```java
package test;
import java.io.Serializable;
public class Actotal extends JFrame implements Serializable {
    private int total;
    private int account;
    private int depo;
    private int withdraw1;
    private String date;
    private String time;
    public String getDate() {
        return date;
    }
    public void setDate(String date) {
        this.date = date;
    }
    public String getTime() {
        return time;
    }
    public void setTime(String time) {
        this.time = time;
    }
    public int getTotal() {
        return total;
    }
    public void setTotal(int total) {
        this.total = total;
    }
    public int getAccount() {
        return account;
    }
    public void setAccount(int account) {
        this.account = account;
    }
    public int getDepo() {
        return depo;
    }
    public void setDepo(int depo) {
        this.depo = depo;
    }
    public int getWithdraw1() {
        return withdraw1;
    }
    public void setWithdraw1(int withdraw1) {
        this.withdraw1 = withdraw1;
    }
}

```

```java
package test;
import java.util.Calendar;
public class Actotalcreate extends JFrame {
    private static final long serialVersionUID = -948627250006298669L;
    private String sysdate,systime;
    Actotal sd=new Actotal();
    Actotalcreate(int a,int b,int d,int t) {
        Calendar cal = Calendar.getInstance();
        String cday = ""+cal.get(Calendar.DATE);
        int x =cal.get(Calendar.MONTH);
        String cmonth = ""+(x+1);
        String cyear =""+cal.get(Calendar.YEAR);
        sysdate = cday+"/"+cmonth+"/"+cyear;
        String chr = ""+cal.get(Calendar.HOUR_OF_DAY);
        String cmin = ""+cal.get(Calendar.MINUTE);
        String csec =""+cal.get(Calendar.SECOND);
        systime = chr+":"+cmin+":"+csec;
        sd.setAccount(a);
        sd.setWithdraw1(b);
        sd.setDepo(d);
        sd.setTotal(t);
        sd.setDate(sysdate);
        sd.setTime(systime);
        ArrayList<Actotal> list4;
        try {
           FileInputStream fin=new FileInputStream("actotal.dat");
           ObjectInputStream oin=new ObjectInputStream(fin);
           list4=(ArrayList<Actotal>)oin.readObject();
        }catch(Exception e){
             list4=new ArrayList<Actotal>();
        }
        list4.add(sd);
        try {
               FileOutputStream fout=new FileOutputStream("actotal.dat");
               ObjectOutputStream oout=new ObjectOutputStream(fout);
               oout.writeObject(list4);
        }catch(Exception e){}
    }
}

```

```java
package test;
import java.io.*;
public class AddInformation1 {
    ArrayList<Create> list4;
    public AddInformation1(Create rg1) {
        try {
            FileInputStream fin=new FileInputStream("Regis.dat");
            ObjectInputStream oin=new ObjectInputStream(fin);
            list4=(ArrayList<Create>)oin.readObject();
        }catch(Exception e) {
            list4=new ArrayList<Create>();
        }
        list4.add(rg1);
        try {
            FileOutputStream fout=new FileOutputStream("Regis.dat");
            ObjectOutputStream oout=new ObjectOutputStream(fout);
            oout.writeObject(list4);
        }catch(Exception e){}
    }
}

```

```java
package test;
import java.awt.event.*;
class Admin extends JFrame implements ActionListener {
    private JLabel l1,l2;
    private JButton b1,b2,b3,b4,b5,b6,b7,b8;
    private JButton b11;
    private JButton b12;
    private JButton b13;
    public Admin() {
        super("Welcome Administrator");
        Container c=getContentPane();
        c.setLayout(new GridLayout(8,2));
        Font f1=new Font("Times New Roman",Font.BOLD,20);
        l1=new JLabel("ADMINISTRATOR");
        l1.setFont(f1);
        l1.setForeground(Color.RED);
        JPanel bpanel=new JPanel();
        bpanel.add(l1);
        bpanel.setBackground(new Color(0,0,64));
        l2=new JLabel("");
        l2.setFont(f1);
        l2.setForeground(Color.RED);
        JPanel cpanel=new JPanel();
        cpanel.add(l2);
        cpanel.setBackground(new Color(0,0,64));
        b1=new JButton("ADD AN OPERATOR");
        b1.addActionListener(this);
        JPanel apanel=new JPanel();
        apanel.add(b1);
        apanel.setBackground(new Color(0,0,64));
        b2=new JButton("SEARCH AN OPERATOR");
        b2.addActionListener(this);
        JPanel dpanel=new JPanel();
        dpanel.add(b2);
        dpanel.setBackground(new Color(0,0,64));
        b3=new JButton("DELETE AN OPERATOR");
        b3.addActionListener(this);
        JPanel epanel=new JPanel();
        epanel.add(b3);
        epanel.setBackground(new Color(0,0,64));
        b4=new JButton("SHOW ALL OPERATORS");
        b4.addActionListener(this);
        JPanel fpanel=new JPanel();
        fpanel.add(b4);
        fpanel.setBackground(new Color(0,0,64));
        b5=new JButton("SHOW ALL USERS");
        b5.addActionListener(this);
        JPanel gpanel=new JPanel();
        gpanel.add(b5);
        gpanel.setBackground(new Color(0,0,64));
        b6=new JButton("SEARCH AN USER");
        b6.addActionListener(this);
        JPanel hpanel=new JPanel();
        hpanel.add(b6);
        hpanel.setBackground(new Color(0,0,64));
        b7=new JButton("DELETE AN USER");
        b7.addActionListener(this);
        JPanel opanel=new JPanel();
        opanel.add(b7);
        opanel.setBackground(new Color(0,0,64));
        b8=new JButton("LOG OUT");
        b8.addActionListener(this);
```

```
65          JPanel qpanel=new JPanel();
66          qpanel.add(b8);
67          qpanel.setBackground(new Color(0,0,64));
68          b11=new JButton("LOCKER DETAILS OF USER");
69          b11.addActionListener(this);
70          JPanel qoopanel=new JPanel();
71          qoopanel.add(b11);
72          qoopanel.setBackground(new Color(0,0,64));
73          b12=new JButton("FIXED DEPOSIT DETAILS");
74          b12.addActionListener(this);
75          JPanel qipanel=new JPanel();
76          qipanel.add(b12);
77          qipanel.setBackground(new Color(0,0,64));
78          b13=new JButton("LOCKER APPROVE");
79          b13.addActionListener(this);
80          JPanel q2panel=new JPanel();
81          q2panel.add(b13);
82          q2panel.setBackground(new Color(0,0,64));
83          c.add(bpanel);c.add(cpanel);
84          c.add(apanel);c.add(dpanel);
85          c.add(epanel);c.add(fpanel);
86          c.add(gpanel);c.add(hpanel);
87          c.add(qoopanel);c.add(qipanel);
88          c.add(q2panel);c.add(opanel);
89          c.add(qpanel);
90          setSize(600,525);
91          setLocation(200,200);
92          setResizable(false);
93          setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
94          setVisible(true);
95      }
96      public void actionPerformed(ActionEvent e) {
97          if(e.getSource()==b1) {
98              new OperatorReg("");
99              setVisible(false);
100         }
101         if(e.getSource()==b2) {
102             new OperatorSearch();
103             setVisible(false);
104         }
105         if(e.getSource()==b3) {
106             new OperatorDelete();
107             setVisible(false);
108         }
109         if(e.getSource()==b4)
110             new OperatorDisplay();
111         if(e.getSource()==b5)
112             new DiaplayAll() ;
113         if(e.getSource()==b6) {
114             new UserSearch();
115             setVisible(false);
116         }
117         if(e.getSource()==b7) {
118             new AdminUserRemove();
119             setVisible(false);
120         }
121         if(e.getSource()==b8) {
122             int con=JOptionPane.showConfirmDialog(this, "Are You Sure to cancel?");
123             if(con==JOptionPane.YES_OPTION) {
124                 new SecondWindow();
125                 setVisible(false);
126             }
```

```
        }
    }
}
public class AdminLogin {
   public static void main(String[] args) {}
}

```

```java
package test;
import java.awt.event.*;
public class AdminUserRemove extends JFrame implements ActionListener {
    private JLabel l1,l2,l3;
    private JTextField t1;
    private JButton submit,back;
    private boolean flagAccNum = false;
    public AdminUserRemove() {
        super("Remove an user");
        Container c=getContentPane();
        c.setLayout(new GridLayout(3,2));
        Font f1=new Font("Times New Roman",Font.BOLD,20);
        l1=new JLabel("USER");
        l1.setFont(f1);
        l1.setForeground(Color.RED);
        JPanel fpanel=new JPanel();
        fpanel.add(l1);
        fpanel.setBackground(new Color(0,0,64));
        l2=new JLabel("Delete");
        l2.setFont(f1);
        l2.setForeground(Color.RED);
        JPanel apanel=new JPanel();
        apanel.add(l2);
        apanel.setBackground(new Color(0,0,64));
        l3=new JLabel("Give USER A/C no");
        l3.setFont(f1);
        l3.setForeground(Color.RED);
        JPanel bpanel=new JPanel();
        bpanel.add(l3);
        bpanel.setBackground(new Color(0,0,64));
        t1=new JTextField();
        submit=new JButton("Delete");
        submit.addActionListener(this);
        JPanel cpanel=new JPanel();
        cpanel.add(submit);
        cpanel.setBackground(new Color(0,0,64));
        back=new JButton("BACK");
        back.addActionListener(this);
        JPanel dpanel=new JPanel();
        dpanel.add(back);
        dpanel.setBackground(new Color(0,0,64));
        c.add(fpanel);c.add(apanel);
        c.add(bpanel);c.add(t1);
        c.add(cpanel);c.add(dpanel);
        setSize(550,300);
        setLocation(200,200);
        setResizable(false);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }
    public void actionPerformed(ActionEvent e) {
        if(e.getSource() == submit) {
            String vaccNum = t1.getText();
            String accNumpattern = "^[0-9]{4}" ;
            Scanner   scan  =  new Scanner( vaccNum ) ;
            String   matched =  scan.findInLine( accNumpattern ) ;
            if ( matched == null ) {
                JOptionPane.showMessageDialog(this, "INVALID ACCOUNT NUMBER\nIt should contain only digits with minimum length of 4 digit.", "Error", JOptionPane.ERROR_MESSAGE);
                t1.setText("");
            }
```

```
64              else
65                  flagAccNum    = true;
66              if(flagAccNum == true) {
67                  String sname = t1.getText().trim();
68                  new UserRemoveDisplay(sname);
69                  t1.setText("");
70              }
71              flagAccNum = false;
72          }
73          if(e.getSource()==back) {
74              int con=JOptionPane.showConfirmDialog(this, "Are You Sure to cancel?");
75              if(con==JOptionPane.YES_OPTION) {
76                 new Admin();
77                 setVisible(false);
78              }
79          }
80      }
81  }
82
```

```java
package test;
import java.awt.*;
public class ChangePassword extends JFrame implements ActionListener {
    private JLabel lHeading,lFirstName,lCurrentPassword,lNewPassword;
    private JTextField tFirstName;
    private JPasswordField pCurrentPassword,pNewPassword;
    private JButton bBack,bChange;
    ArrayList<OperatorRegData> list;
    private boolean flagFirstName = false;
    private boolean flagNewPassword = false;
    private boolean flagCurrentPassword = false;
    private boolean flagFirstNameMatch = false;
    private boolean flagCurrentPasswordMatch = false;
    private JLabel lHeading2;
    public ChangePassword() {
        super("Change Password");
        Container c=getContentPane();
        c.setLayout(new GridLayout(5,2));
        Font f1=new Font("Chiller",Font.BOLD,22);
        Font f2=new Font("Times New Roman",Font.BOLD,20);
        lHeading=new JLabel("CHANGE PASSWORD");
        lHeading.setFont(f1);
        lHeading.setForeground(Color.RED);
        JPanel apanel=new JPanel();
        apanel.add(lHeading);
        apanel.setBackground(new Color(0,0,64));
        lHeading2=new JLabel("");
        lHeading2.setFont(f1);
        lHeading2.setForeground(Color.RED);
        JPanel abpanel=new JPanel();
        abpanel.add(lHeading2);
        abpanel.setBackground(new Color(0,0,64));
        lFirstName=new JLabel("Enter First Name");
        lFirstName.setFont(f2);
        lFirstName.setForeground(Color.GRAY);
        JPanel bpanel=new JPanel();
        bpanel.add(lFirstName);
        bpanel.setBackground(new Color(0,0,64));
        lCurrentPassword=new JLabel("Enter Current Password");
        lCurrentPassword.setFont(f2);
        lCurrentPassword.setForeground(Color.GRAY);
        JPanel cpanel=new JPanel();
        cpanel.add(lCurrentPassword);
        cpanel.setBackground(new Color(0,0,64));
        lNewPassword=new JLabel("Enter New Password");
        lNewPassword.setFont(f2);
        lNewPassword.setForeground(Color.GRAY);
        JPanel dpanel=new JPanel();
        dpanel.add(lNewPassword);
        dpanel.setBackground(new Color(0,0,64));
        tFirstName=new JTextField();
        pCurrentPassword=new JPasswordField();
        pNewPassword=new JPasswordField();
        bChange=new JButton("Change");
        bChange.addActionListener(this);
        JPanel epanel=new JPanel();
        epanel.add(bChange);
        epanel.setBackground(new Color(0,0,64));
        bBack=new JButton("Back");
        bBack.addActionListener(this);
        JPanel fpanel=new JPanel();
        fpanel.add(bBack);
```

```
67          fpanel.setBackground(new Color(0,0,64));
68          c.add(apanel);c.add(abpanel);
69          c.add(bpanel);c.add(tFirstName);
70          c.add(cpanel);c.add(pCurrentPassword);
71          c.add(dpanel);c.add(pNewPassword);
72          c.add(epanel);c.add(fpanel);
73          setSize(500,500);
74          setLocation(100,100);
75          setResizable(false);
76          setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
77          setVisible(true);
78      }
79      public void actionPerformed(ActionEvent e) {
80          if(e.getSource() == bChange) {
81              String vFirstName = tFirstName.getText();
82              String vCurrentPassword = pCurrentPassword.getText();
83              String vNewPassword = pNewPassword.getText();
84              String firstNamepattern = "^[a-zA-Z]" ;
85              Scanner   scan1  =  new Scanner( vFirstName ) ;
86              String   matched1  =  scan1.findInLine( firstNamepattern ) ;
87              if ( matched1 == null ) {
88                  JOptionPane.showMessageDialog(this, "INVALID FIRST NAME\nIt should
    contain only alphabets.", "Error", JOptionPane.ERROR_MESSAGE);
89                  tFirstName.setText("");
90              }
91              else
92                  flagFirstName   = true;
93              String currentPasswordpattern = "^[0-9]{4}";
94              Scanner   scan2  =  new Scanner( vCurrentPassword );
95              String   matched2  =  scan2.findInLine( currentPasswordpattern );
96              if ( matched2 == null ) {
97                  JOptionPane.showMessageDialog(this, "INVALID ACCOUNT NUMBER\nIt
    should contain only digits with minimum length of 4 digit.", "Error",
    JOptionPane.ERROR_MESSAGE);
98                  pCurrentPassword.setText("");
99              }
100             else
101                 flagCurrentPassword  = true;
102             String newPasswordpattern = "^[0-9]{4}" ;
103             Scanner   scan3  =  new Scanner( vNewPassword) ;
104             String   matched3  =  scan3.findInLine( currentPasswordpattern ) ;
105             if ( matched3 == null ) {
106                 JOptionPane.showMessageDialog(this, "INVALID NEW PASSWORD\nIt
    should contain only digits with minimum length of 4 digit.", "Error",
    JOptionPane.ERROR_MESSAGE);
107                 pNewPassword.setText("");
108             }
109             else
110                 flagNewPassword   = true;
111             if(flagFirstName == true) {
112                 if(flagCurrentPassword == true) {
113                     if(flagNewPassword == true) {
114                         try {
115                            FileInputStream fin=new FileInputStream("Reg.dat");
116                            ObjectInputStream oin=new ObjectInputStream(fin);
117                            list=(ArrayList<OperatorRegData>)oin.readObject();
118                         }catch(Exception e1) {
119                             JOptionPane.showMessageDialog(this, "No file found in
    database", "Error", JOptionPane.ERROR_MESSAGE);
120                             list=new ArrayList<OperatorRegData>();
121                         }
122                         for(int i=0;i<list.size();i++) {
```

```
123                             if(list.get(i).getName().equals(tFirstName.getText()))
    {
124                                 flagFirstNameMatch = true;
125                                 if(list.get(i).getPassword().equals(pCurrentPasswor
    d.getText())) {
126                                     flagCurrentPasswordMatch = true;
127                                     list.get(i).setPassword(pNewPassword.getText());
128                                 }
129                             }
130                         }
131                         try {
132                            FileOutputStream fout=new FileOutputStream("Reg.dat");
133                            ObjectOutputStream oout=new ObjectOutputStream(fout);
134                            oout.writeObject(list);
135                         }catch(Exception e1){}
136                         if((flagFirstNameMatch == true) &&
    (flagCurrentPasswordMatch == true)) {
137                             JOptionPane.showMessageDialog(this, "Password has been
    successfully changed");
138                             tFirstName.setText("");
139                             pCurrentPassword.setText("");
140                             pNewPassword.setText("");
141                         }
142                         if((flagFirstNameMatch == true) &&
    (flagCurrentPasswordMatch == false)) {
143                             JOptionPane.showMessageDialog(this, "Password
    incorrect", "Error", JOptionPane.ERROR_MESSAGE);
144                             pCurrentPassword.setText("");
145                         }
146                         if((flagFirstNameMatch == false) &&
    (flagCurrentPasswordMatch == false)) {
147                             JOptionPane.showMessageDialog(this, "Operator does not
    exist.", "Error", JOptionPane.ERROR_MESSAGE);
148                             tFirstName.setText("");
149                             pCurrentPassword.setText("");
150                             pNewPassword.setText("");
151                         }
152                     }
153                 }
154             }
155             flagFirstName = false;
156             flagCurrentPassword = false;
157             flagNewPassword = false;
158             flagFirstNameMatch = false;
159             flagCurrentPasswordMatch = false;
160         }
161         if(e.getSource()==bBack) {
162             int rply = JOptionPane.showConfirmDialog(this, "Are you sure to
    quit?");
163             if(rply == JOptionPane.YES_OPTION) {
164                 new Userwindow("");
165                 setVisible(false);
166             }
167         }
168     }
169 }
```

```java
package test;
import java.io.Serializable;
public class Create implements Serializable {
    private String ac;
    private String name;
    private String name1;
    private String address;
    private String email;
    private String nationality;
    private String acctype;
    private String city;
    private String gender;
    private String dob;
    private String date;
    private String time;
    private String Identity;
    private String profession;
    private String initialamnt;
    public String getAc() {
        return ac;
    }
    public void setAc(String ac) {
        this.ac = ac;
    }
    public String getInitialamnt() {
        return initialamnt;
    }
    public void setInitialamnt(String initialamnt) {
        this.initialamnt = initialamnt;
    }
    public String getNationality() {
        return nationality;
    }
    public void setNationality(String nationality) {
        this.nationality = nationality;
    }
    public String getAcctype() {
        return acctype;
    }
    public void setAcctype(String acctype) {
        this.acctype = acctype;
    }
    public String getIdentity() {
        return Identity;
    }
    public void setIdentity(String identity) {
        Identity = identity;
    }
    public String getProfession() {
        return profession;
    }
    public void setProfession(String profession) {
        this.profession = profession;
    }
    public String getName1() {
        return name1;
    }
    public void setName1(String name1) {
        this.name1 = name1;
    }
    public String getEmail() {
        return email;
```

```java
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }
    public String getCity() {
        return city;
    }
    public void setCity(String city) {
        this.city = city;
    }
    public String getGender() {
        return gender;
    }
    public void setGender(String gender) {
        this.gender = gender;
    }
    public String getDob() {
        return dob;
    }
    public void setDob(String dob) {
        this.dob = dob;
    }
    public String getDate() {
        return date;
    }
    public void setDate(String date) {
        this.date = date;
    }
    public String getTime() {
        return time;
    }
    public void setTime(String time) {
        this.time = time;
    }
}
```

```java
package test;
import java.awt.*;
public class CreateFrame extends JFrame implements ActionListener {
    private JLabel
  l0,l01,l1,l2,l3,l4,l40,l5,l6,l7,l8,l9,l10,l11,l12,l13,l14,l16,l17,l18,l19,l20;
    private JTextField t1,tid1,tid2,tid3,tid4,tid5,tid6,tid7,tid8,tid10;
    private JComboBox city,day,month,year,annualincome,identity,profession,acno;
    private JRadioButton rmale,rfemale,rsav,rothers,married,unmarried;
    private JButton bsubmit,back;
    private String sysdate,systime;
    private int count = 0;
    private String ini;
    private boolean flagName = false;
    private boolean flagName1 = false;
    private boolean flagAddress = false;
    private boolean flagNationality = false;
    private ArrayList<Create> alist;
    private boolean flagAccountNumber = false;
    private boolean flagInitialAmount = false;
    private boolean flagLandline = false;
    private boolean flagMobile = false;
    private boolean flagAge = false;
    private boolean flagProof = false;
    CreateFrame() {
        super("User registration form");
        Container c=getContentPane();
        c.setLayout(new GridLayout(21,2));
        tid1=new JTextField();
        tid2=new JTextField();
        tid3=new JTextField();
        tid4=new JTextField();
        tid5=new JTextField();
        tid6=new JTextField();
        tid7=new JTextField();
        tid8=new JTextField();
        tid10=new JTextField();
        String cvalue[]={"<50000","50000-100000","100000-600000",">600000"};
        annualincome=new JComboBox(cvalue);
        String d1value[]={"Government Service","Business","Private
  Sector","Student","Others"};
        profession=new JComboBox(d1value);
        String evalue[]={"Pan Card","Voter Card","Ration Card","Driving License"};
        identity=new JComboBox(evalue);
        rsav=new JRadioButton("Savings");
        rothers=new JRadioButton("Savings");
        ButtonGroup r1group=new ButtonGroup();
        r1group.add(rsav);
        JPanel kpanel=new JPanel();
        kpanel.add(rsav);
        rmale=new JRadioButton("Male");
        rfemale=new JRadioButton("Female");
        ButtonGroup rgroup=new ButtonGroup();
        rgroup.add(rmale);
        rgroup.add(rfemale);
        JPanel gpanel=new JPanel();
        married=new JRadioButton("Married");
        unmarried=new JRadioButton("Single");
        ButtonGroup r2group=new ButtonGroup();
        r2group.add(married);
        r2group.add(unmarried);
        JPanel gopanel=new JPanel();
        gopanel.add(married);
```

```
        gopanel.add(unmarried);
        gpanel.add(rmale);
        gpanel.add(rfemale);
        t1=new JTextField();
        String dvalue[]=new String[31];
        for(int i=0;i<=30;i++)
            dvalue[i]=String.valueOf(i+1);
        day=new JComboBox(dvalue);
        String mvalue[]=new String[12];
        for(int i=0;i<=11;i++)
            mvalue[i]=String.valueOf(i+1);
        month=new JComboBox(mvalue);
        String yvalue[]=new String[25];
        int cnt=0;
        for(int i=1989;i<=2013;i++) {
            yvalue[cnt]=String.valueOf(i);
            cnt++;
        }
        year=new JComboBox(yvalue);
        JPanel cpanel=new JPanel();
        cpanel.add(day);
        cpanel.add(month);
        cpanel.add(year);
        bsubmit=new JButton("Register");
        bsubmit.addActionListener(this);
        Font f1=new Font("Times New Roman",Font.BOLD,14);
        l01=new JLabel("                                        ACCOUNT OPENING");
        l01.setFont(f1);
        l01.setForeground(Color.RED);
        JPanel fpanel=new JPanel();
        fpanel.add(l01);
        fpanel.setBackground(new Color(0,0,64));
        l0=new JLabel("FORM                                                    ");
        l0.setFont(f1);
        l0.setForeground(Color.RED);
        JPanel apanel=new JPanel();
        apanel.add(l0);
        apanel.setBackground(new Color(0,0,64));
        l1=new JLabel("Enter First Name");
        l1.setFont(f1);
        l1.setForeground(Color.RED);
        JPanel bpanel=new JPanel();
        bpanel.add(l1);
        bpanel.setBackground(new Color(0,0,64));
        l2=new JLabel("Enter Last Name");
        l2.setFont(f1);
        l2.setForeground(Color.RED);
        JPanel copanel=new JPanel();
        copanel.add(l2);
        copanel.setBackground(new Color(0,0,64));
        l3=new JLabel("Enter Address");
        l3.setFont(f1);
        l3.setForeground(Color.RED);
        JPanel dpanel=new JPanel();
        dpanel.add(l3);
        dpanel.setBackground(new Color(0,0,64));
        l4=new JLabel("Enter Phone Number(Landline)");
        l4.setFont(f1);
        l4.setForeground(Color.RED);
        JPanel epanel=new JPanel();
        epanel.add(l4);
        epanel.setBackground(new Color(0,0,64));
```

```
127         l40=new JLabel("Enter Phone Number(Mobile)");
128         l40.setFont(f1);
129         l40.setForeground(Color.RED);
130         JPanel vpanel=new JPanel();
131         vpanel.add(l40);
132         vpanel.setBackground(new Color(0,0,64));
133         l5=new JLabel("Select Sex");
134         l5.setFont(f1);
135         l5.setForeground(Color.RED);
136         JPanel qpanel=new JPanel();
137         qpanel.add(l5);
138         qpanel.setBackground(new Color(0,0,64));
139         l6=new JLabel("Natinality");
140         l6.setFont(f1);
141         l6.setForeground(Color.RED);
142         JPanel wpanel=new JPanel();
143         wpanel.add(l6);
144         wpanel.setBackground(new Color(0,0,64));
145         l7=new JLabel("Select Age");
146         l7.setFont(f1);
147         l7.setForeground(Color.RED);
148         JPanel opanel=new JPanel();
149         opanel.add(l7);
150         opanel.setBackground(new Color(0,0,64));
151         l8=new JLabel("Select A/C Type");
152         l8.setFont(f1);
153         l8.setForeground(Color.RED);
154         JPanel zpanel=new JPanel();
155         zpanel.add(l8);
156         zpanel.setBackground(new Color(0,0,64));
157         l9=new JLabel("Date of Birth");
158         l9.setFont(f1);
159         l9.setForeground(Color.RED);
160         JPanel xpanel=new JPanel();
161         xpanel.add(l9);
162         xpanel.setBackground(new Color(0,0,64));
163         l10=new JLabel("Enter Annual Income");
164         l10.setFont(f1);
165         l10.setForeground(Color.RED);
166         JPanel vxpanel=new JPanel();
167         vxpanel.add(l10);
168         vxpanel.setBackground(new Color(0,0,64));
169         l11=new JLabel("Enter Profession");
170         l11.setFont(f1);
171         l11.setForeground(Color.RED);
172         JPanel vppanel=new JPanel();
173         vppanel.add(l11);
174         vppanel.setBackground(new Color(0,0,64));
175         l12=new JLabel("Enter Marital Status");
176         l12.setFont(f1);
177         l12.setForeground(Color.RED);
178         JPanel npanel=new JPanel();
179         npanel.add(l12);
180         npanel.setBackground(new Color(0,0,64));
181         l13=new JLabel("Enter Identity Proof");
182         l13.setFont(f1);
183         l13.setForeground(Color.RED);
184         JPanel mpanel=new JPanel();
185         mpanel.add(l13);
186         mpanel.setBackground(new Color(0,0,64));
187         l14=new JLabel("Enter Identity proof no:");
188         l14.setFont(f1);
```

```
189         l14.setForeground(Color.RED);
190         JPanel jpanel=new JPanel();
191         jpanel.add(l14);
192         jpanel.setBackground(new Color(0,0,64));
193         Calendar cal = Calendar.getInstance();
194         String cday = ""+cal.get(Calendar.DATE);
195         int x =cal.get(Calendar.MONTH);
196         String cmonth = ""+(x+1);
197         String cyear =""+cal.get(Calendar.YEAR);
198         sysdate = cday+"/"+cmonth+"/"+cyear;
199         l16=new JLabel(sysdate);
200         String chr = ""+cal.get(Calendar.HOUR_OF_DAY);
201         String cmin = ""+cal.get(Calendar.MINUTE);
202         String csec =""+cal.get(Calendar.SECOND);
203         systime = chr+":"+cmin+":"+csec;
204         l17=new JLabel(systime);
205         l18=new JLabel("Current Time");
206         l18.setFont(f1);
207         l18.setForeground(Color.RED);
208         JPanel vopanel=new JPanel();
209         vopanel.add(l18);
210         vopanel.setBackground(new Color(0,0,64));
211         l19=new JLabel("Current Date");
212         l19.setFont(f1);
213         l19.setForeground(Color.RED);
214         JPanel vipanel=new JPanel();
215         vipanel.add(l19);
216         vipanel.setBackground(new Color(0,0,64));
217         l20=new JLabel("Give a 4-digit Account number");
218         l20.setFont(f1);
219         l20.setForeground(Color.RED);
220         JPanel poppanel=new JPanel();
221         poppanel.add(l20);
222         poppanel.setBackground(new Color(0,0,64));
223         l20=new JLabel("Initial Amount");
224         l20.setFont(f1);
225         l20.setForeground(Color.RED);
226         JPanel kop=new JPanel();
227         kop.add(l20);
228         kop.setBackground(new Color(0,0,64));
229         bsubmit=new JButton("SUBMIT");
230         bsubmit.addActionListener(this);
231         JPanel dopanel=new JPanel();
232         dopanel.add(bsubmit);
233         dopanel.setBackground(new Color(0,0,64));
234         back=new JButton("BACK");
235         back.addActionListener(this);
236         JPanel dipanel=new JPanel();
237         dipanel.add(back);
238         dipanel.setBackground(new Color(0,0,64));
239         c.add(fpanel);c.add(apanel);
240         c.add(bpanel);c.add(tid1);
241         c.add(copanel);c.add(tid2);
242         c.add(dpanel);c.add(tid3);
243         c.add(epanel);c.add(tid4);
244         c.add(vpanel);c.add(tid5);
245         c.add(qpanel);c.add(gpanel);
246         c.add(wpanel);c.add(tid6);
247         c.add(opanel);c.add(tid7);
248         c.add(zpanel);c.add(kpanel);
249         c.add(xpanel);c.add(cpanel);
250         c.add(vxpanel);c.add(annualincome);
```

```
251         c.add(vppanel);c.add(profession);
252         c.add(npanel);c.add(gopanel);
253         c.add(mpanel);c.add(identity);
254         c.add(jpanel);c.add(tid8);
255         c.add(poppanel);c.add(t1);
256         c.add(kop);c.add(tid10);
257         c.add(vipanel);c.add(l16);
258         c.add(vopanel);c.add(l17);
259         c.add(dopanel);c.add(dipanel);
260         setSize(600,625);
261         setLocation(200,200);
262         setResizable(false);
263         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
264         setVisible(true);
265     }
266     public void actionPerformed(ActionEvent e) {
267          String name;
268          String name1;
269          String address;
270          String nationality;
271          String acc;
272          String gn;
273          String dob;
274          String date;
275          String time;
276          String Iden;
277          String prof;
278          String inti;
279          String ac1;
280          String land;
281          String mob;
282          String age;
283          String proof;
284          Create reg;
285          if(e.getSource()==bsubmit) {
286             name=tid1.getText();
287             name1=tid2.getText();
288             address=tid3.getText();
289             nationality=tid6.getText();
290             ac1=t1.getText();
291             ini = tid10.getText();
292             land = tid4.getText();
293             mob = tid5.getText();
294             age = tid7.getText();
295             proof = tid8.getText();
296             String namepattern = "^[A-Za-z]";
297             Scanner   scan  =  new Scanner( name ) ;
298             String   matched  =  scan.findInLine( namepattern ) ;
299             if ( matched == null ) {
300                  JOptionPane.showMessageDialog(this, "INVALID   FIRST NAME\nIt
    should contain only alphabet.", "Error", JOptionPane.ERROR_MESSAGE);
301                  tid1.setText("");
302              }
303              else
304                  flagName = true;
305              String namepattern1 = "^[A-Za-z]";
306              Scanner   scan1  =  new Scanner( name1 ) ;
307              String   matched1  =  scan1.findInLine( namepattern1 ) ;
308              if ( matched1 == null ) {
309                  JOptionPane.showMessageDialog(this, "INVALID   LAST NAME\nIt
    should contain only alphabet.", "Error", JOptionPane.ERROR_MESSAGE);
310                  tid2.setText("");
```

```
311             }
312             else
313                 flagName1 = true;
314             String namepattern2 = "^[A-Za-z0-9]{1}";
315             Scanner   scan2  =  new Scanner( address ) ;
316             String   matched2  =  scan2.findInLine( namepattern2 ) ;
317             if ( matched2 == null ) {
318                 JOptionPane.showMessageDialog(this, "INVALID ADDRESS\nIT SHOLUD
    NOT BLANK.", "Error", JOptionPane.ERROR_MESSAGE);
319                 tid3.setText("");
320             }
321             else
322                 flagAddress = true;
323             String landlinepattern = "^[0-9]{8}" ;
324             Scanner   scan6  =  new Scanner( land ) ;
325             String   matched6  =  scan6.findInLine( landlinepattern ) ;
326             if ( matched6 == null ) {
327                 JOptionPane.showMessageDialog(this, "INVALID LANDLINE NUMBER\nIt
    should contain only digits with minimum length of 8 digit.", "Error",
    JOptionPane.ERROR_MESSAGE);
328                 tid4.setText("");
329             }
330             else
331                 flagLandline    = true;
332             String mobpattern = "^[0-9]{10}" ;
333             Scanner   scan7  =  new Scanner( mob ) ;
334             String   matched7  =  scan7.findInLine( mobpattern ) ;
335             if ( matched7 == null ) {
336                 JOptionPane.showMessageDialog(this, "INVALID MOBILE NUMBER\nIt
    should contain only digits with minimum length of 10 digit.", "Error",
    JOptionPane.ERROR_MESSAGE);
337                 tid5.setText("");
338             }
339             else
340                 flagMobile    = true;
341             String namepattern3 = "^[A-Za-z]";
342             Scanner   scan3  =  new Scanner( nationality ) ;
343             String   matched3  =  scan3.findInLine( namepattern3 ) ;
344             if ( matched3 == null ) {
345                 JOptionPane.showMessageDialog(this, "INVALID   NATIONALITY\nIt
    should contain only alphabet.", "Error", JOptionPane.ERROR_MESSAGE);
346                 tid6.setText("");
347             }
348             else
349                 flagNationality = true;
350             String agepattern = "^[0-9]{2}" ;
351             Scanner   scan8  =  new Scanner( age ) ;
352             String   matched8  =  scan8.findInLine( agepattern ) ;
353             if ( matched8 == null ) {
354                 JOptionPane.showMessageDialog(this, "INVALID AGE\nAge must be
    minimum 10 years & it should contain only digit.", "Error",
    JOptionPane.ERROR_MESSAGE);
355                 tid7.setText("");
356             }
357             else
358                 flagAge     = true;
359             String proofpattern2 = "^[A-Za-z0-9]{5}";
360             Scanner   scan9  =  new Scanner( proof ) ;
361             String   matched9  =  scan9.findInLine( proofpattern2 ) ;
362             if ( matched9 == null ) {
363                 JOptionPane.showMessageDialog(this, "INVALID IDENTITY PROOF
    NUMBER\nIT SHOLUD CONTAIN MINIMUM 5 ALPHABET OR DIGIT.", "Error",
```

```
    JOptionPane.ERROR_MESSAGE);
364                 tid8.setText("");
365             }
366             else
367                 flagProof  = true;
368             String accpattern = "^[0-9]{4}" ;
369             Scanner   scan4  =  new Scanner( ac1 ) ;
370             String   matched4  =  scan4.findInLine( accpattern ) ;
371             if ( matched4 == null ) {
372                 JOptionPane.showMessageDialog(this, "INVALID ACCOUNT NUMBER\nIt
    should contain only digits with minimum length of 4 digit.", "Error",
    JOptionPane.ERROR_MESSAGE);
373                 t1.setText("");
374             }
375             else
376                 flagAccountNumber  = true;
377             String inipattern = "^[0-9]{3}" ;
378             Scanner   scan5  =  new Scanner( ini ) ;
379             String   matched5  =  scan5.findInLine( inipattern ) ;
380             if ( matched5 == null ) {
381                 JOptionPane.showMessageDialog(this, "INVALID INITIAL AMOUNT\nIt
    should contain only digits with minimum amount of INR 500.", "Error",
    JOptionPane.ERROR_MESSAGE);
382                 tid10.setText("");
383             }
384             else
385                 flagInitialAmount   = true;
386             if(flagName == true) {
387                 if(flagName1 == true) {
388                     if(flagAddress == true) {
389                         if(flagNationality == true) {
390                             if(flagAccountNumber == true) {
391                                 if(flagInitialAmount == true) {
392                                     if(flagLandline == true) {
393                                         if(flagMobile == true) {
394                                             if(flagAge == true) {
395                                                 if(flagProof == true) {
396                                                     try {
397                                                         FileInputStream fin=new
    FileInputStream("Regis.dat");
398                                                         ObjectInputStream oin=new
    ObjectInputStream(fin);
399
    alist=(ArrayList<Create>)oin.readObject();
400                                                     }catch(Exception ex) {
401                                                         alist=new
    ArrayList<Create>();
402                                                     }
403                                                     for(Create element : alist) {
404
    if(element.getAc().equals(t1.getText()))
405                                                         count++;
406                                                     }
407                                                     if(count > 0) {
408
    JOptionPane.showMessageDialog(this, "Account numner is already registered.\nTry
    another one", "Error", JOptionPane.ERROR_MESSAGE);
409                                                         t1.setText("");
410                                                     }
411                                                     else {
412                                                         if(Integer.parseInt(ini)
    >= 500) {
```

```
413                                                                 inti=tid10.getText();
414                                                                 name=tid1.getText();
415                                                                 name1=tid2.getText();
416
    address=tid3.getText();
417
    nationality=tid6.getText();
418                                                                 ac1=t1.getText();
419                                                                 Iden=
    (String)identity.getSelectedItem();
420                                                                 prof=
    (String)profession.getSelectedItem();
421                                                                 acc="";
422                                                                 if(rsav.isSelected())
423                                                                     acc="Savings";
424                                                                 else
    if(rothers.isSelected())
425                                                                         acc="Others";
426                                                                 gn="";
427                                                                if(rmale.isSelected())
428                                                                    gn="Male";
429                                                                else
    if(rfemale.isSelected())
430                                                                    gn="Female";
431                                                                String d=
    (String)day.getSelectedItem();
432                                                                String m=
    (String)month.getSelectedItem();
433                                                                String y=
    (String)year.getSelectedItem();
434                                                                dob=d + "-" + m + "-" +
    y;
435                                                                reg=new Create();
436                                                                reg.setName(name);
437                                                                reg.setName1(name1);
438
                        reg.setAddress(address);
439
                        reg.setNationality(nationality);
440                                                                reg.setAcctype(acc);
441                                                                reg.setGender(gn);
442                                                                reg.setDob(dob);
443                                                                reg.setIdentity(Iden);
444
                        reg.setProfession(prof);
445                                                                reg.setAc(ac1);
446
                        reg.setInitialamnt(inti);
447
                        reg.setDate(l16.getText());
448
                        reg.setTime(l17.getText());
449                                                                int
    con=JOptionPane.showConfirmDialog(this, "Are You Sure to Register?");
450
                        if(con==JOptionPane.YES_OPTION) {
451                                                                    new
    AddInformation1(reg);
452                                                                    new
    Actotalcreate(Integer.parseInt(t1.getText()),0,Integer.parseInt(tid10.getText()),In
    teger.parseInt(tid10.getText()));
453                                                                    new
```

```
   TotalUpdate(Integer.parseInt(t1.getText()),Integer.parseInt(tid10.getText()));
454
                            JOptionPane.showMessageDialog(this, "Successfully
   registered");
455                                                                 new
   Individualpassbook(ac1);
456                                                                 tid10.setText("");
457                                                                 tid1.setText("");
458                                                                 tid2.setText("");
459                                                                 tid3.setText("");
460                                                                 tid6.setText("");
461                                                                 t1.setText("");
462                                                                 tid4.setText("");
463                                                                 tid5.setText("");
464                                                                 tid7.setText("");
465                                                                 tid8.setText("");
466                                                             }
467                                                         }
468                                                         else {
469
   JOptionPane.showMessageDialog(this, "Minimum initial amount should be 500",
   "Error", JOptionPane.ERROR_MESSAGE);
470                                                             tid10.setText("");
471                                                         }
472                                                     }
473                                                 }
474                                             }
475                                         }
476                                     }
477                                 }
478                             }
479                         }
480                     }
481                 }
482             }
483             flagName = false;
484             flagName1 = false;
485             flagAddress = false;
486             flagNationality = false;
487             flagAccountNumber = false;
488             flagInitialAmount = false;
489             count =0;
490             flagLandline = false;
491             flagMobile = false;
492             flagAge = false;
493             flagProof = false;
494         }
495         if(e.getSource()==back) {
496             int rply = JOptionPane.showConfirmDialog(this, "Are you sure to
   quit?");
497             if(rply == JOptionPane.YES_OPTION) {
498                 new Userwindow("");
499                 setVisible(false);
500             }
501         }
502     }
503 }
504
```

```java
package test;
import java.io.Serializable;
public class Depo extends JFrame implements Serializable {
    private String savingsacc;
    private String fixeddepoacc;
    private String savingsaccbal;
    private String fixedamnt;
    private String terms;
    private String rate;
    private String date;
    private String time;
    public String getDate() {
        return date;
    }
    public void setDate(String date) {
        this.date = date;
    }
    public String getTime() {
        return time;
    }
    public void setTime(String time) {
        this.time = time;
    }
    public String getSavingsacc() {
        return savingsacc;
    }
    public void setSavingsacc(String savingsacc) {
        this.savingsacc = savingsacc;
    }
    public String getFixeddepoacc() {
        return fixeddepoacc;
    }
    public void setFixeddepoacc(String fixeddepoacc) {
        this.fixeddepoacc = fixeddepoacc;
    }
    public String getSavingsaccbal() {
        return savingsaccbal;
    }
    public void setSavingsaccbal(String savingsaccbal) {
        this.savingsaccbal = savingsaccbal;
    }
    public String getFixedamnt() {
        return fixedamnt;
    }
    public void setFixedamnt(String fixedamnt) {
        this.fixedamnt = fixedamnt;
    }
    public String getTerms() {
        return terms;
    }
    public void setTerms(String terms) {
        this.terms = terms;
    }
    public String getRate() {
            return rate;
    }
    public void setRate(String rate) {
        this.rate = rate;
    }
}

```

```java
package test;
import java.util.*;
public class DepoCreate extends JFrame {
    private String sysdate,systime;
    DepoCreate(String a,String b,String c,String d,String e,String f) {
        Calendar cal = Calendar.getInstance();
        String cday = ""+cal.get(Calendar.DATE);
        int x =cal.get(Calendar.MONTH);
        String cmonth = ""+(x+1);
        String cyear =""+cal.get(Calendar.YEAR);
        sysdate = cday+"/"+cmonth+"/"+cyear;
        String chr = ""+cal.get(Calendar.HOUR_OF_DAY);
        String cmin = ""+cal.get(Calendar.MINUTE);
        String csec =""+cal.get(Calendar.SECOND);
        systime = chr+":"+cmin+":"+csec;
        Depo sd1=new Depo();
        sd1.setSavingsacc(a);
        sd1.setFixeddepoacc(b);
        sd1.setSavingsaccbal(c);
        sd1.setFixedamnt(d);
        sd1.setTerms(e);
        sd1.setRate(f);
        sd1.setDate(sysdate);
        sd1.setTime(systime);
        ArrayList<Depo> list4;
        try {
            FileInputStream fin=new FileInputStream("fixed.dat");
            ObjectInputStream oin=new ObjectInputStream(fin);
            list4=(ArrayList<Depo>)oin.readObject();
        }catch(Exception e1) {
            list4=new ArrayList<Depo>();
        }
        list4.add(sd1);
        try {
            FileOutputStream fout=new FileOutputStream("fixed.dat");
            ObjectOutputStream oout=new ObjectOutputStream(fout);
            oout.writeObject(list4);
        }catch(Exception e1){}
    }
}
```

```java
package test;
import java.awt.event.*;
public class Deposit extends JFrame implements ActionListener {
    private JLabel l1,l2,l3;
    private JTextField t1;
    private JButton submit,cancel;
    public Deposit() {
        Container c=getContentPane();
        c.setLayout(new GridLayout(3,2));
        Font f1=new Font("Times New Roman",Font.BOLD,20);
        l1=new JLabel("                          DEPOSIT");
        l1.setFont(f1);
        l1.setForeground(Color.GRAY);
        JPanel fpanel=new JPanel();
        fpanel.add(l1);
        fpanel.setBackground(new Color(0,0,64));
        l1=new JLabel("SLIP                              ");
        l1.setFont(f1);
        l1.setForeground(Color.GRAY);
        JPanel apanel=new JPanel();
        apanel.add(l1);
        apanel.setBackground(new Color(0,0,64));
        l1=new JLabel("Enter ACCOUNT NO:");
        l1.setFont(f1);
        l1.setForeground(Color.GRAY);
        JPanel bpanel=new JPanel();
        bpanel.add(l1);
        bpanel.setBackground(new Color(0,0,64));
        t1=new JTextField();
        submit=new JButton("SUBMIT");
        submit.addActionListener(this);
        JPanel dpanel=new JPanel();
        dpanel.add(submit);
        dpanel.setBackground(new Color(0,0,64));
        cancel=new JButton("CANCEL");
        cancel.addActionListener(this);
        JPanel epanel=new JPanel();
        epanel.add(cancel);
        epanel.setBackground(new Color(0,0,64));
        c.add(fpanel);c.add(apanel);
        c.add(bpanel);c.add(t1);
        c.add(dpanel);c.add(epanel);
        setSize(500,400);
        setLocation(200,200);
        setResizable(false);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }
    public void actionPerformed(ActionEvent e) {
        if(e.getSource()==submit) {
            new Deposit1();
            setVisible(false);
        }
        if(e.getSource()==cancel) {
            new Userwindow("");
            setVisible(false);
        }
    }
}
```

```java
package test;
import java.awt.event.*;
public class Deposit1 extends JFrame implements ActionListener {
    private JLabel l1,l2,l3,l4,l5;
    private JTextField t1,t2,t3;
    private JButton proceed,back;
    public Deposit1() {
        Container c=getContentPane();
        c.setLayout(new GridLayout(5,2));
        Font f1=new Font("Times New Roman",Font.BOLD,20);
        l1=new JLabel("                    WITHDRAWAL");
        l1.setFont(f1);
        l1.setForeground(Color.WHITE);
        JPanel fpanel=new JPanel();
        fpanel.add(l1);
        fpanel.setBackground(new Color(0,0,64));
        l2=new JLabel("DETAILS           ");
        l2.setFont(f1);
        l2.setForeground(Color.WHITE);
        JPanel apanel=new JPanel();
        apanel.add(l2);
        apanel.setBackground(new Color(0,0,64));
        l3=new JLabel("Account Holder Name");
        l3.setFont(f1);
        l3.setForeground(Color.GRAY);
        JPanel bpanel=new JPanel();
        bpanel.add(l3);
        bpanel.setBackground(new Color(0,0,64));
        l4=new JLabel("BALANCE IN ACCOUNT");
        l4.setFont(f1);
        l4.setForeground(Color.GRAY);
        JPanel cpanel=new JPanel();
        cpanel.add(l4);
        cpanel.setBackground(new Color(0,0,64));
        l5=new JLabel("Amount To Be Deposited");
        l5.setFont(f1);
        l5.setForeground(Color.GRAY);
        JPanel dpanel=new JPanel();
        dpanel.add(l5);
        dpanel.setBackground(new Color(0,0,64));
        t1=new JTextField();
        t2=new JTextField();
        t3=new JTextField();
        proceed=new JButton("PROCEED");
        proceed.addActionListener(this);
        JPanel epanel=new JPanel();
        epanel.add(proceed);
        epanel.setBackground(new Color(0,0,64));
        back=new JButton("BACK");
        back.addActionListener(this);
        JPanel gpanel=new JPanel();
        gpanel.add(back);
        gpanel.setBackground(new Color(0,0,64));
        c.add(fpanel);c.add(apanel);
        c.add(bpanel);c.add(t1);
        c.add(cpanel);c.add(t2);
        c.add(dpanel);c.add(t3);
        c.add(epanel);c.add(gpanel);
        setSize(500,400);
        setLocation(200,200);
        setResizable(false);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
65          setVisible(true);
66      }
67      public void actionPerformed(ActionEvent e) {
68          if(e.getSource()==proceed) {
69              int con=JOptionPane.showConfirmDialog(this, "Are You Sure to Deposit?");
70              if(con==JOptionPane.YES_OPTION) {
71                  setVisible(false);
72              }
73          }
74          if(e.getSource()==back) {
75              new Deposit();
76              setVisible(false);
77          }
78      }
79  }
```

```java
package test;
import javax.swing.*;
public class DiaplayAll extends JFrame {
    public DiaplayAll() {
        super("DISPLAY All USERS INFORMATION");
        String heading[]={"First Name","Last name","Address","Nationality","Account Type","Date Of Birth","Identity","Profession","Reg Date","Reg Time","A/c no","Balance"};
        String data[][];
        ArrayList<Create> list;
        try {
            FileInputStream fin=new FileInputStream("Regis.dat");
            ObjectInputStream oin=new ObjectInputStream(fin);
            list=(ArrayList<Create>)oin.readObject();
            data = new String[list.size()][heading.length+1];
            int r=0;
            for(Create re : list) {
                data[r][0]=re.getName();
                data[r][1]=re.getName1();
                data[r][2]=re.getAddress();
                data[r][3]=re.getNationality();
                data[r][4]=re.getAcctype();
                data[r][5]=re.getDob();
                data[r][6]=re.getIdentity();
                data[r][7]=re.getProfession();
                data[r][8]=re.getDate();
                data[r][9]=re.getTime();
                data[r][10]=re.getAc();
                data[r][11]=re.getInitialamnt();
                r++;
            }
            Container con=getContentPane();
            con.setLayout(new BorderLayout());
            JTable datatable=new JTable(data, heading);
            JScrollPane jsp=new JScrollPane(datatable);
            con.add(new JLabel("All Registration Details"),BorderLayout.NORTH);
            con.add(jsp,BorderLayout.CENTER);
            setSize(650, 300);
            setLocation(200, 200);
            setVisible(true);
        }catch(Exception e) {
            JOptionPane.showMessageDialog(this, "No file found in data base", "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
}

```

```java
package test;
import javax.swing.*;
public class Dispwithdraw extends JFrame {
    String heading[]={"First Name","Last name","Address","Nationality","Account Type","Date Of Birth","Identity","Profession","Reg Date","Reg Time","A/c no","Initial amnt"};
    public Dispwithdraw(String b[][]) {
        Container con=getContentPane();
        con.setLayout(new BorderLayout());
        JTable datatable=new JTable(b, heading);
        JScrollPane jsp=new JScrollPane(datatable);
        con.add(new JLabel("Registration Details"),BorderLayout.NORTH);
        con.add(jsp,BorderLayout.CENTER);
        setSize(650, 650);
        setLocation(600, 200);
        setVisible(true);
    }
}
```

```java
package test;
import javax.swing.*;
public class Individualpassbook extends JFrame {
    String heading[]={"Accno","Withdraw","Deposit","Total","Date","Time"};
    String data[][];
    ArrayList<Actotal> list;
    public Individualpassbook(String name){
        super("Search for " + name);
        int acc=Integer.parseInt(name);
        try {
            FileInputStream fin=new FileInputStream("actotal.dat");
            ObjectInputStream oin=new ObjectInputStream(fin);
            list=(ArrayList<Actotal>)oin.readObject();
        }catch(Exception e1) {
            JOptionPane.showMessageDialog(this, "No file found in data base",
  "Error", JOptionPane.ERROR_MESSAGE);
        }
        int row=0;
        for(Actotal re : list) {
            if(re.getAccount()==(acc)){
                row++;
            }
        }
        data = new String[row][heading.length+1];
        int r=0;
        for(Actotal re : list) {
            if(re.getAccount()==(acc)) {
                data[r][0]=String.valueOf(re.getAccount());
                data[r][1]=String.valueOf(re.getWithdraw1());
                data[r][2]=String.valueOf(re.getDepo());
                data[r][3]=String.valueOf(re.getTotal());
                data[r][4]=re.getDate();
                data[r][5]=re.getTime();
                r++;
            }
        }
        Container con=getContentPane();
        con.setLayout(new BorderLayout());
        JTable datatable=new JTable(data,heading);
        JScrollPane jsp=new JScrollPane(datatable);
        con.add(new JLabel("PASSBOOK DETAILS"),BorderLayout.NORTH);
        con.add(jsp,BorderLayout.CENTER);
        setSize(650, 300);
        setLocation(200,200);
        setVisible(true);
    }
}
```

```java
package test;
import java.awt.event.*;
class FirstWindow extends JFrame implements ActionListener {
    private JLabel l0,l1,l2,l3;
    private JButton admin,user;
    public FirstWindow(String title) {
        super("WELCOME TO UNITED BANK OF PARTNERS");
        Container c=getContentPane();
        c.setLayout(new GridLayout(3,2));
        Font f1=new Font("Times New Roman",Font.BOLD,25);
        l0=new JLabel("WELCOME TO UNITED");
        l0.setFont(f1);
        l0.setForeground(Color.WHITE);
        JPanel fpanel=new JPanel();
        fpanel.add(l0);
        fpanel.setBackground(new Color(0,0,64));
        l1=new JLabel("BANK OF PARTNERS      ");
        l1.setFont(f1);
        l1.setForeground(Color.WHITE);
        JPanel spanel=new JPanel();
        spanel.add(l1);
        spanel.setBackground(new Color(0,0,64));
        l3=new JLabel("AS:                                                  ");
        l3.setFont(f1);
        l3.setForeground(Color.WHITE);
        JPanel xpanel=new JPanel();
        xpanel.add(l3);
        xpanel.setBackground(new Color(0,0,64));
        Font f2=new Font("Times New Roman",Font.BOLD,25);
        l2=new JLabel("                                    LOGIN");
        l2.setFont(f2);
        l2.setForeground(Color.WHITE);
        JPanel ppanel=new JPanel();
        ppanel.add(l2);
        ppanel.setBackground(new Color(0,0,64));
        admin=new JButton("ADMINISTRATOR");
        JPanel gpanel=new JPanel();
        gpanel.add(admin);
        gpanel.setBackground(new Color(0,0,64));
        admin.addActionListener(this);
        gpanel.add(new JLabel(""));
        user=new JButton("OPERATOR");
        JPanel qpanel=new JPanel();
        qpanel.add(user);
        qpanel.setBackground(new Color(0,0,64));
        user.addActionListener(this);
        qpanel.add(user);
        c.add(fpanel);c.add(spanel);
        c.add(ppanel);c.add(xpanel);
        c.add(gpanel);c.add(qpanel);
        setSize(600,450);
        setLocation(200,200);
        setResizable(false);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }
    public void actionPerformed(ActionEvent e) {
        if(e.getSource()==admin) {
            new SecondWindow() ;
            setVisible(false);
        }
        if(e.getSource()==user) {
```

```
65              new ThirdWindow() ;
66              setVisible(false);
67          }
68      }
69  }
70  public class Main {
71      public static void main(String[] args) {
72          new FirstWindow("WELCOME TO UNITED BANK OF PARTNERS");
73      }
74  }
75
```

```java
package test;
import java.awt.event.*;

class SecondWindow extends JFrame implements ActionListener {
    private JLabel l0,l1,l2,l3;
    private JTextField t1;
    private JPasswordField p1;
    private JButton login,cancel;
    private String sysDate,sysTime;
    private boolean flagID = false;
    private boolean flagPass = false;
    public SecondWindow() {
        super("Login as Administrator");
        Container c=getContentPane();
        c.setLayout(new GridLayout(4,2));
        Font f1=new Font("Times New Roman",Font.BOLD,20);
        l1=new JLabel("                ADMINISTRATOR");
        l1.setFont(f1);
        l1.setForeground(Color.RED);
        JPanel fpanel=new JPanel();
        fpanel.add(l1);
        fpanel.setBackground(new Color(0,0,64));
        l0=new JLabel("LOGIN                                ");
        l0.setFont(f1);
        l0.setForeground(Color.RED);
        JPanel epanel=new JPanel();
        epanel.add(l0);
        epanel.setBackground(new Color(0,0,64));
        l2=new JLabel("USERNAME");
        l2.setFont(f1);
        l2.setForeground(Color.RED);
        JPanel apanel=new JPanel();
        apanel.add(l2);
        apanel.setBackground(new Color(0,0,64));
        l3=new JLabel("PASSWORD");
        l3.setFont(f1);
        l3.setForeground(Color.RED);
        JPanel bpanel=new JPanel();
        bpanel.add(l3);
        bpanel.setBackground(new Color(0,0,64));
        t1=new JTextField();
        p1=new JPasswordField();
        login=new JButton("LOGIN");
        login.addActionListener(this);
        JPanel cpanel=new JPanel();
        cpanel.add(login);
        cpanel.setBackground(new Color(0,0,64));
        cancel=new JButton("CANCEL");
        cancel.addActionListener(this);
        JPanel dpanel=new JPanel();
        dpanel.add(cancel);
        dpanel.setBackground(new Color(0,0,64));
        Calendar cal = Calendar.getInstance();
        String cday = ""+cal.get(Calendar.DATE);
        int x =cal.get(Calendar.MONTH);
        String cmonth = ""+(x+1);
        String cyear =""+cal.get(Calendar.YEAR);
        sysDate = cday+"/"+cmonth+"/"+cyear;
        String chr = ""+cal.get(Calendar.HOUR_OF_DAY);
        String cmin = ""+cal.get(Calendar.MINUTE);
        String csec =""+cal.get(Calendar.SECOND);
        sysTime = chr+":"+cmin+":"+csec;
```

```
66          c.add(fpanel);c.add(epanel);
67          c.add(apanel);c.add(t1);
68          c.add(bpanel);c.add(p1);
69          c.add(cpanel);c.add(dpanel);
70          setSize(500,425);
71          setLocation(200,200);
72          setResizable(false);
73          setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
74          setVisible(true);
75      }
76      public void actionPerformed(ActionEvent e) {
77          if(e.getSource()==login) {
78              String n = t1.getText().trim();
79              String ps = p1.getText().trim();
80              String namepattern = "^[A-Za-z]";
81              Scanner   scan  =  new Scanner( n ) ;
82              String   matched  =  scan.findInLine( namepattern ) ;
83              if ( matched == null ) {
84                  JOptionPane.showMessageDialog(this, "INVALID USER NAME\nIt should
    contain only alphabet.", "Error", JOptionPane.ERROR_MESSAGE);
85                  t1.setText("");
86              }
87              else {
88                  flagID = true;
89              }
90              String passwordpattern = "^[0-9]{4}" ;
91              Scanner   scan2  =  new Scanner( ps ) ;
92              String   matched2  =  scan2.findInLine( passwordpattern ) ;
93              if ( matched2 == null ) {
94                  JOptionPane.showMessageDialog(this, "INVALID PASSWORD\nIt should
    contain only digits with minimum password length of 4 digit.", "Error",
    JOptionPane.ERROR_MESSAGE);
95                  p1.setText("");
96              }
97              else
98                  flagPass = true;
99              if(flagID == true )  {
100                 if(flagPass == true) {
101                     if(t1.getText().equals("admin") && p1.getText().equals("1234"))
    {
102                         JOptionPane.showMessageDialog(this, "Welcome
    Administrator");
103                         new Admin() ;
104                         setVisible(false);
105                     }
106                     else {
107                         JOptionPane.showMessageDialog(this, "Wrong
    username/password...\nTry again", "Error", JOptionPane.ERROR_MESSAGE);
108                         t1.setText("");
109                         p1.setText("");
110                     }
111                 }
112             }
113             flagID = false;
114             flagPass = false;
115         }
116         if(e.getSource()==cancel) {
117             int con=JOptionPane.showConfirmDialog(this, "Are you sure to cancel?");
118             if(con==JOptionPane.YES_OPTION) {
119                 new FirstWindow("");
120                 setVisible(false);
121             }
```

```
        }
    }
}
public class Main1 {
    public static void main(String[] args) {}
}

```

```java
package test;
import java.io.*;
public class OperatorAddInfo {
    ArrayList<OperatorRegData> list4;
    public OperatorAddInfo(OperatorRegData rg1) {
        try {
            FileInputStream fin=new FileInputStream("Reg.dat");
            ObjectInputStream oin=new ObjectInputStream(fin);
            list4=(ArrayList<OperatorRegData>)oin.readObject();
        }catch(Exception e) {
            list4=new ArrayList<OperatorRegData>();
        }
        list4.add(rg1);
        try {
            FileOutputStream fout=new FileOutputStream("Reg.dat");
            ObjectOutputStream oout=new ObjectOutputStream(fout);
            oout.writeObject(list4);
        }catch(Exception e){}
    }
}

```

```java
package test;
import java.awt.event.*;
public class OperatorDelete extends JFrame implements ActionListener {
    private JLabel l1,l2,l3;
    private JTextField t1;
    private JButton submit,back;
    private boolean flag = false;
    public OperatorDelete() {
        super("Operator Delete");
        Container c=getContentPane();
        c.setLayout(new GridLayout(3,2));
        Font f1=new Font("Times New Roman",Font.BOLD,20);
        l1=new JLabel("OPERATOR");
        l1.setFont(f1);
        l1.setForeground(Color.RED);
        JPanel fpanel=new JPanel();
        fpanel.add(l1);
        fpanel.setBackground(new Color(0,0,64));
        l2=new JLabel("Delete");
        l2.setFont(f1);
        l2.setForeground(Color.RED);
        JPanel apanel=new JPanel();
        apanel.add(l2);
        apanel.setBackground(new Color(0,0,64));
        l3=new JLabel("Give OPERATOR First name");
        l3.setFont(f1);
        l3.setForeground(Color.RED);
        JPanel bpanel=new JPanel();
        bpanel.add(l3);
        bpanel.setBackground(new Color(0,0,64));
        t1=new JTextField();
        submit=new JButton("Delete");
        submit.addActionListener(this);
        JPanel cpanel=new JPanel();
        cpanel.add(submit);
        cpanel.setBackground(new Color(0,0,64));
        back=new JButton("BACK");
        back.addActionListener(this);
        JPanel dpanel=new JPanel();
        dpanel.add(back);
        dpanel.setBackground(new Color(0,0,64));
        c.add(fpanel);c.add(apanel);
        c.add(bpanel);c.add(t1);
        c.add(cpanel);c.add(dpanel);
        setSize(550,300);
        setLocation(200,200);
        setResizable(false);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }
    public void actionPerformed(ActionEvent e) {
        if(e.getSource() == submit) {
            String n = t1.getText().trim();
            String namepattern = "^[A-Za-z]";
            Scanner   scan   =   new Scanner( n ) ;
            String   matched   =   scan.findInLine( namepattern ) ;
            if ( matched == null ) {
                JOptionPane.showMessageDialog(this, "INVALID    OPERATOR NAME\nIt should starts with alphabet.", "Error", JOptionPane.ERROR_MESSAGE);
                t1.setText("");
            }
            else
```

```
65                  flag  = true;
66              if(flag == true) {
67                  String sname = t1.getText().trim();
68                  new OperatorDeleteDisplay(sname);
69                  t1.setText("");
70              }
71              flag = false;
72          }
73          if(e.getSource()==back) {
74              int con=JOptionPane.showConfirmDialog(this, "Are You Sure you want to
  cancel?");
75              if(con==JOptionPane.YES_OPTION) {
76                  new Admin();
77                  setVisible(false);
78              }
79          }
80      }
81  }
82
```

```java
package test;
import java.io.*;
public class OperatorDeleteDisplay extends JFrame {
    ArrayList<OperatorRegData> list1;
    public OperatorDeleteDisplay(String name) {
        try {
            FileInputStream fin=new FileInputStream("Reg.dat");
            ObjectInputStream oin=new ObjectInputStream(fin);
            list1=(ArrayList<OperatorRegData>)oin.readObject();
            int r=0;
            for(int i=0;i<list1.size();) {
                if(list1.get(i).getName().equals(name)) {
                    list1.remove(i);
                    r++;
                }
                else
                    i++;
            }
            if(r==0)
                JOptionPane.showMessageDialog(this, "Match not Found");
            else {
                try  {
                    FileOutputStream fout=new FileOutputStream("Reg.dat");
                    ObjectOutputStream oout=new ObjectOutputStream(fout);
                    oout.writeObject(list1);
                }catch(Exception e){}
                JOptionPane.showMessageDialog(this, "Successfully deleted");
            }
        }
        catch(Exception e) {
            JOptionPane.showMessageDialog(this, "No file found in data base",
  "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
}


```

```java
package test;
import javax.swing.*;
public class OperatorDisplay extends JFrame {
    public OperatorDisplay() {
        super("DISPLAY All OPERATORS INFORMATION");
        String heading[]={"First Name","Last name","Password","Address","City","Gender","Date Of Birth","Reg Date","Reg Time"};
        String data[][];
        ArrayList<OperatorRegData> list;
        try {
            FileInputStream fin=new FileInputStream("Reg.dat");
            ObjectInputStream oin=new ObjectInputStream(fin);
            list=(ArrayList<OperatorRegData>)oin.readObject();
            data = new String[list.size()][heading.length+1];
            int r=0;
            for(OperatorRegData re : list) {
                data[r][0]=re.getName();
                data[r][1]=re.getName1();
                data[r][2]=re.getPassword();
                data[r][3]=re.getAddress();
                data[r][4]=re.getCity();
                data[r][5]=re.getGender();
                data[r][6]=re.getDob();
                data[r][7]=re.getDate();
                data[r][8]=re.getTime();
                r++;
            }
            Container con=getContentPane();
            con.setLayout(new BorderLayout());
            JTable datatable=new JTable(data, heading);
            JScrollPane jsp=new JScrollPane(datatable);
            con.add(new JLabel("All Registration Details"),BorderLayout.NORTH);
            con.add(jsp,BorderLayout.CENTER);
            setSize(650, 300);
            setLocation(200, 200);
            setVisible(true);
        }catch(Exception e) {
            JOptionPane.showMessageDialog(this, "No file found in data base", "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
}
```

```java
package test;
import java.awt.*;
public class OperatorReg extends JFrame implements ActionListener {
    private JLabel l0,l1,l2,l3,l4,l5,l6,l7,l8,l9,l10,l11,l12,l;
    private JTextField tid,tod;
    private JPasswordField tpass;
    private JTextArea tadd;
    private JComboBox city,day,month,year,annualincome;
    private JCheckBox cmo,cmu,cco;
    private JRadioButton rmale,rfemale;
    private JButton bsubmit,back;
    private String sysdate,systime;
    private boolean flagName=false, flagName1 = false, flagPass=false, flagAdd = false;
    private ArrayList<OperatorRegData> alist = new ArrayList<OperatorRegData>();
    public OperatorReg(String title) {
        super("Operator registration");
        Container c=getContentPane();
        c.setLayout(new GridLayout(12,2));
        tid=new JTextField();
        tod=new JTextField();
        tpass=new JPasswordField();
        tadd=new JTextArea(5,20);
        JScrollPane tapan=new JScrollPane(tadd);
        String cvalue[]={"<50000","50000-100000","100000-600000",">600000"};
        annualincome=new JComboBox(cvalue);
        String ctvalue[]={"Kolkata","Delhi","Mumbai","Chennai","Hydrabad","Pune","Bangalore"};
        city=new JComboBox(ctvalue);
        rmale=new JRadioButton("Male");
        rfemale=new JRadioButton("Female");
        ButtonGroup rgroup=new ButtonGroup();
        rgroup.add(rmale);
        rgroup.add(rfemale);
        JPanel gpanel=new JPanel();
        gpanel.add(rmale);
        gpanel.add(rfemale);
        String dvalue[]=new String[31];
        for(int i=0;i<=30;i++)
            dvalue[i]=String.valueOf(i+1);
        day=new JComboBox(dvalue);
        String mvalue[]=new String[12];
        for(int i=0;i<=11;i++)
            mvalue[i]=String.valueOf(i+1);
        month=new JComboBox(mvalue);
        String yvalue[]=new String[25];
        int cnt=0;
        for(int i=1989;i<=2013;i++) {
            yvalue[cnt]=String.valueOf(i);
            cnt++;
        }
        year=new JComboBox(yvalue);
        JPanel cpanel=new JPanel();
        cpanel.add(day);
        cpanel.add(month);
        cpanel.add(year);
        bsubmit=new JButton("Register");
        bsubmit.addActionListener(this);
        back=new JButton("BACK");
        back.addActionListener(this);
        Font f=new Font("TIMES NEW ROMAN",Font.BOLD,27);
        l1=new JLabel("OPERATOR");
```

```
64          l1.setFont(f);
65          l1.setForeground(Color.BLUE);
66          JPanel apanel=new JPanel();
67          apanel.add(l1);
68          apanel.setBackground(new Color(0,0,64));
69          l0=new JLabel("REGISTRATION");
70          l0.setFont(f);
71          l0.setForeground(Color.BLUE);
72          JPanel bpanel=new JPanel();
73          bpanel.add(l0);
74          bpanel.setBackground(new Color(0,0,64));
75          Font f1=new Font("comic sans ms",Font.BOLD,14);
76          l2=new JLabel("Enter First Name");
77          l2.setFont(f1);
78          l2.setForeground(Color.RED);
79          JPanel dpanel=new JPanel();
80          dpanel.add(l2);
81          dpanel.setBackground(new Color(0,0,64));
82          l=new JLabel("Enter Last Name");
83          l.setFont(f1);
84          l.setForeground(Color.RED);
85          JPanel qpanel=new JPanel();
86          qpanel.add(l);
87          qpanel.setBackground(new Color(0,0,64));
88          l3=new JLabel("Enter Password");
89          l3.setFont(f1);
90          l3.setForeground(Color.RED);
91          JPanel epanel=new JPanel();
92          epanel.add(l3);
93          epanel.setBackground(new Color(0,0,64));
94          l4=new JLabel("Enter Address");
95          l4.setFont(f1);
96          l4.setForeground(Color.RED);
97          JPanel fpanel=new JPanel();
98          fpanel.add(l4);
99          fpanel.setBackground(new Color(0,0,64));
100         l5=new JLabel("Select City");
101         l5.setFont(f1);
102         l5.setForeground(Color.RED);
103         JPanel hpanel=new JPanel();
104         hpanel.add(l5);
105         hpanel.setBackground(new Color(0,0,64));
106         l6=new JLabel("Select Gender");
107         l6.setFont(f1);
108         l6.setForeground(Color.RED);
109         JPanel ipanel=new JPanel();
110         ipanel.add(l6);
111         ipanel.setBackground(new Color(0,0,64));
112         l7=new JLabel("Select DOB");
113         l7.setFont(f1);
114         l7.setForeground(Color.RED);
115         JPanel jpanel=new JPanel();
116         jpanel.add(l7);
117         jpanel.setBackground(new Color(0,0,64));
118         l8=new JLabel("Salary");
119         l8.setFont(f1);
120         l8.setForeground(Color.RED);
121         JPanel kpanel=new JPanel();
122         kpanel.add(l8);
123         kpanel.setBackground(new Color(0,0,64));
124         l9=new JLabel("Current Date");
125         l9.setFont(f1);
```

```
126         l9.setForeground(Color.RED);
127         JPanel npanel=new JPanel();
128         npanel.add(l9);
129         npanel.setBackground(new Color(0,0,64));
130         Calendar cal = Calendar.getInstance();
131         String cday = ""+cal.get(Calendar.DATE);
132         int x =cal.get(Calendar.MONTH);
133         String cmonth = ""+(x+1);
134         String cyear =""+cal.get(Calendar.YEAR);
135         sysdate = cday+"/"+cmonth+"/"+cyear;
136         l11=new JLabel(sysdate);
137         String chr = ""+cal.get(Calendar.HOUR_OF_DAY);
138         String cmin = ""+cal.get(Calendar.MINUTE);
139         String csec =""+cal.get(Calendar.SECOND);
140         systime = chr+":"+cmin+":"+csec;
141         l12=new JLabel(systime);
142         l10=new JLabel("Current Time");
143         l10.setFont(f1);
144         l10.setForeground(Color.RED);
145         JPanel opanel=new JPanel();
146         opanel.add(l10);
147         opanel.setBackground(new Color(0,0,64));
148         c.add(apanel);c.add(bpanel);
149         c.add(dpanel);c.add(tid);
150         c.add(qpanel);c.add(tod);
151         c.add(epanel);c.add(tpass);
152         c.add(fpanel);c.add(tapan);
153         c.add(hpanel);c.add(city);
154         c.add(ipanel);c.add(gpanel);
155         c.add(jpanel);c.add(cpanel);
156         c.add(kpanel);c.add(annualincome);
157         c.add(npanel);c.add(l11);
158         c.add(opanel);c.add(l12);
159         c.add(bsubmit);c.add(back);
160         setSize(500, 425);
161         setLocation(600,200);
162         setResizable(false);
163         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
164         setVisible(true);
165     }
166     public void actionPerformed(ActionEvent e) {
167         String name,name1,pass,add,ct,gn,dob,ins;
168         OperatorRegData reg;
169         if(e.getSource() == bsubmit) {
170             String n = tid.getText().trim();
171             String n1 = tod.getText().trim();
172             String ps = tpass.getText().trim();
173             String ad = tadd.getText().trim();
174             String namepattern = "^[A-Za-z]";
175             Scanner   scan  =  new Scanner( n ) ;
176             String   matched  =  scan.findInLine( namepattern ) ;
177             if ( matched == null ) {
178                 JOptionPane.showMessageDialog(this, "INVALID    FIRST NAME\nIt
    should contain only alphabet.", "Error", JOptionPane.ERROR_MESSAGE);
179                 tid.setText("");
180             }
181             else
182                 flagName = true;
183             String namepattern1 = "^[A-Za-z]";
184             Scanner   scan1  =  new Scanner( n1 ) ;
185             String   matched1  =  scan1.findInLine( namepattern1 ) ;
186             if ( matched1 == null ) {
```

```
187                 JOptionPane.showMessageDialog(this, "INVALID LAST NAME\nIt should
    contain only alphabet.", "Error", JOptionPane.ERROR_MESSAGE);
188                 tod.setText("");
189             }
190             else
191                 flagName1 = true;
192             String passwordpattern = "^[0-9]{4}" ;
193             Scanner   scan2  =  new Scanner( ps ) ;
194             String   matched2  =  scan2.findInLine( passwordpattern ) ;
195             if ( matched2 == null ) {
196                 JOptionPane.showMessageDialog(this, "INVALID PASSWORD\nIt should
    contain only digits with minimum password length of 4 digit.", "Error",
    JOptionPane.ERROR_MESSAGE);
197                 tpass.setText("");
198              }
199             else
200                 flagPass = true;
201             String addpattern = "^[A-Za-z0-9]{1}";
202             Scanner   scan3  =  new Scanner( ad ) ;
203             String   matched3  =  scan3.findInLine( addpattern ) ;
204             if ( matched3 == null ) {
205                 JOptionPane.showMessageDialog(this, "INVALID ADDRESS\nIt should not
    blank.", "Error", JOptionPane.ERROR_MESSAGE);
206                 tadd.setText("");
207             }
208             else
209                 flagAdd = true;
210             name =tid.getText().trim();
211             name1=tod.getText().trim();
212             pass=tpass.getText().trim();
213             add=tadd.getText().trim();
214             ct=(String)city.getSelectedItem();
215             gn="";
216             if(rmale.isSelected())
217                 gn="Male";
218             else if(rfemale.isSelected())
219                 gn="Female";
220             String d=(String)day.getSelectedItem();
221             String m=(String)month.getSelectedItem();
222             String y=(String)year.getSelectedItem();
223             dob=d + "-" + m + "-" + y;
224             if(flagName == true) {
225                 if(flagName1 == true) {
226                     if(flagPass == true) {
227                         if(flagAdd == true) {
228                             reg = new OperatorRegData();
229                             reg.setName(name);
230                             reg.setName1(name1);
231                             reg.setPassword(pass);
232                             reg.setAddress(add);
233                             reg.setCity(ct);
234                             reg.setGender(gn);
235                             reg.setDob(dob);
236                             reg.setDate(l11.getText());
237                             reg.setTime(l12.getText());
238                             alist.add(reg);
239                             int con=JOptionPane.showConfirmDialog(this, "Are You
    Sure to Register?");
240                             if(con==JOptionPane.YES_OPTION) {
241                                 new OperatorAddInfo(reg);
242                                 JOptionPane.showMessageDialog(this, "Successfully
    registered");
```

```
243                                 tid.setText("");
244                                 tod.setText("");
245                                 tadd.setText("");
246                                 tpass.setText("");
247                             }
248                         }
249                     }
250                 }
251             }
252             flagName = false;
253             flagName1 = false;
254             flagPass = false;
255             flagAdd = false;
256         }
257         if(e.getSource()==back) {
258             int con=JOptionPane.showConfirmDialog(this, "Are You Sure to Quit?");
259             if(con==JOptionPane.YES_OPTION) {
260                 new Admin();
261                 setVisible(false);
262             }
263         }
264     }
265 }
```

```java
package test;
import java.io.Serializable;
public class OperatorRegData implements Serializable {
    private String name;
    private String name1;
    private String password;
    private String address;
    private String city;
    private String gender;
    private String salary;
    private String dob;
    private String date;
    private String time;
    public String getName1() {
        return name1;
    }
    public void setName1(String name1) {
        this.name1 = name1;
    }
    public String getSalary() {
        return salary;
    }
    public void setSalary(String salary) {
        this.salary = salary;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }
    public String getCity() {
        return city;
    }
    public void setCity(String city) {
        this.city = city;
    }
    public String getGender() {
        return gender;
    }
    public void setGender(String gender) {
        this.gender = gender;
    }
    public String getDob() {
        return dob;
    }
    public void setDob(String dob) {
        this.dob = dob;
    }
        public String getDate() {
```

```
63          return date;
64      }
65      public void setDate(String date) {
66          this.date = date;
67      }
68      public String getTime() {
69          return time;
70      }
71      public void setTime(String time) {
72          this.time = time;
73      }
74  }
```

```java
package test;
import java.awt.event.*;
public class OperatorSearch extends JFrame implements ActionListener {
    private JLabel l1,l2,l3;
    private JTextField t1;
    private JButton submit,back;
    private boolean flag = false;
    public OperatorSearch() {
        super("Operator Search");
        Container c=getContentPane();
        c.setLayout(new GridLayout(3,2));
        Font f1=new Font("Times New Roman",Font.BOLD,20);
        l1=new JLabel("OPERATOR");
        l1.setFont(f1);
        l1.setForeground(Color.RED);
        JPanel fpanel=new JPanel();
        fpanel.add(l1);
        fpanel.setBackground(new Color(0,0,64));
        l2=new JLabel("SEARCH");
        l2.setFont(f1);
        l2.setForeground(Color.RED);
        JPanel apanel=new JPanel();
        apanel.add(l2);
        apanel.setBackground(new Color(0,0,64));
        l3=new JLabel("Give OPERATOR First name");
        l3.setFont(f1);
        l3.setForeground(Color.RED);
        JPanel bpanel=new JPanel();
        bpanel.add(l3);
        bpanel.setBackground(new Color(0,0,64));
        t1=new JTextField();
        submit=new JButton("SUBMIT");
        submit.addActionListener(this);
        JPanel cpanel=new JPanel();
        cpanel.add(submit);
        cpanel.setBackground(new Color(0,0,64));
        back=new JButton("BACK");
        back.addActionListener(this);
        JPanel dpanel=new JPanel();
        dpanel.add(back);
        dpanel.setBackground(new Color(0,0,64));
        c.add(fpanel);c.add(apanel);
        c.add(bpanel);c.add(t1);
        c.add(cpanel);c.add(dpanel);
        setSize(550,300);
        setLocation(200,200);
        setResizable(false);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }
    public void actionPerformed(ActionEvent e) {
        if(e.getSource() == submit) {
            String n = t1.getText().trim();
            String namepattern = "^[A-Za-z]";
            Scanner   scan   =   new Scanner( n ) ;
            String   matched   =   scan.findInLine( namepattern ) ;
            if ( matched == null ) {
                JOptionPane.showMessageDialog(this, "INVALID    OPERATOR NAME\nIt should starts with alphabet.", "Error", JOptionPane.ERROR_MESSAGE);
                t1.setText("");
            }
            else
```

```
66                  flag = true;
67              if(flag == true) {
68                  String sname = t1.getText().trim();
69                  new OperatorSearchDisplay(sname);
70                  t1.setText("");
71              }
72              flag = false;
73          }
74          if(e.getSource()==back) {
75              int con=JOptionPane.showConfirmDialog(this, "Are you sure to cancel?");
76              if(con==JOptionPane.YES_OPTION) {
77                  new Admin();
78                  setVisible(false);
79              }
80          }
81      }
82  }
```

```java
package test;
import java.awt.*;
public class OperatorSearchDisplay extends JFrame {
    String data[][];
    String heading[]={"First Name","Last name","Password","Address","City","Gender","Date Of Birth","Reg Date","Reg Time"};
    ArrayList<OperatorRegData> list;
    private boolean flagIn;
    public OperatorSearchDisplay(String name) {
        super("Search for user account number" + name);
        try {
            FileInputStream fin=new FileInputStream("Reg.dat");
            ObjectInputStream oin=new ObjectInputStream(fin);
            list=(ArrayList<OperatorRegData>)oin.readObject();
            int row=0;
            for(OperatorRegData re : list) {
                if(re.getName().equals(name)) {
                    row++;
                }
            }
            data = new String[row][heading.length+1];
            int r=0;
            for(OperatorRegData re : list) {
                if(re.getName().equals(name)) {
                    data[r][0]=re.getName();
                    data[r][1]=re.getName1();
                    data[r][2]=re.getPassword();
                    data[r][3]=re.getAddress();
                    data[r][4]=re.getCity();
                    data[r][5]=re.getGender();
                    data[r][6]=re.getDob();
                    data[r][7]=re.getDate();
                    data[r][8]=re.getTime();
                    flagIn = true;
                }
            }
            if(flagIn == true) {
                Container con=getContentPane();
                con.setLayout(new BorderLayout());
                JTable datatable=new JTable(data, heading);
                JScrollPane jsp=new JScrollPane(datatable);
                con.add(new JLabel("Registration Details for " + name),BorderLayout.NORTH);
                con.add(jsp,BorderLayout.CENTER);
                setSize(650, 300);
                setLocation(200, 200);
                setVisible(true);
            }
            else
                JOptionPane.showMessageDialog(this, "Match not Found");
            flagIn = false;

        }catch(Exception e) {
            JOptionPane.showMessageDialog(this, "No file found in data base", "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
}
```

```java
package test;
import java.awt.event.*;
public class OperatorUserSearch extends JFrame implements ActionListener {
    private JLabel l1,l2,l3;
    private JTextField t1;
    private JButton submit,back;
    private boolean flag = false;
    public OperatorUserSearch() {
        super("User Search");
        Container c=getContentPane();
        c.setLayout(new GridLayout(3,2));
        Font f1=new Font("Times New Roman",Font.BOLD,20);
        l1=new JLabel("CUSTOMER");
        l1.setFont(f1);
        l1.setForeground(Color.RED);
        JPanel fpanel=new JPanel();
        fpanel.add(l1);
        fpanel.setBackground(new Color(0,0,64));
        l2=new JLabel("SEARCH");
        l2.setFont(f1);
        l2.setForeground(Color.RED);
        JPanel apanel=new JPanel();
        apanel.add(l2);
        apanel.setBackground(new Color(0,0,64));
        l3=new JLabel("Give User A/C no");
        l3.setFont(f1);
        l3.setForeground(Color.RED);
        JPanel bpanel=new JPanel();
        bpanel.add(l3);
        bpanel.setBackground(new Color(0,0,64));
        t1=new JTextField();
        submit=new JButton("SUBMIT");
        submit.addActionListener(this);
        JPanel cpanel=new JPanel();
        cpanel.add(submit);
        cpanel.setBackground(new Color(0,0,64));
        back=new JButton("BACK");
        back.addActionListener(this);
        JPanel dpanel=new JPanel();
        dpanel.add(back);
        dpanel.setBackground(new Color(0,0,64));
        c.add(fpanel);c.add(apanel);
        c.add(bpanel);c.add(t1);
        c.add(cpanel);c.add(dpanel);
        setSize(550,300);
        setLocation(200,200);
        setResizable(false);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }
    public void actionPerformed(ActionEvent e) {
        if(e.getSource() == submit) {
            String n = t1.getText().trim();
            String namepattern = "^[0-9]{4}";
            Scanner   scan   =   new Scanner( n ) ;
            String   matched   =   scan.findInLine( namepattern ) ;
            if ( matched == null ) {
                JOptionPane.showMessageDialog(this, "INVALID    OPERATOR NAME\nIt should contain only digit.", "Error", JOptionPane.ERROR_MESSAGE);
                t1.setText("");
            }
            else
```

```
65                  flag  = true;
66              if(flag == true) {
67                  String sname = t1.getText().trim();
68                  new UserSearchDisplay(sname);
69                  t1.setText("");
70              }
71              flag = false;
72          }
73          if(e.getSource()==back) {
74              int con=JOptionPane.showConfirmDialog(this, "Are you sure to cancel?");
75              if(con==JOptionPane.YES_OPTION) {
76                  new Userwindow("");
77                  setVisible(false);
78              }
79          }
80      }
81  }
```

```
 1 package test;
 2 import javax.swing.*;
 8 public class PassBook extends JFrame implements ActionListener {
 9     private JLabel l1;
10     private JTextField tid;
11     private JButton bdisplay,cancel;
12     private boolean flagAccNum = false;
13     private ArrayList<Actotal> list4;
14     private boolean flagAccMatch = false;
15     public PassBook() {
16         super("Indivisual Passbook Display");
17         Container c=getContentPane();
18         c.setLayout(new GridLayout(2,2));
19         tid=new JTextField();
20         bdisplay=new JButton("SUBMIT");
21         bdisplay.addActionListener(this);
22         JPanel apanel=new JPanel();
23         apanel.add(bdisplay);
24         apanel.setBackground(new Color(0,0,64));
25         cancel=new JButton("CANCEL");
26         cancel.addActionListener(this);
27         JPanel bpanel=new JPanel();
28         bpanel.add(cancel);
29         bpanel.setBackground(new Color(0,0,64));
30         Font f1=new Font("comic sans ms",Font.BOLD,14);
31         l1=new JLabel("Enter A/C no");
32         l1.setFont(f1);
33         l1.setForeground(Color.RED);
34         JPanel cpanel=new JPanel();
35         cpanel.add(l1);
36         cpanel.setBackground(new Color(0,0,64));
37         c.add(cpanel);c.add(tid);
38         c.add(apanel);c.add(bpanel);
39         setSize(400,175);
40         setLocation(600,200);
41         setResizable(false);
42         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
43         setVisible(true);
44     }
45     public void actionPerformed(ActionEvent e) {
46         if(e.getSource()==bdisplay) {
47             String vAccNum = tid.getText();
48             String accNumpattern = "^[0-9]{4}" ;
49             Scanner   scan1  =  new Scanner( vAccNum ) ;
50             String   matched1  =  scan1.findInLine( accNumpattern ) ;
51             if ( matched1 == null ) {
52                 JOptionPane.showMessageDialog(this, "INVALID ACCOUNT NUMBER\nIt
   should contain only digits with minimum length of 4 digit.", "Error",
   JOptionPane.ERROR_MESSAGE);
53                 tid.setText("");
54             }
55             else
56                 flagAccNum     = true;
57             try {
58                 FileInputStream fin=new FileInputStream("actotal.dat");
59                 ObjectInputStream oin=new ObjectInputStream(fin);
60                 list4=(ArrayList<Actotal>)oin.readObject();
61             }catch(Exception ex) {
62                 JOptionPane.showMessageDialog(this, "No file found in data base",
   "Error", JOptionPane.ERROR_MESSAGE);
63             }
64             if(flagAccNum == true) {
```

```
65                  for(Actotal re : list4) {
66                      if(String.valueOf(re.getAccount()).equals(tid.getText())) {
67                          new Individualpassbook(tid.getText());
68                          tid.setText("");
69                          flagAccMatch = true;
70                      }
71                  }
72                  if(flagAccMatch == false) {
73                      JOptionPane.showMessageDialog(this, "Account number does not
  exist.", "Error", JOptionPane.ERROR_MESSAGE);
74                      tid.setText("");
75                  }
76                  flagAccMatch = false;
77              }
78              flagAccNum = false;
79          }
80          if(e.getSource()==cancel) {
81              int con=JOptionPane.showConfirmDialog(this, "Are you sure to cancel?");
82              if(con==JOptionPane.YES_OPTION) {
83                  new Userwindow("");
84                  setVisible(false);
85              }
86          }
87      }
88  }
```

```java
package test;
import javax.swing.*;
public class Passdisp extends JFrame {
    public Passdisp(String accNumber) {
        super("Passbook");
        String heading[]={"Accno","Withdraw","Deposit","Total","Date","Time"};
        String data[][];//=new String[1000][6];
        ArrayList<Actotal> list=new ArrayList<Actotal>();
        try {
            FileInputStream fin=new FileInputStream("actotal.dat");
            ObjectInputStream oin=new ObjectInputStream(fin);
            list=(ArrayList<Actotal>)oin.readObject();
        }catch(Exception e) {
            JOptionPane.showMessageDialog(this, "No file found in data base",
  "Error", JOptionPane.ERROR_MESSAGE);
        }
        int row=0;
        for(Actotal re : list) {
            if(String.valueOf(re.getAccount()).equals(accNumber))
                row++;
        }
        data = new String[row][heading.length+1];
        int r=0,i=0;
        for(Actotal re : list) {
            if(String.valueOf(re.getAccount()).equals(accNumber)) {
                data[r][0]=String.valueOf(re.getAccount());
                data[r][1]=String.valueOf(re.getWithdraw1());
                data[r][2]=String.valueOf(re.getDepo());
                data[r][3]=String.valueOf(re.getTotal());
                data[r][4]=re.getDate();
                data[r][5]=re.getTime();
                r++;
            }
        }
        Container con=getContentPane();
        con.setLayout(new BorderLayout());
        JTable datatable=new JTable(data,heading);
        JScrollPane jsp=new JScrollPane(datatable);
        con.add(new JLabel("All Registration Details"),BorderLayout.NORTH);
        con.add(jsp,BorderLayout.CENTER);
        setSize(650, 300);
        setLocation(200, 200);
        setVisible(true);
    }
}
```

```java
package test;
import java.io.*;
public class SavingsUpdate  {
    public SavingsUpdate(int acc,int total) {
        ArrayList<Depo> list4;
        String ac,amt;
        ac = String.valueOf(acc);
        amt = String.valueOf(total);
        try {
            FileInputStream fin=new FileInputStream("fixed.dat");
            ObjectInputStream oin=new ObjectInputStream(fin);
            list4=(ArrayList<Depo>)oin.readObject();
        }catch(Exception e) {
            list4=new ArrayList<Depo>();
        }
        for(int i = 0 ;i<list4.size();i++) {
            if(list4.get(i).getSavingsacc().equals(ac)) {
                list4.get(i).setSavingsaccbal(amt);
            }
        }
        try {
            FileOutputStream fout=new FileOutputStream("fixed.dat");
            ObjectOutputStream oout=new ObjectOutputStream(fout);
            oout.writeObject(list4);
        }catch(Exception e){}
    }
}

```

```java
package test;
import java.io.*;
public class TotalUpdate {
    public TotalUpdate(int acc,int total) {
        ArrayList<Create> list4;
        String ac,amt;
        ac = String.valueOf(acc);
        amt = String.valueOf(total);
        try {
            FileInputStream fin=new FileInputStream("Regis.dat");
            ObjectInputStream oin=new ObjectInputStream(fin);
            list4=(ArrayList<Create>)oin.readObject();
        }catch(Exception e) {
            list4=new ArrayList<Create>();
        }
        for(int i = 0 ;i<list4.size();i++) {
            if(list4.get(i).getAc().equals(ac)) {
                 list4.get(i).setInitialamnt(amt);
            }
        }
        try {
            FileOutputStream fout=new FileOutputStream("Regis.dat");
            ObjectOutputStream oout=new ObjectOutputStream(fout);
            oout.writeObject(list4);
        }catch(Exception e){}
    }
}
```

```java
package test;
import java.awt.*;
public class Transfer extends JFrame implements ActionListener {
    private JLabel l1,l2,l3;
    private JTextField tid1,tid2,tid3;
    private JButton bsubmit,bBack;
    private int r=0,sourceamnt,destamnt,tran,t,acc,d=0,w=0;
    ArrayList<Create> list1;
    private boolean flagSource = false;
    private boolean flagDestination = false;
    private boolean flagAmount = false;
    private boolean flagSourceMatch = false;
    private boolean flagDestinationMatch = false;
    private boolean flagSourceAmountMatch = false;
    private boolean flagSourceInsufficientBalance = false;
    public Transfer() {
        super("Money Transfer");
        tid1=new JTextField();
        tid2=new JTextField();
        tid3=new JTextField();
        Container c=getContentPane();
        c.setLayout(new GridLayout(4,2));
        bsubmit=new JButton("Submit");
        bsubmit.addActionListener(this);
        JPanel epanel=new JPanel();
        epanel.add(bsubmit);
        epanel.setBackground(new Color(0,0,64));
        bBack = new JButton("Back");
        bBack.addActionListener(this);
        JPanel fpanel=new JPanel();
        fpanel.add(bBack);
        fpanel.setBackground(new Color(0,0,64));
        Font f1=new Font("comic sans ms",Font.BOLD,14);
        l1=new JLabel("Enter Source A/C no");
        l1.setFont(f1);
        l1.setForeground(Color.RED);
        JPanel gpanel=new JPanel();
        gpanel.add(l1);
        gpanel.setBackground(new Color(0,0,64));
        l2=new JLabel("Enter Destination A/C no");
        l2.setFont(f1);
        l2.setForeground(Color.RED);
        JPanel apanel=new JPanel();
        apanel.add(l2);
        apanel.setBackground(new Color(0,0,64));
        l3=new JLabel("Enter Amount to be Transferred");
        l3.setFont(f1);
        l3.setForeground(Color.RED);
        JPanel bpanel=new JPanel();
        bpanel.add(l3);
        bpanel.setBackground(new Color(0,0,64));
        c.add(gpanel);c.add(tid1);
        c.add(apanel);c.add(tid2);
        c.add(bpanel);c.add(tid3);
        c.add(epanel);c.add(fpanel);
        setSize(500, 425);
        setLocation(600,200);
        setResizable(false);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }
    public void actionPerformed(ActionEvent e) {
```

```
 67         String source,dest;
 68         source=tid1.getText();
 69         dest=tid2.getText();
 70         if(e.getSource()==bsubmit) {
 71             String vSource = tid1.getText();
 72             String vDestination = tid2.getText();
 73             String vAmount = tid3.getText();
 74             String sourcepattern = "^[0-9]{4}" ;
 75             Scanner   scan  =  new Scanner( vSource) ;
 76             String   matched =  scan.findInLine( sourcepattern ) ;
 77             if ( matched == null ) {
 78                 JOptionPane.showMessageDialog(this, "INVALID SOURCE ACCOUNT
    NUMBER\nIt should contain only digits with minimum length of 4 digit.", "Error",
    JOptionPane.ERROR_MESSAGE);
 79                 tid1.setText("");
 80             }
 81             else
 82                 flagSource    = true;
 83             String destinationpattern = "^[0-9]{4}" ;
 84             Scanner   scan1  =  new Scanner( vDestination) ;
 85             String   matched1 =  scan1.findInLine( destinationpattern ) ;
 86             if ( matched1 == null ) {
 87                 JOptionPane.showMessageDialog(this, "INVALID DESTINATION ACCOUNT
    NUMBER\nIt should contain only digits with minimum length of 4 digit.", "Error",
    JOptionPane.ERROR_MESSAGE);
 88                 tid2.setText("");
 89             }
 90             else
 91                 flagDestination    = true;
 92             String amountwpattern = "^[0-9]{3}" ;
 93             Scanner   scan2  =  new Scanner( vAmount ) ;
 94             String   matched2  =  scan2.findInLine( amountwpattern ) ;
 95             if ( matched2 == null ) {
 96                 JOptionPane.showMessageDialog(this, "INVALID AMOUNT\nIt should
    contain only digits with minimum amount of INR 100.", "Error",
    JOptionPane.ERROR_MESSAGE);
 97                 tid3.setText("");
 98             }
 99             else
100                 flagAmount    = true;
101             if(flagSource == true) {
102                 if(flagDestination == true) {
103                     if(flagAmount == true) {
104                         try  {
105                             FileInputStream fin=new FileInputStream("Regis.dat");
106                             ObjectInputStream oin=new ObjectInputStream(fin);
107                             list1=(ArrayList<Create>)oin.readObject();
108                         }
109                         catch(Exception e1){}
110                         for(Create re : list1) {
111                             if(re.getAc().equals(source)) {
112                                 flagSourceMatch = true;
113                                 for(Create re2 : list1) {
114                                     if(re2.getAc().equals(dest)) {
115                                         flagDestinationMatch = true;
116                                         sourceamnt=Integer.parseInt(re.getInitialam
    nt());
117                                         destamnt=Integer.parseInt(re2.getInitialamn
    t());
118                                         if(sourceamnt>500) {
119                                             flagSourceAmountMatch = true;
120                                             tran=Integer.parseInt(tid3.getText());
```

```
121                                             t=sourceamnt-tran;
122                                             if((t>0)&&(t>=500)) {
123                                                 flagSourceInsufficientBalance =
    true;
124                                                 acc=Integer.parseInt(tid1.getText()
    );
125                                                 new TotalUpdate(acc,t);
126                                                 new Actotalcreate(acc,tran,d,t);
127                                                 t=destamnt+tran;
128                                                 acc=Integer.parseInt(tid2.getText()
    );
129                                                 new TotalUpdate(acc,t);
130                                                 new Actotalcreate(acc,w,tran,t);
131                                                 r++;
132                                             }
133                                         }
134                                     }
135                                 }
136                             }
137                         }
138                         if(r==1) {
139                             JOptionPane.showMessageDialog(this, "MONEY TRANSFERED
    SUCCESSFULLY");
140                             tid1.setText("");
141                             tid2.setText("");
142                             tid3.setText("");
143                             JOptionPane.showMessageDialog(this, "Showing both
    account holders passbook.");
144                             new Individualpassbook(source);
145                             new Individualpassbook(dest);
146                         }
147                         if(flagSourceMatch == false) {
148                             JOptionPane.showMessageDialog(this, "Source account
    does not match", "Source account error", JOptionPane.ERROR_MESSAGE);
149                             tid1.setText("");
150                         }
151                         if(flagSourceMatch == true && flagDestinationMatch == true
    && flagSourceAmountMatch == false) {
152                             JOptionPane.showMessageDialog(this, "Source account
    does not have minimum balance of INR 500", "Insufficient balance error",
    JOptionPane.ERROR_MESSAGE);
153                             tid1.setText("");
154                             tid2.setText("");
155                             tid3.setText("");
156                         }
157                         else if(flagSourceMatch == true && flagDestinationMatch ==
    true && flagSourceAmountMatch == true && flagSourceInsufficientBalance == false) {
158                             JOptionPane.showMessageDialog(this, "Insufficient
    balance of source account to transfer money.", "nsufficient balance error",
    JOptionPane.ERROR_MESSAGE);
159                             tid3.setText("");
160                         }
161                         if(flagSourceMatch == true && flagDestinationMatch ==
    false) {
162                             JOptionPane.showMessageDialog(this, "Destination
    account does not match", "Destination account error", JOptionPane.ERROR_MESSAGE);
163                             tid2.setText("");
164                         }
165                     }
166                 }
167             }
168             flagSource = false;
```

```
169             flagDestination = false;
170             flagAmount = false;
171             flagSourceMatch = false;
172             flagDestinationMatch = false;
173             flagSourceAmountMatch = false;
174             flagSourceInsufficientBalance = false;
175             r=0;
176         }
177         if(e.getSource() == bBack) {
178             int rply = JOptionPane.showConfirmDialog(this, "Are you sure to
    quit?");
179             if(rply == JOptionPane.YES_OPTION) {
180                 new Userwindow("");
181                 setVisible(false);
182             }
183         }
184     }
185 }
```

```java
package test;
import java.awt.event.*;
class ThirdWindow extends JFrame implements ActionListener {
    private JLabel l0,l1,l2,l3;
    private JTextField t1;
    private JPasswordField p1;
    private JButton login,back;
    private boolean flagID = false;
    private boolean flagPass = false;
    private ArrayList<OperatorRegData> alistTemp;
    private boolean flag = false;
    public ThirdWindow() {
        super("Log in as operator");
        Container c=getContentPane();
        c.setLayout(new GridLayout(4,2));
        Font f1=new Font("Times New Roman",Font.BOLD,28);
        l0=new JLabel("            OPERATOR");
        l0.setFont(f1);
        l0.setForeground(Color.RED);
        JPanel fpanel=new JPanel();
        fpanel.add(l0);
        fpanel.setBackground(new Color(0,0,64));
        Font f2=new Font("Times New Roman",Font.BOLD,20);
        l1=new JLabel("LOGIN               ");
        l1.setFont(f1);
        l1.setForeground(Color.RED);
        JPanel fxpanel=new JPanel();
        fxpanel.add(l1);
        fxpanel.setBackground(new Color(0,0,64));
        l2=new JLabel("USERNAME");
        l2.setFont(f2);
        l2.setForeground(Color.RED);
        l3=new JLabel("PASSWORD");
        l3.setFont(f2);
        l3.setForeground(Color.RED);
        t1=new JTextField();
        p1=new JPasswordField();
        JPanel xpanel=new JPanel();
        xpanel.add(l2);
        xpanel.setBackground(new Color(0,0,64));
        JPanel xppanel=new JPanel();
        xppanel.add(t1);
        xppanel.setBackground(new Color(0,0,64));
        JPanel lpanel=new JPanel();
        lpanel.add(l3);
        lpanel.setBackground(new Color(0,0,64));
        JPanel xlpanel=new JPanel();
        xlpanel.add(p1);
        xlpanel.setBackground(new Color(0,0,64));
        login=new JButton("LOGIN");
        JPanel gpanel=new JPanel();
        gpanel.add(login);
        gpanel.setBackground(new Color(0,0,64));
        login.addActionListener(this);
        back=new JButton("BACK");
        JPanel apanel=new JPanel();
        apanel.add(back);
        apanel.setBackground(new Color(0,0,64));
        back.addActionListener(this);
        c.add(fpanel);c.add(fxpanel);
        c.add(xpanel);c.add(t1);
        c.add(lpanel);c.add(p1);
```

```
 67         c.add(gpanel);c.add(apanel);
 68         setSize(500,425);
 69         setLocation(200,200);
 70         setResizable(false);
 71         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
 72         setVisible(true);
 73     }
 74     public void actionPerformed(ActionEvent e)
 75     {
 76         if(e.getSource()==login) {
 77             String n = t1.getText().trim();
 78             String ps = p1.getText().trim();
 79             String namepattern = "^[A-Za-z]";
 80             Scanner   scan  =  new Scanner( n ) ;
 81             String   matched  =  scan.findInLine( namepattern ) ;
 82             if ( matched == null ) {
 83                 JOptionPane.showMessageDialog(this, "INVALID USER NAME\nIt should
    contain only alphabet.", "Error", JOptionPane.ERROR_MESSAGE);
 84                 t1.setText("");
 85             }
 86             else
 87                 flagID = true;
 88             String passwordpattern = "^[0-9]{4}" ;
 89             Scanner   scan2  =  new Scanner( ps ) ;
 90             String   matched2  =  scan2.findInLine( passwordpattern ) ;
 91             if ( matched2 == null ) {
 92                 JOptionPane.showMessageDialog(this, "INVALID PASSWORD\nIt should
    contain only digits with minimum password length of 4 digit.", "Error",
    JOptionPane.ERROR_MESSAGE);
 93                 p1.setText("");
 94             }
 95             else
 96                 flagPass = true;
 97             if(flagID == true ) {
 98                 if(flagPass == true) {
 99                     String enteredID = t1.getText().trim();
100                     String enteredPassword = p1.getText().trim();
101                     try {
102                         FileInputStream fin = new FileInputStream("Reg.dat");
103                         ObjectInputStream oin = new ObjectInputStream(fin);
104                         alistTemp = (ArrayList<OperatorRegData>)oin.readObject();
105                     }catch(Exception ex) {
106                         JOptionPane.showMessageDialog(this, "No file found in
    database.", "Error", JOptionPane.ERROR_MESSAGE);
107                     }
108                     for(OperatorRegData element : alistTemp) {
109                         if(element.getName().equals(enteredID)) {
110                             if(element.getPassword().equals(enteredPassword))
111                                 flag  = true;
112                         }
113                     }
114                     if(flag == true) {
115                         JOptionPane.showMessageDialog(this, "Successfully logged
    in");
116                         new Userwindow(t1.getText());
117                         setVisible(false);
118                     }
119                     else {
120                         JOptionPane.showMessageDialog(this, "Incorrect UserID or
    Password\nTry again.", "Error", JOptionPane.ERROR_MESSAGE);
121                         t1.setText("");
122                         p1.setText("");
```

```
123                     }
124                 }
125             }
126             flagID = false;
127             flagPass = false;
128         }
129         if(e.getSource()==back) {
130             int con=JOptionPane.showConfirmDialog(this, "Are You Sure to go
    back?");
131             if(con==JOptionPane.YES_OPTION) {
132                 new FirstWindow("");
133                 setVisible(false);
134             }
135         }
136     }
137 }
138 public class UserMain1 {
139     public static void main(String[] args) {}
140 }
```

```java
package test;
import java.awt.event.*;
public class UserRemove extends JFrame implements ActionListener {
    private JLabel l1,l2,l3;
    private JTextField t1;
    private JButton submit,back;
    private boolean flagAccNum = false;
    public UserRemove() {
        super("Remove an user");
        Container c=getContentPane();
        c.setLayout(new GridLayout(3,2));
        Font f1=new Font("Times New Roman",Font.BOLD,20);
        l1=new JLabel("USER");
        l1.setFont(f1);
        l1.setForeground(Color.RED);
        JPanel fpanel=new JPanel();
        fpanel.add(l1);
        fpanel.setBackground(new Color(0,0,64));
        l2=new JLabel("Delete");
        l2.setFont(f1);
        l2.setForeground(Color.RED);
        JPanel apanel=new JPanel();
        apanel.add(l2);
        apanel.setBackground(new Color(0,0,64));
        l3=new JLabel("Give USER A/C no");
        l3.setFont(f1);
        l3.setForeground(Color.RED);
        JPanel bpanel=new JPanel();
        bpanel.add(l3);
        bpanel.setBackground(new Color(0,0,64));
        t1=new JTextField();
        submit=new JButton("Delete");
        submit.addActionListener(this);
        JPanel cpanel=new JPanel();
        cpanel.add(submit);
        cpanel.setBackground(new Color(0,0,64));
        back=new JButton("BACK");
        back.addActionListener(this);
        JPanel dpanel=new JPanel();
        dpanel.add(back);
        dpanel.setBackground(new Color(0,0,64));
        c.add(fpanel);c.add(apanel);
        c.add(bpanel);c.add(t1);
        c.add(cpanel);c.add(dpanel);
        setSize(550,300);
        setLocation(200,200);
        setResizable(false);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }
    public void actionPerformed(ActionEvent e) {
        if(e.getSource() == submit) {
            String vaccNum = t1.getText();
            String accNumpattern = "^[0-9]{4}" ;
            Scanner   scan  =  new Scanner( vaccNum ) ;
            String   matched =  scan.findInLine( accNumpattern ) ;
            if ( matched == null ) {
                JOptionPane.showMessageDialog(this, "INVALID ACCOUNT NUMBER\nIt
  should contain only digits with minimum length of 4 digit.", "Error",
  JOptionPane.ERROR_MESSAGE);
                t1.setText("");
            }
```

```
65              else
66                  flagAccNum    = true;
67              if(flagAccNum == true) {
68                  String sname = t1.getText().trim();
69                  new UserRemoveDisplay(sname);
70                  t1.setText("");
71              }
72              flagAccNum = false;
73          }
74          if(e.getSource()==back) {
75              int con=JOptionPane.showConfirmDialog(this, "Are You Sure to cancel?");
76              if(con==JOptionPane.YES_OPTION) {
77                  new Userwindow("");
78                  setVisible(false);
79              }
80          }
81      }
82  }
```

```java
package test;
import java.io.*;
public class UserRemoveDisplay extends JFrame {
    ArrayList<Actotal>list1;
    ArrayList<Create> list2;
    int p;
    public UserRemoveDisplay(String name) {
        p=Integer.parseInt(name);
        try {
            FileInputStream fin=new FileInputStream("Regis.dat");
            ObjectInputStream oin=new ObjectInputStream(fin);
            list2=(ArrayList<Create>)oin.readObject();
        }catch(Exception e) {
            list2 = new ArrayList<Create>();
        }
        int r=0,c=0;
        for(int i=0;i<list2.size();i++) {
            if(list2.get(i).getAc().equals(name)) {
                r++;
                String r1=String.valueOf(r);
                list2.remove(i);
            }
        }
        try  {
            FileOutputStream fout=new FileOutputStream("Regis.dat");
            ObjectOutputStream oout=new ObjectOutputStream(fout);
            oout.writeObject(list2);
        }catch(Exception e){}
        try {
            FileInputStream fin=new FileInputStream("actotal.dat");
            ObjectInputStream oin=new ObjectInputStream(fin);
            list1=(ArrayList<Actotal>)oin.readObject();
        }catch(Exception e) {
            list1 = new ArrayList<Actotal>();
        }
        for(int i=0;i<list1.size();) {
            if(list1.get(i).getAccount()==p) {
                c++;
                String c1=String.valueOf(c);
                list1.remove(i);
            }
            else
                i++;
        }
        try  {
            FileOutputStream fout=new FileOutputStream("actotal.dat");
            ObjectOutputStream oout=new ObjectOutputStream(fout);
            oout.writeObject(list1);
        }catch(Exception e){}
        if(r==1)
            JOptionPane.showMessageDialog(this, "USER RECORD REMOVED");
        else
            JOptionPane.showMessageDialog(this, "No Data Found");
    }
}
```

```java
package test;
import java.awt.event.*;
public class UserSearch extends JFrame implements ActionListener {
    private JLabel l1,l2,l3;
    private JTextField t1;
    private JButton submit,back;
    private boolean flag = false;
    public UserSearch() {
        super("User Search");
        Container c=getContentPane();
        c.setLayout(new GridLayout(3,2));
        Font f1=new Font("Times New Roman",Font.BOLD,20);
        l1=new JLabel("CUSTOMER");
        l1.setFont(f1);
        l1.setForeground(Color.RED);
        JPanel fpanel=new JPanel();
        fpanel.add(l1);
        fpanel.setBackground(new Color(0,0,64));
        l2=new JLabel("SEARCH");
        l2.setFont(f1);
        l2.setForeground(Color.RED);
        JPanel apanel=new JPanel();
        apanel.add(l2);
        apanel.setBackground(new Color(0,0,64));
        l3=new JLabel("Give User A/C no");
        l3.setFont(f1);
        l3.setForeground(Color.RED);
        JPanel bpanel=new JPanel();
        bpanel.add(l3);
        bpanel.setBackground(new Color(0,0,64));
        t1=new JTextField();
        submit=new JButton("SUBMIT");
        submit.addActionListener(this);
        JPanel cpanel=new JPanel();
        cpanel.add(submit);
        cpanel.setBackground(new Color(0,0,64));
        back=new JButton("BACK");
        back.addActionListener(this);
        JPanel dpanel=new JPanel();
        dpanel.add(back);
        dpanel.setBackground(new Color(0,0,64));
        c.add(fpanel);c.add(apanel);
        c.add(bpanel);c.add(t1);
        c.add(cpanel);c.add(dpanel);
        setSize(550,300);
        setLocation(200,200);
        setResizable(false);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }
    public void actionPerformed(ActionEvent e) {
        if(e.getSource() == submit) {
            String n = t1.getText().trim();
            String namepattern = "^[0-9]{4}";
            Scanner   scan  =  new Scanner( n ) ;
            String   matched  =  scan.findInLine( namepattern ) ;
            if ( matched == null ) {
                JOptionPane.showMessageDialog(this, "INVALID USER ACCOUNT NUMBER\nIt
  should contain only digit with minimum of 4 digit.", "Error",
  JOptionPane.ERROR_MESSAGE);
                t1.setText("");
            }
```

```
64              else
65                  flag  = true;
66              if(flag == true) {
67                  String sname = t1.getText().trim();
68                  new UserSearchDisplay(sname);
69                  t1.setText("");
70              }
71              flag = false;
72          }
73          if(e.getSource()==back) {
74              int con=JOptionPane.showConfirmDialog(this, "Are you sure to cancel?");
75              if(con==JOptionPane.YES_OPTION) {
76                  new Admin();
77                  setVisible(false);
78              }
79          }
80      }
81  }
```

```java
package test;
import java.awt.*;
public class UserSearchDisplay extends JFrame {
    String data[][];
    String heading[]={"First Name","Last name","Address","Nationality","Account Type","Date Of Birth","Identity","Profession","Reg Date","Reg Time","A/c no","Balance"};
    ArrayList<Create> list;
    public UserSearchDisplay(String name) {
        super("Search for user account number " + name);
        try {
            FileInputStream fin=new FileInputStream("Regis.dat");
            ObjectInputStream oin=new ObjectInputStream(fin);
            list=(ArrayList<Create>)oin.readObject();
            int row=0;
            for(Create re : list) {
                if(re.getAc().equals(name))
                    row++;
            }
            data = new String[row][heading.length+1];
            int r=0;
            for(Create re : list) {
                if(re.getAc().equals(name)) {
                      data[r][0]=re.getName();
                      data[r][1]=re.getName1();
                      data[r][2]=re.getAddress();
                      data[r][3]=re.getNationality();
                      data[r][4]=re.getAcctype();
                      data[r][5]=re.getDob();
                      data[r][6]=re.getIdentity();
                      data[r][7]=re.getProfession();
                      data[r][8]=re.getDate();
                      data[r][9]=re.getTime();
                      data[r][10]=re.getAc();
                      data[r][11]=re.getInitialamnt();
                      r++;
                }
            }
            if (r>0) {
                Container con=getContentPane();
                con.setLayout(new BorderLayout());
                JTable datatable=new JTable(data, heading);
                JScrollPane jsp=new JScrollPane(datatable);
                con.add(new JLabel("Registration Details"),BorderLayout.NORTH);
                con.add(jsp,BorderLayout.CENTER);
                setSize(650, 300);
                setLocation(200, 200);
                setVisible(true);
            }
            else
                JOptionPane.showMessageDialog(this, "Match not Found");
        }catch(Exception e) {
            JOptionPane.showMessageDialog(this, "No file found in data base", "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
}
```

```java
package test;
import java.awt.*;
class Userwindow extends JFrame implements ActionListener {
    private JLabel l1,l2;
    private JButton
 bcreate,bremove,bdetails,bwithdraw,btransfer,bpass,bexit,bChangePassword;
    Userwindow(String currentOperator) {
        super("Welcome " + currentOperator);
        Container c=getContentPane();
        c.setLayout(new GridLayout(5,2));
        Font f1=new Font("Times New Roman",Font.BOLD,25);
        l1=new JLabel("                OPERATOR");
        l1.setFont(f1);
        l1.setForeground(Color.WHITE);
        JPanel fpanel=new JPanel();
        fpanel.add(l1);
        fpanel.setBackground(new Color(0,0,64));
        l2=new JLabel("FUNCTIONS              ");
        l2.setFont(f1);
        l2.setForeground(Color.WHITE);
        JPanel ppanel=new JPanel();
        ppanel.add(l2);
        ppanel.setBackground(new Color(0,0,64));
        bcreate=new JButton("CREATE NEW ACCOUNT");
        bcreate.addActionListener(this);
        JPanel apanel=new JPanel();
        apanel.add(bcreate);
        apanel.setBackground(new Color(0,0,64));
        bremove=new JButton("REMOVE ACCOUNT");
        bremove.addActionListener(this);
        JPanel bpanel=new JPanel();
        bpanel.add(bremove);
        bpanel.setBackground(new Color(0,0,64));
        bdetails=new JButton("VIEW DETAILS OF AN ACCOUNT");
        bdetails.addActionListener(this);
        JPanel cpanel=new JPanel();
        cpanel.add(bdetails);
        cpanel.setBackground(new Color(0,0,64));
        bwithdraw=new JButton("WITHDRAWAL/DEPOSIT");
        bwithdraw.addActionListener(this);
        JPanel dpanel=new JPanel();
        dpanel.add(bwithdraw);
        dpanel.setBackground(new Color(0,0,64));
        btransfer=new JButton("TRANSFER MONEY");
        btransfer.addActionListener(this);
        JPanel epanel=new JPanel();
        epanel.add(btransfer);
        epanel.setBackground(new Color(0,0,64));
        bpass=new JButton("VIEW ACCOUNT PASS BOOK");
        bpass.addActionListener(this);
        JPanel qpanel=new JPanel();
        qpanel.add(bpass);
        qpanel.setBackground(new Color(0,0,64));
        bexit=new JButton("LOG OUT");
        bexit.addActionListener(this);
        JPanel gpanel=new JPanel();
        gpanel.add(bexit);
        gpanel.setBackground(new Color(0,0,64));
        bChangePassword=new JButton("CHANGE PASSWORD");
        bChangePassword.addActionListener(this);
        JPanel aapanel=new JPanel();
        aapanel.add(bChangePassword);
```

```
64          aapanel.setBackground(new Color(0,0,64));
65          c.add(fpanel);c.add(ppanel);
66          c.add(apanel);c.add(dpanel);
67          c.add(bpanel);c.add(epanel);
68          c.add(qpanel);c.add(cpanel);
69          c.add(aapanel);c.add(gpanel);
70          setSize(500,400);
71          setLocation(200,200);
72          setResizable(false);
73          setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
74          setVisible(true);
75      }
76      public void actionPerformed(ActionEvent e) {
77          if(e.getSource()==bcreate) {
78              new CreateFrame();
79              setVisible(false);
80          }
81          if(e.getSource()==bwithdraw) {
82              new Withdraw();
83              setVisible(false);
84          }
85          if(e.getSource()==bremove) {
86              new UserRemove();
87              setVisible(false);
88          }
89          if(e.getSource()==btransfer) {
90              new Transfer();
91              setVisible(false);
92          }
93          if(e.getSource()==bpass) {
94              new PassBook();
95              setVisible(false);
96          }
97          if(e.getSource()==bdetails) {
98              new OperatorUserSearch();
99              setVisible(false);
100         }
101         if(e.getSource()==bexit) {
102              int con=JOptionPane.showConfirmDialog(this, "Are you sure to
    cancel?");
103                 if(con==JOptionPane.YES_OPTION) {
104                     new ThirdWindow();
105                     setVisible(false);
106                 }
107         }
108         if(e.getSource()==bChangePassword) {
109             new ChangePassword();
110             setVisible(false);
111         }
112     }
113 }
```

```java
package test;
import java.awt.event.*;
public class Withdraw extends JFrame implements ActionListener {
    private JLabel l1,l2,l3,l4;
    private JTextField t1,t2;
    private JButton withdraw,cancel,deposit;
    private int x,amnt,t,w,account,d,r=0;
    ArrayList<Create> list1;
    private boolean flagWithdraw = false;
    private boolean flagAccNum = false;
    private boolean flagYes = false;
    private boolean flagAccMatch = false;
    private boolean flagNo = false;
    public Withdraw() {
        super("DEPOSIT/WITHDRAWAL SLIP");
        Container c=getContentPane();
        c.setLayout(new GridLayout(4,2));
        Font f1=new Font("Times New Roman",Font.BOLD,20);
        l1=new JLabel("Enter ACCOUNT NO:");
        l1.setFont(f1);
        l1.setForeground(Color.GRAY);
        JPanel bpanel=new JPanel();
        bpanel.add(l1);
        bpanel.setBackground(new Color(0,0,64));
        l2=new JLabel("ENTER AMOUNT:");
        l2.setFont(f1);
        l2.setForeground(Color.GRAY);
        JPanel b1panel=new JPanel();
        b1panel.add(l2);
        b1panel.setBackground(new Color(0,0,64));
        l3=new JLabel("DEPOSIT:");
        l3.setFont(f1);
        l3.setForeground(Color.GRAY);
        JPanel b2panel=new JPanel();
        b2panel.add(l3);
        b2panel.setBackground(new Color(0,0,64));
        t1=new JTextField();
        t2=new JTextField();
        withdraw=new JButton("WITHDRAW");
        withdraw.addActionListener(this);
        JPanel dpanel=new JPanel();
        dpanel.add(withdraw);
        dpanel.setBackground(new Color(0,0,64));
        deposit=new JButton("DEPOSIT");
        deposit.addActionListener(this);
        JPanel d1panel=new JPanel();
        d1panel.add(deposit);
        d1panel.setBackground(new Color(0,0,64));
        cancel=new JButton("CANCEL");
        cancel.addActionListener(this);
        JPanel epanel=new JPanel();
        epanel.add(cancel);
        epanel.setBackground(new Color(0,0,64));
        l4=new JLabel("");
        l4.setFont(f1);
        l4.setForeground(Color.GRAY);
        JPanel b3panel=new JPanel();
        b3panel.add(l4);
        b3panel.setBackground(new Color(0,0,64));
        c.add(bpanel);c.add(t1);
        c.add(b1panel);c.add(t2);
        c.add(dpanel);c.add(d1panel);
```

```
68          c.add(epanel);c.add(b3panel);
69          setSize(500,500);
70          setLocation(100,100);
71          setResizable(false);
72          setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
73          setVisible(true);
74      }
75      public void actionPerformed(ActionEvent e) {
76          String search=t1.getText();//search  ------->account number
77          if(e.getSource()==withdraw) {
78              String vaccNum = t1.getText();
79              String vwithdrw = t2.getText();
80              String accNumpattern = "^[0-9]{4}" ;
81              Scanner   scan1  =  new Scanner( vaccNum ) ;
82              String   matched1  =  scan1.findInLine( accNumpattern ) ;
83              if ( matched1 == null ) {
84                  JOptionPane.showMessageDialog(this, "INVALID ACCOUNT NUMBER\nIt
    should contain only digits with minimum length of 4 digit.", "Error",
    JOptionPane.ERROR_MESSAGE);
85                  t1.setText("");
86              }
87              else
88                  flagAccNum   = true;
89              String withdrawpattern = "^[0-9]{3}" ;
90              Scanner   scan  =  new Scanner( vwithdrw ) ;
91              String   matched  =  scan.findInLine( withdrawpattern ) ;
92              if ( matched == null ) {
93                  JOptionPane.showMessageDialog(this, "INVALID WITHDRAW AMOUNT\nIt
    should contain only digits with minimum withdraw amount of INR 100.", "Error",
    JOptionPane.ERROR_MESSAGE);
94                  t2.setText("");
95              }
96              else
97                  flagWithdraw    = true;
98              if(flagAccNum == true) {
99                  if(flagWithdraw == true) {
100                     try {
101                         FileInputStream fin=new FileInputStream("Regis.dat");
102                         ObjectInputStream oin=new ObjectInputStream(fin);
103                         list1=(ArrayList<Create>)oin.readObject();
104                         for(Create re : list1) {
105                             if(re.getAc().equals(search)) {
106                                 flagAccMatch  = true;
107                                 w=Integer.parseInt(t2.getText());
108                                 amnt=Integer.parseInt(re.getInitialamnt());
109                                 if(amnt>=500) {
110                                     flagNo  = true;
111                                     t = amnt - w;
112                                     if(t >= 500) {
113                                         account=Integer.parseInt(t1.getText());
114                                         d=0;
115                                         flagYes = true;
116                                     }
117                                 }
118                             }
119                         }
120                         if((flagAccMatch == true) && (flagNo == true) && (flagYes
    == true)) {
121                             new Actotalcreate(account,w,d,t);
122                             new TotalUpdate(account,t);
123                             JOptionPane.showMessageDialog(this, "Withdraw done
    successfully");
```

```
124                             new Passdisp(t1.getText());
125                             t1.setText("");
126                             t2.setText("");
127                         }
128                         if((flagAccMatch == true) && (flagNo == true) && (flagYes
    == false)) {
129                             JOptionPane.showMessageDialog(this, "Insufficient
    balance.\nAccount balance must be 500 after withdraw.", "Error",
    JOptionPane.ERROR_MESSAGE);
130                             t1.setText("");
131                             t2.setText("");
132                         }
133                         if((flagAccMatch == true) && (flagNo == false) && (flagYes
    == false)) {
134                             JOptionPane.showMessageDialog(this, "Account number has
    not the minimum balance of INR 500.", "Error", JOptionPane.ERROR_MESSAGE);
135                             t1.setText("");
136                             t2.setText("");
137                         }
138                         if((flagAccMatch == false) && (flagNo == false) && (flagYes
    == false)) {
139                             JOptionPane.showMessageDialog(this, "Account number
    does not exists", "Error", JOptionPane.ERROR_MESSAGE);
140                             t1.setText("");
141                             t2.setText("");
142                         }
143                         flagYes = false;
144                         flagAccMatch = false;
145                         flagNo = false;
146                     }
147                     catch(Exception e1) {
148                         JOptionPane.showMessageDialog(this, "No file found in data
    base", "Error", JOptionPane.ERROR_MESSAGE);
149                         t1.setText("");
150                         t2.setText("");
151                     }
152                 }
153             }
154             flagAccNum =false;
155             flagWithdraw = false;
156         }
157         if(e.getSource()==cancel) {
158             int rply = JOptionPane.showConfirmDialog(this, "Are you sure to
    quit?");
159             if(rply == JOptionPane.YES_OPTION) {
160                 new Userwindow("");
161                 setVisible(false);
162             }
163         }
164         if(e.getSource()==deposit) {
165             String vaccNum = t1.getText();
166             String vwithdrw = t2.getText();
167             String accNumpattern = "^[0-9]{4}" ;
168             Scanner   scan1  =  new Scanner( vaccNum ) ;
169             String   matched1  =  scan1.findInLine( accNumpattern ) ;
170             if ( matched1 == null ) {
171                 JOptionPane.showMessageDialog(this, "INVALID ACCOUNT NUMBER\nIt
    should contain only digits with minimum length of 4 digit.", "Error",
    JOptionPane.ERROR_MESSAGE);
172                 t1.setText("");
173             }
174             else
```

```
175                 flagAccNum   = true;
176             String withdrawpattern = "^[0-9]{3}" ;
177             Scanner   scan  =  new Scanner( vwithdrw ) ;
178             String   matched  =  scan.findInLine( withdrawpattern ) ;
179             if ( matched == null ) {
180                 JOptionPane.showMessageDialog(this, "INVALID WITHDRAW AMOUNT\nIt
    should contain only digits with minimum withdraw amount of INR 100.", "Error",
    JOptionPane.ERROR_MESSAGE);
181                 t2.setText("");
182             }
183             else
184                 flagWithdraw    = true;
185             if(flagAccNum == true) {
186                 if(flagWithdraw == true) {
187                     try {
188                         FileInputStream fin=new FileInputStream("Regis.dat");
189                         ObjectInputStream oin=new ObjectInputStream(fin);
190                         list1=(ArrayList<Create>)oin.readObject();
191                         for(Create re : list1) {
192                             if(re.getAc().equals(search)) {
193                                 flagAccMatch = true;
194                                 amnt=Integer.parseInt(re.getInitialamnt());
195                                 w=Integer.parseInt(t2.getText());
196                                 t=(amnt+w);
197                                 account=Integer.parseInt(t1.getText());
198                                 d=0;
199                                 r++;
200                             }
201                         }
202                         if((flagAccNum == true) && (flagWithdraw== true) &&
    (flagAccMatch == true)) {
203                                 new Actotalcreate(account,d,w,t);
204                                 new TotalUpdate(account,t);
205                                 JOptionPane.showMessageDialog(this, "Deposite done
    successfully");
206                                 new Passdisp(t1.getText());
207                                 t1.setText("");
208                                 t2.setText("");
209                         }
210                         if(flagAccMatch == false) {
211                             JOptionPane.showMessageDialog(this, "Account number
    does not exists");
212                             t1.setText("");
213                             t2.setText("");
214                         }
215                         flagAccMatch = false;
216                     }catch(Exception e1) {
217                         JOptionPane.showMessageDialog(this, "No file found in data
    base", "Error", JOptionPane.ERROR_MESSAGE);
218                         t1.setText("");
219                         t2.setText("");
220                     }
221                 }
222             }
223             flagAccNum =false;
224             flagWithdraw = false;
225         }
226     }
227 }
```