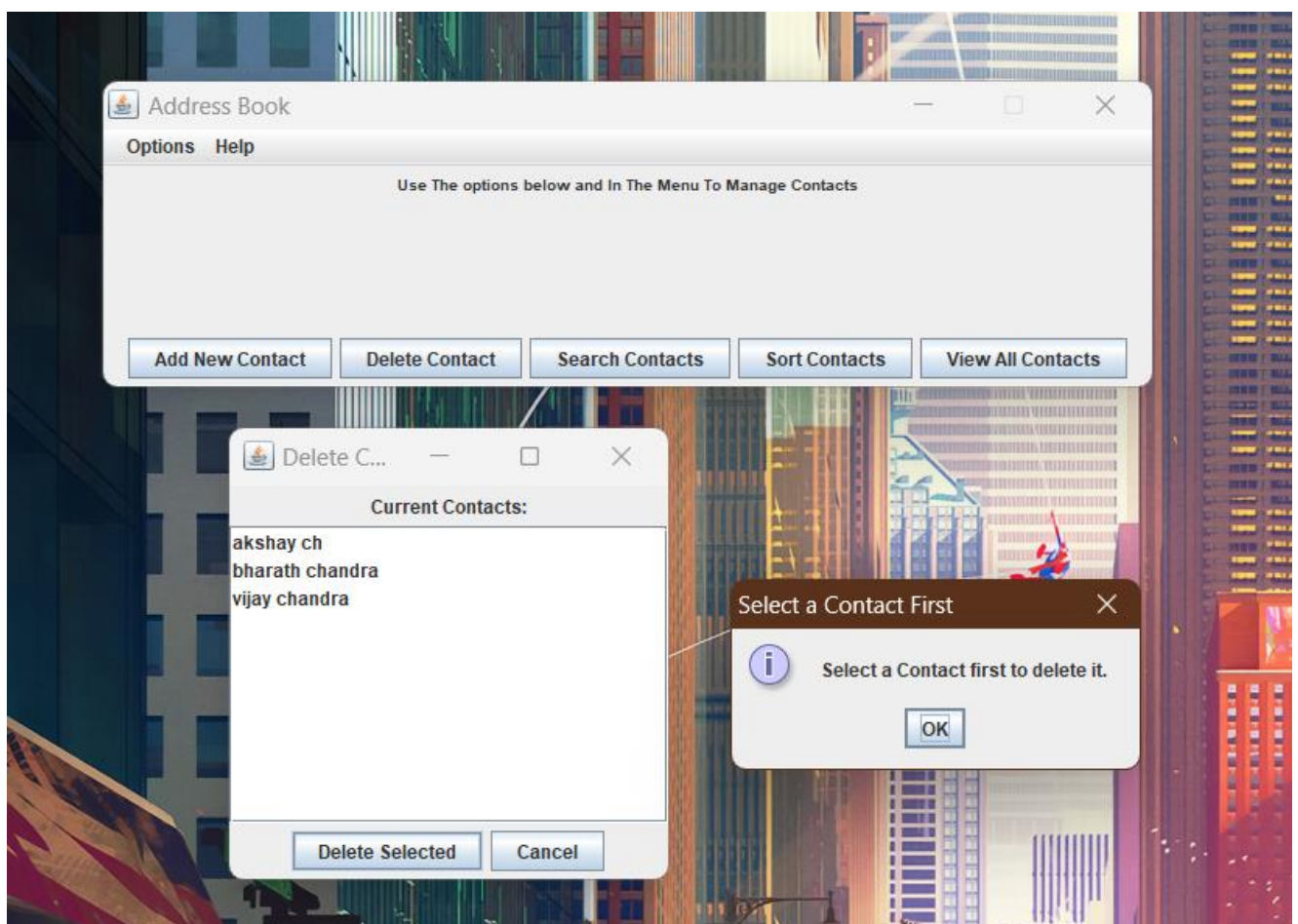
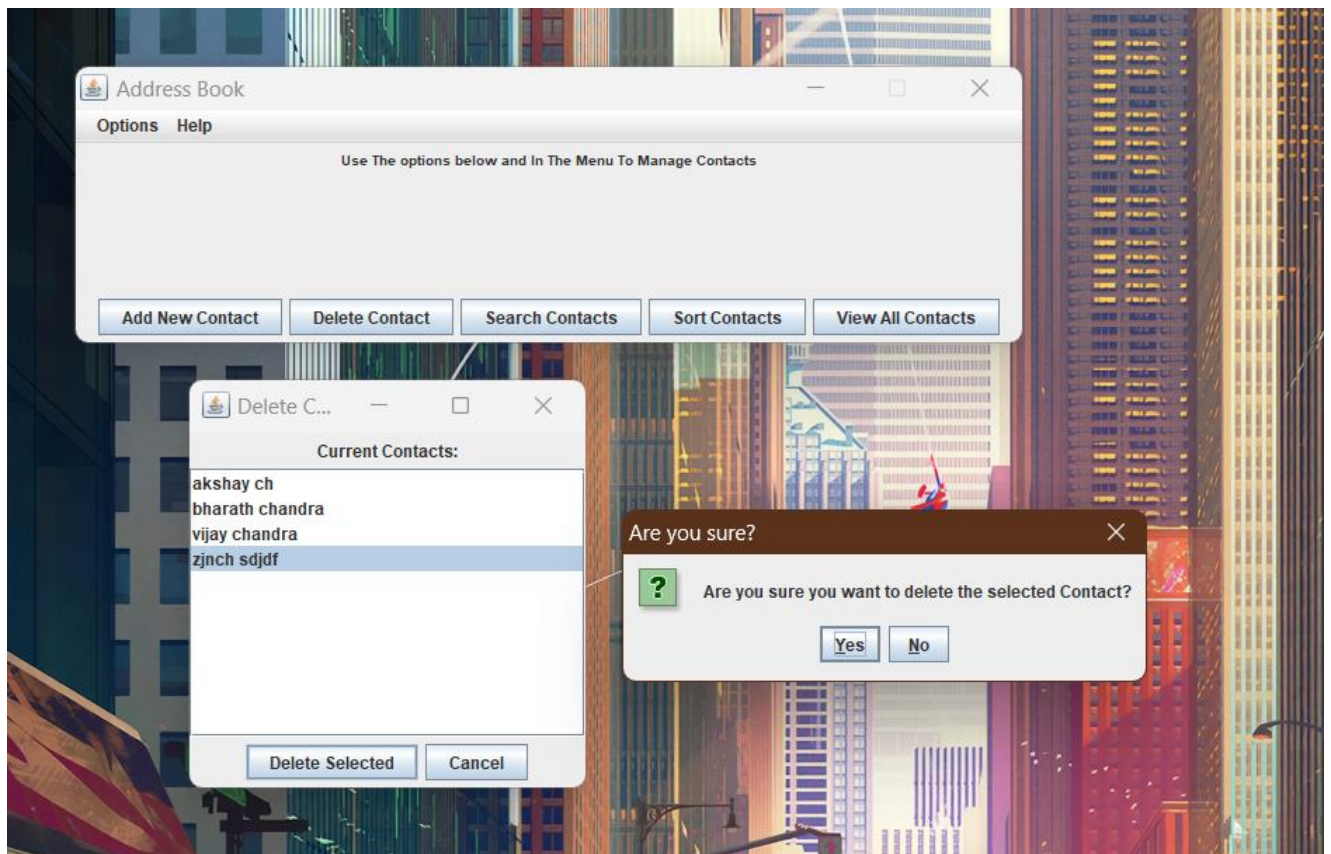
# 5.Result Screens











## **6.Conclusions :**

Address books now-a-day have the advantage of modernization. Computer have done the work more easy. Computer is playing a important role in data management. Addresses are very important nowadays because we need to stay in contact with many of your people, you cant remember them all.. A software for computers makes the things many times easy, these are made as user friendly and to keep an check the details of your friends and family. So ,these things are important and we have made things like these easy using our project