```
import javax.swing.*;
import java.io.IOException;
import javax.imageio.ImageIO;

//import Dementia.GetReady.AudioPlayer.TextAreaHandler;


import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.io.*;
import java.util.Calendar;
import java.text.SimpleDateFormat;


import javax.sound.sampled.AudioInputStream;
import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.Clip;
import javax.sound.sampled.FloatControl;
import javax.sound.sampled.LineUnavailableException;
import javax.sound.sampled.UnsupportedAudioFileException;


//dementia spin - what abt people that are poor? they can't afford assisting living...
//bridging gap between lower-class dementia people and upper-class dementia people


public class Dementia extends JFrame implements ActionListener{
  CardLayout cl;
  JPanel contentPane;
  public static void main(String[] args) throws UnsupportedAudioFileException, IOException,
LineUnavailableException {
      Dementia dd = new Dementia();
  }
  public Dementia() throws UnsupportedAudioFileException, IOException,
LineUnavailableException{
      setSize(500, 900);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      setVisible(true);
      contentPane = new JPanel();
```

```
    JButton back = new JButton("Back to Home");
    back.addActionListener(this);
    add(back, BorderLayout.NORTH);
    add(contentPane, BorderLayout.CENTER);
    cl = new CardLayout();
    contentPane.setLayout(cl);
    contentPane.add(new HomePage(), "homePage");
    contentPane.add(new GetReady(), "getReady");
    contentPane.add(new JournalPanel(), "journalPanel");
    contentPane.add(new LovedOnes(), "lovedOnes");
    contentPane.add(new AboutDementia(), "aboutDementia");
    cl.show(contentPane, "homePage");
  }
  public void actionPerformed(ActionEvent e){
   if(e.getActionCommand().equals("Back to Home")){
      cl.show(contentPane, "homePage");
   }
  }
  class AboutDementia extends JPanel{
   public AboutDementia(){
      setLayout(new BorderLayout());
      Font font = new Font("SanSerif", Font.PLAIN, 30);
         setLayout(new BorderLayout());
         Color green = new Color(255,226,246);
        Color pink = new Color(173, 232, 162);
         setBackground(pink);
         JLabel aboutDementia = new JLabel("About Dementia");
         //aboutDementia.setFont(font);
         JPanel east = new JPanel();
         east.setBackground(pink);
         JPanel west = new JPanel();
         west.setBackground(pink);
         JPanel south = new JPanel();
         south.setBackground(pink);
         add(east, BorderLayout.EAST);
         add(west, BorderLayout.WEST);
         add(south, BorderLayout.SOUTH);
         add(aboutDementia, BorderLayout.NORTH);
         JTextArea dementiaInfo = new JTextArea("Dementia is a general term that describes a
decline in cognitive function, including memory loss, difficulty with problem-solving, and changes
in behavior or personality. Dementia is not a specific disease, but rather a group of symptoms
caused by several underlying conditions, including Alzheimer's disease, vascular dementia,
Lewy body dementia, frontotemporal dementia, and others.\n\nSymptoms of dementia: \n -
Memory loss \n - Difficulty with communication\n - Difficulty with problem-solving and
```

```
reasoning\n - Difficulty with daily tasks, such as dressing and bathing\n - Changes in personality and behavior, including depression, anxiety, and      agitation\n - Loss of interest in previously enjoyed activities\n - Misplacing items or getting lost in familiar places \n\n\n\nRisk Factors: \n - age(above 65) \n - high blood pressure \n - high blood sugar \n - overweight/obese \n - smoking \n - too much alcohol \n - physically inactive \n - socially isolated \n - depression ");
            dementiaInfo.setBackground(green);
            dementiaInfo.setEditable(false);
            dementiaInfo.setWrapStyleWord(true);
            dementiaInfo.setLineWrap(true);
            add(dementiaInfo, BorderLayout.CENTER);
        }
    }

    class HomePage extends JPanel implements ActionListener {

        JLabel timeLabel;
        public HomePage(){
            Color color = new Color(253, 202, 162);
            Font font = new Font("Monospaced", Font.PLAIN, 40);
            Font font2 = new Font("Monospaced", Font.PLAIN, 30);

            Color pink = new Color(255,226,246);
            Color green = new Color(173, 232, 162);
            setBackground(pink);
            setLayout(new BorderLayout());
            SouthPanel sp = new SouthPanel();
            sp.setPreferredSize(new Dimension(40,30));
            add(sp, BorderLayout.NORTH);
            SouthPanel ss = new SouthPanel();
            ss.setPreferredSize(new Dimension(40,30));
            add(ss, BorderLayout.SOUTH);
            SouthPanel1 sk = new SouthPanel1();
            sk.setPreferredSize(new Dimension(40,30));
            add(sk, BorderLayout.EAST);
            SouthPanel1 sk2 = new SouthPanel1();
            sk2.setPreferredSize(new Dimension(40,30));
            add(sk2, BorderLayout.WEST);
            JPanel center = new JPanel();
            add(center, BorderLayout.CENTER);
            center.setBackground(green);
            center.setLayout(new GridLayout(5,1, 30, 30));
            JLabel connect = new JLabel("Connect Now");
            connect.setPreferredSize(new Dimension(70, 40));
             connect.setFont(font);
```

```
    center.add(connect);
    connect.setHorizontalAlignment(JLabel.CENTER);
    JButton getReadyButton = new JButton("GetReady");
    Color lav = new Color(190, 205, 247);
    getReadyButton.setOpaque(true);

    getReadyButton.setBackground(lav);
    getReadyButton.setPreferredSize(new Dimension(100, 30));
    getReadyButton.setFont(font2);
    getReadyButton.addActionListener(this);
    center.add(getReadyButton);
    JButton lovedOneButton = new JButton("Connect");
    lovedOneButton.setOpaque(true);
    lovedOneButton.setBackground(lav);
    JButton journalButton = new JButton("Journal");
    journalButton.addActionListener(this);
    journalButton.setFont(font2);
    center.add(journalButton, "journal");
    //pack();

    lovedOneButton.setFont(font2);

    lovedOneButton.setHorizontalAlignment(JLabel.CENTER);
    lovedOneButton.addActionListener(this);
    center.add(lovedOneButton);
    JButton aboutMeButton = new JButton("AboutDementia");
    aboutMeButton.setOpaque(true);
    aboutMeButton.setBackground(lav);
    aboutMeButton.setFont(font2);
    aboutMeButton.addActionListener(this);
    center.add(aboutMeButton);
}
public void actionPerformed(ActionEvent e){
 if(e.getActionCommand().equals("GetReady")){
     cl.show(contentPane, "getReady");
 }
 if(e.getActionCommand().equals("Connect")){
     cl.show(contentPane, "lovedOnes");
 }
 if(e.getActionCommand().equals("Journal")){
     cl.show(contentPane, "journalPanel");
 }if(e.getActionCommand().equals("AboutDementia")){
     cl.show(contentPane, "aboutDementia");
 }
```

```
 }
 class SouthPanel extends JPanel{
    public SouthPanel(){}
        public void paintComponent(Graphics g){
        Image joe = new ImageIcon("flower.png").getImage();
        for(int x = 10; x<40*16; x+=40){
         g.drawImage(joe, x, 10, 20, 20, null);
        }
    }
 }
 class SouthPanel1 extends JPanel{
    public SouthPanel1(){
    }
    public void paintComponent(Graphics g){
        Image brain = new ImageIcon("braing.png").getImage();

        Image joe = new ImageIcon("flower.png").getImage();
        for(int x = 10; x<40*90; x+=40){
         g.drawImage(joe, 10, x, 20, 20, null);
        }
        g.drawImage(brain, 10, 100, 100, 40, null);
    }
 }
}


//add a

class GetReady extends JPanel {
    JLabel timeLabel;
    JButton yes, no;
    JPanel TwoCardsPanel;
    CardLayout cardLayout2;
    String inputDinner;
    JTextField input;
    JButton submit;
    boolean keepPlayingSound = true;
    boolean dinnerQuestion = false;
    boolean breakQuestion = false;
    boolean snackQuestion = false;
    boolean lunchQuestion = false;
    JPanel card2, card1;
    JLabel question, question2, question4;
```

```
    int count = 0;
    public GetReady() throws UnsupportedAudioFileException, IOException,
LineUnavailableException {
        question2 = new JLabel();
        Color pink = new Color(255,226,246);
        Color green = new Color(173, 232, 162);
      cardLayout2 = new CardLayout();
        setLayout(new BorderLayout());
        timeLabel = new JLabel();
        Font font = new Font("Arial", Font.BOLD, 24);
        timeLabel.setFont(font);
        timeLabel.setHorizontalAlignment(JLabel.RIGHT);
        add(timeLabel, BorderLayout.NORTH);
        TimeHandler ta = new TimeHandler();
        Timer timer = new Timer(1000, ta);
        timer.start();
        TwoCardsPanel = new JPanel();
        TwoCardsPanel.setBackground(pink);
        TwoCardsPanel.setLayout(cardLayout2);
        card1 = new JPanel(new BorderLayout());
        card1.setBackground(pink);
        yes = new JButton("yes");
        yes.addActionListener(new YesNoHandler());
        no = new JButton("no");
        no.addActionListener(new YesNoHandler());
        JPanel yesAndNoPanel = new JPanel(new GridLayout(2,1, 40, 40));
        yesAndNoPanel.add(yes);
        Color lav = new Color(190, 205, 247);
        yesAndNoPanel.setBackground(lav);

        yes.setBackground(lav);
        yesAndNoPanel.add(no);
        no.setBackground(lav);
        card1.add(yesAndNoPanel, BorderLayout.CENTER);
        card2 = new JPanel(new BorderLayout());
        card2.setBackground(pink);

        TAHandler tah  = new TAHandler();


        input = new JTextField("");
        input.addActionListener(tah);
```

```
        JPanel centerPanel = new JPanel(new GridLayout(2,1,30,30));
        card2.add(centerPanel, BorderLayout.CENTER);
        centerPanel.add(input);
        submit = new JButton("I have completed this task!");
        SubmitHandler sh = new SubmitHandler();
        submit.addActionListener(tah);
        centerPanel.add(submit);
        inputDinner = input.getText();
        TwoCardsPanel.add(card1, "card1");
        TwoCardsPanel.add(card2, "card2");
        add(TwoCardsPanel, BorderLayout.CENTER);
    }

    class TimeHandler implements ActionListener{
        public void actionPerformed(ActionEvent e){
            updateTime();
        }

        public void updateTime() {

            Calendar cal = Calendar.getInstance();
            SimpleDateFormat sdf = new SimpleDateFormat("hh:mm:ss a");
            String time = sdf.format(cal.getTime());
            String hours = sdf.format(cal.getTime());
            String temp = hours;
            hours = hours.substring(0, 2);
            String amOrpm = temp.substring(9);
         //    if(Integer.parseInt(hours)>=7 && Integer.parseInt(hours)<=11 &&
amOrpm.equals("PM") && count <1){
         //        card1.add(new JLabel("<html>Good evening! Would you like dinner?</html>"));

         //        question.setText("What do you want for dinner?");
         //        card2.add(question);

         //        count++;
         //    }
              //f(Integer.parseInt(hours)>=7 && Integer.parseInt(hours)<=11 &&
amOrpm.equals("AM") && count< 1){
                 question2.setText("What do you want for breakfast?");
                  JLabel label1 = new JLabel("<html>Hello! Would you like a breakfast?</html>");
                  label1.setPreferredSize(new Dimension(300,100));
                  card1.add(label1, BorderLayout.NORTH);

                  card2.add(question2);
```

```
                count++;
            //}

            //  if(Integer.parseInt(hours)>=12 && Integer.parseInt(hours)<=3 &&
amOrpm.equals("PM") && count < 1){
            //     card1.add(new JLabel("<html>Good afternoon! Would you like lunch?</html>"));

            //     question4.setText("What do you want for lunch?");
            //     card2.add(question4);
            //     count++;
            // }

            if (count <1) {
                //System.out.println("hihihihihihi");
                JLabel label = new JLabel("<html>Hello! Would you like a snack?</html>");
                label.setPreferredSize(new Dimension(300,100));
                card1.add(label);

                JLabel question3 = new JLabel("What do you want for snacks?");
               card2.add(question3, BorderLayout.NORTH);
                count++;
            }
           timeLabel.setText(time);
        }
    }

    class YesNoHandler implements ActionListener{
        public void actionPerformed(ActionEvent e) {
            String command = e.getActionCommand();
            if(command.equals("yes")) {
                cardLayout2.show(TwoCardsPanel, "card2");
            }
        }
    }
    Clip clip;

    class TAHandler implements ActionListener {
        FloatControl fc;
        public void actionPerformed(ActionEvent e){
            if(input.getText().equals("mac and cheese")){
                System.out.println("hi");
                AudioPlayer ap = new AudioPlayer("MacNCheese1 (1).wav");
                fc = (FloatControl)clip.getControl(FloatControl.Type.MASTER_GAIN);
                if (clip == null) return;
```

```
                clip.setFramePosition(0);
                System.out.println("keep playing " + keepPlayingSound);
                if (keepPlayingSound) {
                    System.out.println(clip);
                    clip.loop(Clip.LOOP_CONTINUOUSLY);
                }
                if(!keepPlayingSound) {
                    System.out.println("hi");
                   //clip.open(ais);
                    fc.setValue(0);
                    System.out.println(this);
                    System.out.println(clip);
                    clip.stop();
                    return;
                }
            }
        }
        public void stop() {
            if (clip.isRunning()) {
                System.out.println("running " + keepPlayingSound);
                clip.stop();
            }
        }
       }

    class SubmitHandler implements ActionListener{
        public void actionPerformed(ActionEvent e){
            if(e.getActionCommand().equals("I have completed this task!")){
                keepPlayingSound = false;
                System.out.println("button clicked &&  " + keepPlayingSound);
            }
        }
    }

    class AudioPlayer{
        public AudioPlayer(String filename)
        {
            try
            {
                AudioInputStream audioInputStream =
AudioSystem.getAudioInputStream(getClass().getResourceAsStream(filename));
                clip = AudioSystem.getClip();
                clip.open(audioInputStream);
            }
```

```
            catch(Exception ex)
            {
                System.out.println("Error loading audio: " + ex.getMessage());
            }
        }
    }
}

    class LovedOnes extends JPanel
    {
        private JButton nameOne;
        private JButton nameTwo;
        private JButton nameThree;
        private JButton nameFour;
        private CardLayout cards;
        private JPanel cardsHolder;
        private String choose;
        private JTextArea info;
        private PeopleInfo peopleinfo;
        private JLabel name;

            public LovedOnes()
            {
            peopleinfo = new PeopleInfo();
            cardsHolder = new JPanel();
            cards = new CardLayout();
            cardsHolder.setLayout(cards);
            ChooseContacts cc = new ChooseContacts();
            PeopleInfo pi = new PeopleInfo();
            cardsHolder.add(cc, "Choose Contacts");
            pi.setPreferredSize(new Dimension(300,700));
            cardsHolder.add(pi, "People Info");
            add(cardsHolder);
            cards.show(cardsHolder, "Choose Contacts");
                }
        class ButtonHandler implements ActionListener
        {
            public void actionPerformed(ActionEvent evt)
            {
                if(evt.getSource() == nameOne)
                {
                    choose = "Mika.jpg";
                }
                if(evt.getSource() == nameTwo)
```

```
        {
            choose = "Katrina.jpeg";
        }
        if(evt.getSource() == nameThree)
        {
            choose = "Tashvi.jpg";
        }
        if(evt.getSource() == nameFour)
        {
            choose = "Jivika.jpg";
        }
        peopleinfo.repaint();
        cards.show(cardsHolder, "People Info");
    }
}

 class ChooseContacts extends JPanel
 {
    public ChooseContacts()
    {
        Color pink = new Color(255,226,246);
        Color green = new Color(173, 232, 162);
        setBackground(pink);

        ButtonHandler lo = new ButtonHandler();
        Font font = new Font("SanSerif", Font.PLAIN, 50);
        setLayout(new GridLayout(4, 1));
        JPanel contactOne = new JPanel();
        nameOne = new JButton("Mika - Dog");
        nameOne.setFont(font);
        nameOne.addActionListener(lo);
        contactOne.add(nameOne);
        JPanel contactTwo = new JPanel();
        nameTwo = new JButton("Katrina - Friend");
        nameTwo.addActionListener(lo);
        nameTwo.setFont(font);
        contactTwo.add(nameTwo);
        JPanel contactThree = new JPanel();
        nameThree = new JButton("Tashvi - Friend");
        nameThree.setFont(font);
        nameThree.addActionListener(lo);
        contactThree.add(nameThree);
        JPanel contactFour = new JPanel();
        nameFour = new JButton("Jivika - Friend");
```

```
        nameFour.setFont(font);
        nameFour.addActionListener(lo);

        contactFour.add(nameFour);

        add(contactOne);
        add(contactTwo);
        add(contactThree);
        add(contactFour);
    }
}

class PeopleInfo extends JPanel
{
    public PeopleInfo()
    {
        Color pink = new Color(255,226,246);
        //setBackground(Color.BLUE);
        setBackground(pink);
        setLayout(new BorderLayout());
        name = new JLabel("");
        //name.setBounds(200, 100, 450, 400);
        add(name, BorderLayout.NORTH);
        name.setPreferredSize(new Dimension(400, 400));
        info = new JTextArea("");
        info.setEditable(false);
                    info.setLineWrap(true);
                    info.setWrapStyleWord(true);
        add(info, BorderLayout.CENTER);
        JButton backButton = new JButton("Back");
        BackButtonHandler bbh = new BackButtonHandler();
        backButton.addActionListener(bbh);
        add(backButton, BorderLayout.SOUTH);
    }
    class BackButtonHandler implements ActionListener
    {
        public void actionPerformed(ActionEvent evt)
        {
            cards.show(cardsHolder, "Choose Contacts");
        }
    }
    public void paintComponent(Graphics g)
    {
        super.paintComponent(g);
```

```
            info.setText(choose);

            Image picture = null;
            File image = new File(choose);
            try
            {
               picture = ImageIO.read(image);
            }
            catch(IOException e)
            {
               System.err.println("The image " + choose + " cannot be loaded.");
            }

            g.drawImage(picture, 10, 10, 300, 400, this);

            if(choose.equals("Mika.jpg"))
            {
               info.setText("Name: Mika\n" + "Contact Info: through Lasya
(lasyaissocool@gmail.com)\n"
               + "Fun Facts: Likes to Sleep and Sleep in lap\nFavorite Food: Peanut Butter");
            }
            if(choose.equals("Katrina.jpeg"))
            {
                info.setText("Name: Katrina\n" + "Contact Info: xxx-xxx-xxxx\nFavorite Food:
Anything but Cheese"
               + "(unless its pizza ofc) \nFun Facts: absolute god at coding");
            }
            if(choose.equals("Tashvi.jpg"))
            {
               info.setText("Name: Tashvi\n" +"Contact Info: xxx-xxx-xxxx\nFavorite Food: Pizza"
               + "\nFun Facts: Has the coolest lunch boxes");
            }
            if(choose.equals("Jivika.jpg"))
            {
               info.setText("Name: Jivika\n" + "Contact Info: xxx-xxx-xxxx\nFavorite Food:
Cheetos!!!"
               + "\nFun Facts: Literal Speech God");
            }
         }
      }
   }

   class JournalPanel extends JPanel implements KeyListener
   {
```

```
String text1, text2, text3;
JTextArea sleepText, exerciseText, moodText;
public JournalPanel()
{
   Font font = new Font("SanSerif", Font.PLAIN, 30);
   Color pink = new Color(255,226,246);
   setBackground(pink);
   setLayout(new FlowLayout(FlowLayout.CENTER, 0, 30));
   JLabel journalName = new JLabel("Journal: Write your thoughts here!");
   journalName.setFont(font);
   add(journalName);
   sleepText = new JTextArea("enter # of sleep");
   sleepText.setWrapStyleWord(true);
   sleepText.setLineWrap(true);
   sleepText.setPreferredSize(new Dimension(300, 250));
   sleepText.addKeyListener(this);

   exerciseText = new JTextArea("enter your excercise goals");
   exerciseText.setWrapStyleWord(true);
   exerciseText.setLineWrap(true);
   exerciseText.setPreferredSize(new Dimension(300, 250));
   exerciseText.addKeyListener(this);

   moodText = new JTextArea("enter your mood today");
   moodText.setWrapStyleWord(true);
   moodText.setLineWrap(true);
   moodText.setPreferredSize(new Dimension(300, 250));
   moodText.addKeyListener(this);

   add(sleepText);
   add(exerciseText);
   add(moodText);

File outFile1 = new File("Sleep.txt");
File outFile2 = new File("Exercise.txt");
File outFile3 = new File("Mood.txt");

PrintWriter makesOutput1 = null;
PrintWriter makesOutput2 = null;
PrintWriter makesOutput3 = null;

try{
   makesOutput1 = new PrintWriter ( outFile1 );
   makesOutput2 = new PrintWriter ( outFile2 );
```

```
    makesOutput3 = new PrintWriter ( outFile3 );
}
catch( FileNotFoundException e){
    System.err.println("The file outputtedText.txt cannot be created.");
                        System.exit(1);
}
makesOutput1.println(text1);
makesOutput2.println(text2);
makesOutput3.println(text3);
}
@Override
public void keyTyped(KeyEvent evt) {
        PrintWriter pw1 = null;
        PrintWriter pw2 = null;
        PrintWriter pw3 = null;
        File outFile1 = new File("Sleep.txt");
        File outFile2 = new File("Exercise.txt");
        File outFile3 = new File("Mood.txt");

        try{
            pw1 = new PrintWriter(new FileWriter(outFile1, true));
            pw2 = new PrintWriter(new FileWriter(outFile2, true));
            pw3 = new PrintWriter(new FileWriter(outFile3, true));

        }
        catch (IOException ioex){
            System.err.println("Cannont append to");
            System.exit(1);
        }
        text1 = sleepText.getText();
        text2 = exerciseText.getText();
        text3 = moodText.getText();
        pw1.println(text1);
        pw2.println(text2);
        pw3.println(text3);

        pw1.close();
        pw2.close();
        pw3.close();
}
@Override
public void keyPressed(KeyEvent evt) {
    // TODO Auto-generated method stub
}
```

```
        @Override
        public void keyReleased(KeyEvent evt) {
            // TODO Auto-generated method stub
        }
    }
}
```