“PassBoard”

A MAJOR PROJECT REPORT

Submitted by

CHETAN RAIKWAR

(0203IT131001)

Under the Guidance of

Prof. RICHA NAKRA KURARIA

In partial fulfilment for the award of the degree

Of

BACHELOR OF ENGINEERING

In

Information Technology Engineering

DEPARTMENT OF INFORMATION TECHNOLOGY ENGINEERING

HITKARINI COLLEGE OF ENGINEERING & TECHNOLOGY, JABALPUR (MP)

RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL (MP)

HITKARINI COLLEGE OF ENGINEERING & TECHNOLOGY

JABALPUR (MP)

Approved by AICTE New Delhi & Govt. Of M.P.

(Affiliated to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal)

BONAFIDE CERTIFICATE

*This is to certify that the major project work entitled **"Passboard"** is done by **CHETAN RAIKWAR** under my guidance and supervision for the degree of **Bachelor of Engineering** in **Information Technology Engineering** of Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.) India. To the best of my knowledge and belief the dessertation embodies the work of the candidate themselves and that the work has not been submitted earlier in part or full for the award of any other degree.*

Prof. Amit Chandanan

Head of the Department

DATE .....

HITKARINI COLLEGE OF ENGINEERING & TECHNOLOGY

JABALPUR (MP)

Approved by AICTE New Delhi & Govt. Of M.P.

(Affiliated to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal)

AFFIDAVIT of CERTIFICATE

*This is to certify that the major project work entitled **"PassBoard"** by **CHETAN RAIKWAR** is approved for the award of degree of **Bachelor of Engineering** in **Information Technology Engineering** of **Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.) India**.*

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DATE ..... **DATE .....**

HITKARINI COLLEGE OF ENGINEERING & TECHNOLOGY

JABALPUR (MP)

Approved by AICTE New Delhi & Govt. Of M.P.

(Affiliated to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal)

APPROVAL CERTIFICATE

*This is to certify that the major project work entitled **"PassBoard"** is done by **CHETAN RAIKWAR** is approved for the award of degree of **Bachelor of Engineering** in **Information Technology Engineering** of **Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.) India**.*

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DATE ..... **DATE .....**

HITKARINI COLLEGE OF ENGINEERING & TECHNOLOGY

JABALPUR (MP)

Approved by AICTE New Delhi & Govt. Of M.P.

(Affiliated to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal)

DECLARATION

I declare that the major project entitled ***"PassBoard"*** *is my own work conducted under the supervision of* ***Prof. Amit Chandanan****,* ***Information Technology Engineering at, Hitkarini College of Engineering & Technology, Jabalpur****. I further declare that to the best of my knowledge, the minor project does not contain any part of any work which has been submitted for the award of any degree either in this University or in any other University without proper citation.*

CHETAN RAIKWAR

(0203IT131001)

ACKNOWLEDGEMENT

I am grateful to ***Prof. RICHA NAKRA KURARIA****,* ***Hitkarini College of Engineering & Technology, IT Department****, my* ***Mentor****, for her valuable guidance and encouragement throughut the duration of my major project work. I am greately benefited from the technical discussions I shared with her which helped me in understanding the finer aspects of Application Design.*

I also thank ***Prof. Dr. Shailesh Gupta, Principal Hitkarini College of Engineering & Technlogy, Jabalpur*** *and* ***Prof. Amit Chandanan, Head of Department, Information Technology Engineering*** *for helping me out continuously during the course of my work. Without their help it would have been difficult to complete this project within time.*

Thanks to each and everyone who helped me in some way or the other, in successful completion of the project work.

CHETAN RAIKWAR

(0203IT131001)

Contents

Concept

I have developed Passboard not to totally replace the windows On-screen keyboard, as they both serve two different purposes.

Windows On-Screen Keyboard is basically a part of Microsoft's "Ease of Access" Function. Thus, in Windows OSK, Ease of Access is above the security of course.

Passboard on the other hand, mimics the behavior or Windows OSK as well as Virtual Keyboard programs offered by various Security companies. Thus combining the features of both, Passboard is marginally advanced. However, any digital fortress can crack / fall, facing the mightiest and craftiest hacking attack, I wanted to make sure that Passboard shouldn't fall, not so soon.

The ideas leading to creation of Passboard were :-

- Standalone application (not some plugin).
- Usable for web form feeding as well as desktop application text feeding.
- Significantly consistent across various platform.
- Isolated from the Keyboard hooks (potential keylogging triggers).
- Safer to use on-screen, evading the risk of screen capturing and staring by fellow viewers.
- Capable to defeat the Keylogging activity by a massive margin.
- Lightweight.

How is Passboard safer ?

There are some key points which explain why passboard is a safer.

- ## Doesn't trigger Keyboard hooks

  The passboard is isolated from the keyboard hooks in core design. The buttons of the passboard do not trigger the physical keyboard signal, The activity is restricted within the application's environment only.

- ## Disturbs the concentraion of Fellow viewers, who try to watch & remember your on-screen activity while you feed some data

  The passboard has an "auto-shuffle" feature, this feature, when enabled, shuffles the whole keymap of the passboard everytime you press a "alphanumeric / symbol" key.
  This makes it hard for viewers to concentrate on screen at the same time, makes it impossible for them to track all your activities (button clicks) and remember them chronologically.

- ## Sends the data silently into System Clipboard and empties it after 10 seconds (sufficient time for pasting)

  When you choose the "Copy" option inside Passboard, the data you have fed is directly sent to "System Clipboard" for 10 seconds. In this time frame, you can "paste" this data anywhere.

After 10 seconds, the System Clipboard is emptied, Leaving no trace behind.

❖ Runs in Java Virtual Machine

The Passboard was written in Java, thus, it runs within JVM. Needless to say that all the security measures taken by JVM are automatically a part of Passboard too.

❖ Defeating Most of the Keylogging attempts

The Passboard; being silent, isolated & JVM based application, is mighty enough to evade most of the conventional keylogging attempts.
The Passboard avoids near about all the triggers which can result in beginning of Keylogging session, such few triggers are,

1. Virtual key connected to physical key code behind the scenes.
2. Cut/copy options or their triggers (ctrl+x, ctrl+c).

The Passboard avoids conventional system calls and it works, in a way, somewhat underground.

❖ Versatile

Windows OSK is definitely not meant for anything regarding security. There are some Information Security Vendors who offer virtual keyboards but they have a few drawbacks

1. Their virtual keyboards are browser plugins mostly.
2. They are not standalone, they are a part of bigger security suite.
3. They are costly.

The Passboard effectively deals with these issues, The Passboard is an standalone applications thus it can be used for activity inside the browser as well as on desktop applications too.
The activity of Passboard differs significantly. The other virtual keyboards use direct input into field, the Passboard on the other hand uses System Clipboard and a Timer to ensure security.
If made commercial, The passboard doesn't seem to be as costlier as the Virtual Keyboard + Parent Security Suite combo.

Design

The Passboard is pretty straight forward in the core, the only tricky part was the GUI.

Fragments in which the project was designed :-

1. User Interface.
2. Core Functionality.
3. Security Enhancement.

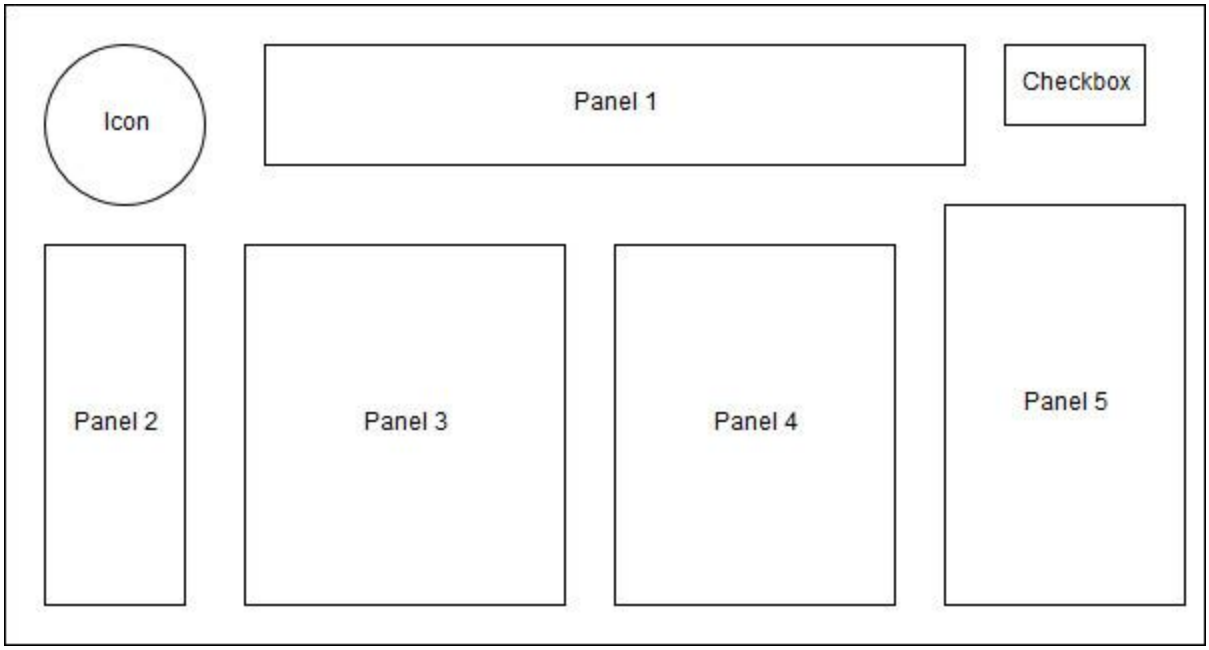
❖ Requirement Gathering

1. A System supporting Java Development Kit latest Version.
2. Java Development Kit (for consistency across platforms).
3. Any tool (preferrably online) to produce icons and images needed.

❖ User Interface Design

The GUI consists of the following :-

1. A Jframe (The main window to contain all panels)
2. 5 Jpanels (Numeric / Alphabetic / Symbolic / Special keys & Result field)
3. Several Buttons (Numeric, Alphabetic, Symbolic & Special)
4. 2 JCheckBoxes (Shuffle, Show/Hide text)



❖ Core Functionality

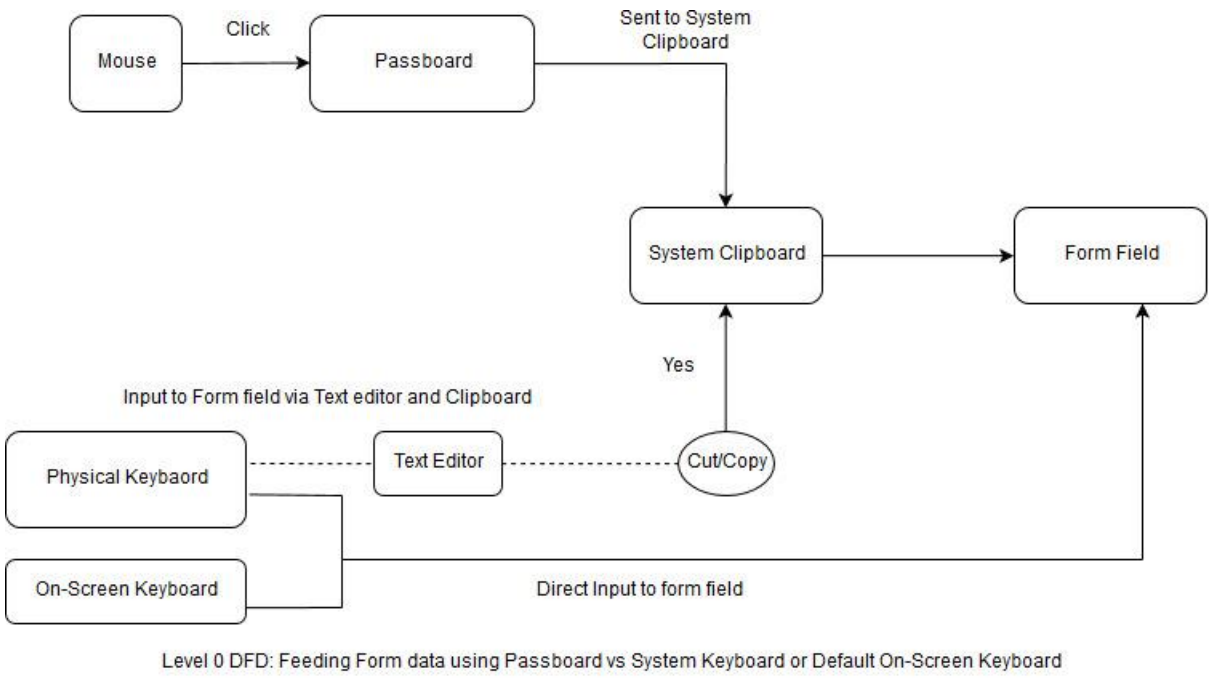
The Core Functionality includes these following activities :-

1. Display of Titles on desired buttons.
2. Adding actionListeners to the buttons.
3. Processing of the main text based on the the inputs.

❖ Security Enhancement

1. Shuffling of the Keymap (if selected)
2. Showing / Hiding the main text (if selected)
3. Timer to automatically empty the System Clipboard.

Data Flow of the Program



Level 0 DFD: Feeding Form data using Passboard vs System Keyboard or Default On-Screen Keyboard

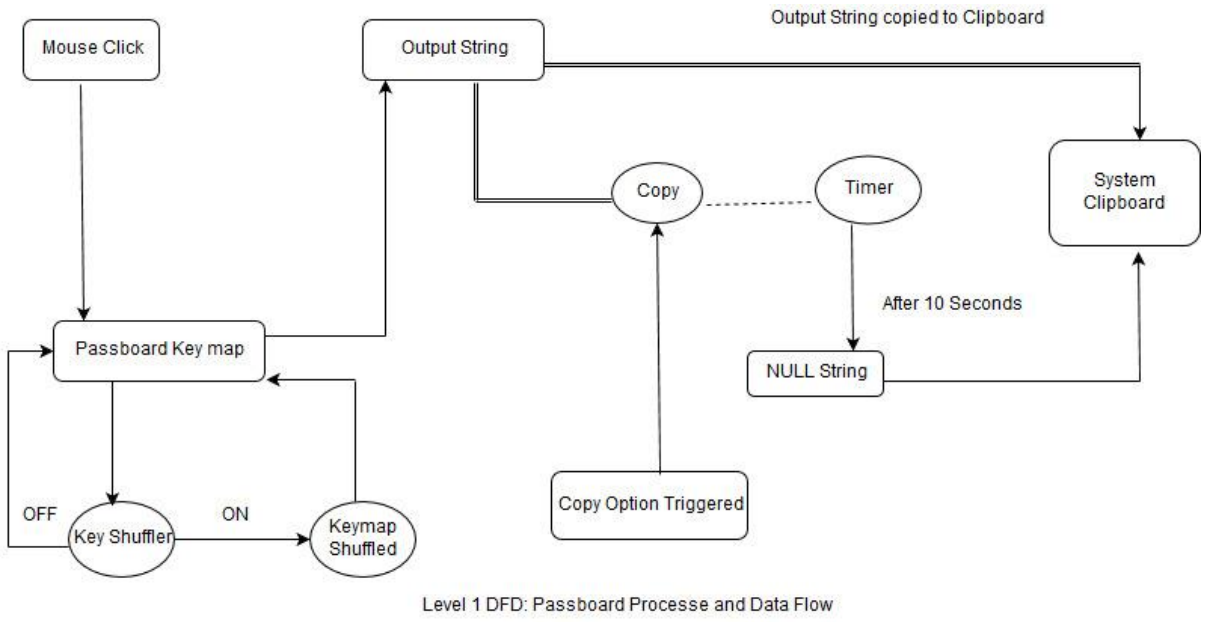
The Level 0 DFD of the Program Presents some interesting Information about how people interact with Passboard and with Physical Keyboard and On-Screen Keyboard of Windows.

Description:

Passboard receives the input by Mouse or any Pointing device. From where, the desired result is sent directly to system clipboard, from where in turn, it can be fed into the desired data field.

On the other hand, Physical Keyboard is used to either feed the data directly into some data field as well as via any intermediate application (text editor followed by copy/pasting).

On-Screen Keyboard is a virtual copy of physical keyboard with no functional differences. Both trigger the Keyboard signals.



Level 1 DFD: Passboard Processe and Data Flow

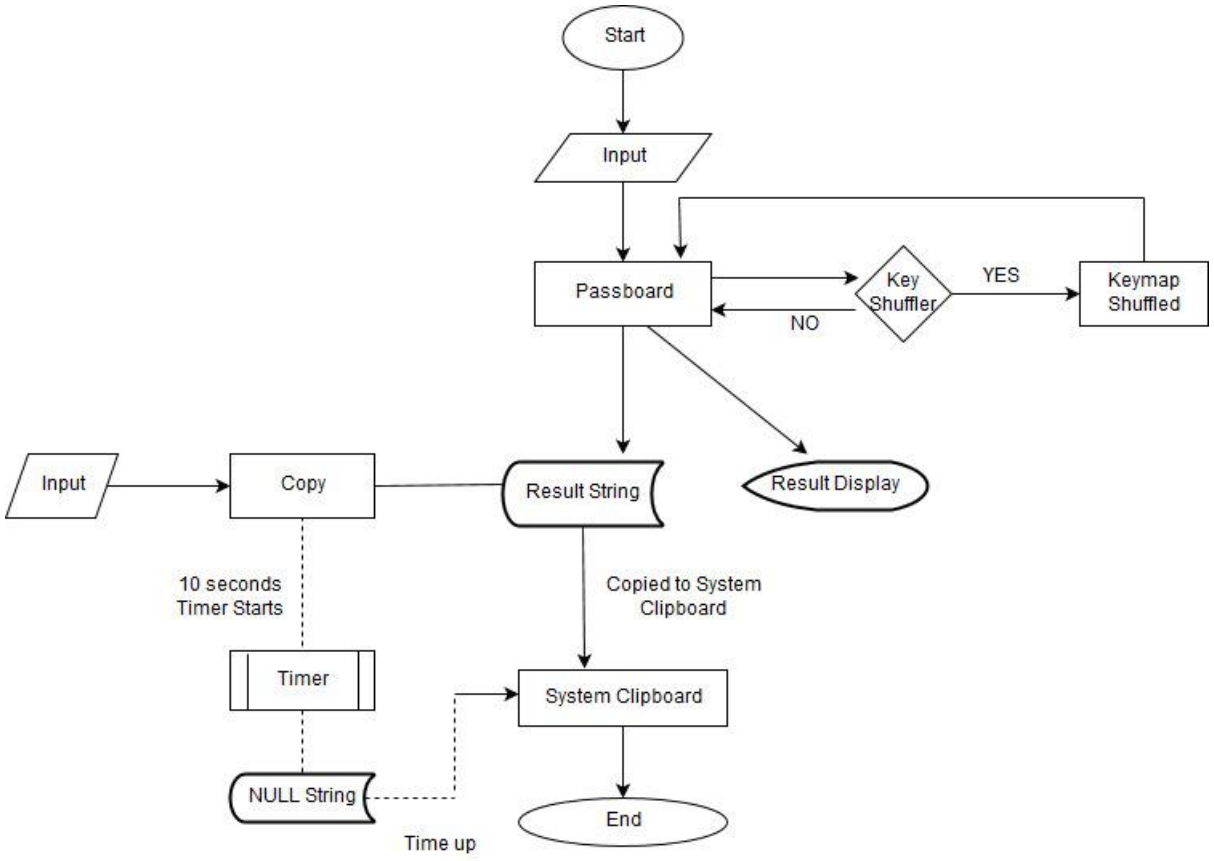
The Level 1 DFD of the Passboard shows the internal processes and Data flow directions.

Description:

The Passboard receives the input from the Mouse or any Pointing device.

The input is directly received on buttons of Passboard. Now after the input, It is checked whether the KeyShuffler is set, for a short time, the program enters in a finite loop.
In case the Keyshuffler is set, the Whole Keymap (except the special keys) is shuffled and the flow goes back to the point prior to the loop.
If Keyshuffler is not set, the Program goes to the point prior to the loop without Keymap shuffling.
The input received on the buttons of Passboard is translated to the desired output and it is appended in the output string (result).

Primary flow of the Program ends here. Now secondary flow starts if user decides to use the result. When the Copy option is selected, A Timer function is invoked and the Result is sent directly to the system clipboard.
The Timer keeps running until next 10 seconds. Once the time is up, the Timer function sets the Result String of System Clipboard to NULL.



Simple Flowchart showing Passboard Operations

The flow diagram above explains the Passboard operations in simplest way.

Source Code

```
import javax.swing.*;
import java.awt.*;
import java.awt.datatransfer.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.border.*;

class KeyPad extends JPanel {

   boolean capsState = false, autoShuffle = false, coverState = false;
   String resultString = "";
   String[] KeyMap = {"1234567890",
                      "qwertyuiopasdfghjklzxcvbnm",
                      "`~!@#$%^&*()_+-=\\|[]{}:;'\"<>,./?"
                     };
   JCheckBox shuffle, textCover;
   JPanel[] panels;
   JTextField resultField;
   JButton[] buttons;
   JButton backSpace, clear, copy, space, capsLock;
   GridBagConstraints GBC;

   public KeyPad() {

      /* switch to desired UI */
      UIManager.LookAndFeelInfo[] infos = UIManager.getInstalledLookAndFeels();

      try {
```

```
    UIManager.setLookAndFeel(infos[1].getClassName());
}
catch (Exception x) {
    Toolkit.getDefaultToolkit().beep();
    JOptionPane.showMessageDialog(null, "Recommended UI support not found. \n"
            + "Switching to default UI.");
}

setBackground(Color.GRAY);
setLayout( new GridBagLayout() );
GBC = new GridBagConstraints();

makePanels();

GBC.weightx = 100;
GBC.weighty = 100;

GBC.gridx = 0;
GBC.gridy = 1;
add( new JLabel(new ImageIcon("F:\\green.png")),GBC);

// result field
GBC.gridx = 1;
GBC.gridy = 1;
GBC.gridwidth = 2;

GBC.fill = GridBagConstraints.HORIZONTAL;
add(panels[0], GBC);

// numeric pad
GBC.gridx = 0;
```

```
    GBC.gridy = 2;
    GBC.gridwidth = 1;
    GBC.fill = GridBagConstraints.NONE;
    add(panels[1], GBC);

    // alphabetic pad
    GBC.gridx = 1;
    GBC.gridy = 2;

    add(panels[2], GBC);

    // symbol pad
    GBC.gridx = 2;
    GBC.gridy = 2;
    add(panels[3], GBC);

    // special key pad
    GBC.gridx = 3;
    GBC.gridy = 2;
    add(panels[4], GBC);

   prepareBoard();

}

private void makePanels() {

   panels = new JPanel[5];

   for(int i=0; i<panels.length; i++) {
```

```
      panels[i] = new JPanel();
      panels[i].setBackground(Color.GRAY);
    }

    Border panelBorder = BorderFactory.createLoweredBevelBorder();

    Border titleBorder = BorderFactory.createTitledBorder(panelBorder, "Confidential Text");
    panels[0].setLayout(new GridLayout(1, 1));
    panels[0].setBorder(titleBorder);

    titleBorder = BorderFactory.createTitledBorder(panelBorder, "Numbers");
    panels[1].setLayout(new GridLayout(5, 2));
    panels[1].setBorder(titleBorder);

    titleBorder = BorderFactory.createTitledBorder(panelBorder, "Alphabets");
    panels[2].setLayout(new GridLayout(5, 6));
    panels[2].setBorder(titleBorder);

    titleBorder = BorderFactory.createTitledBorder(panelBorder, "Symbols");
    panels[3].setLayout(new GridLayout(5, 8));
    panels[3].setBorder(titleBorder);

    panels[4].setLayout(new GridLayout(7, 1));
    panels[4].setBorder(panelBorder);
  }

  private void prepareBoard() {

    // text result field
   resultField = new JTextField(1000);
   resultField.setEditable(false);
```

```
Font font = new Font("Sanserif", Font.PLAIN, 24);
resultField.setFont(font);

resultField.setSize(30, 50);
  panels[0].add(resultField);

  // text coverer

  textCover = new JCheckBox("Cover Text");

  textCover.addActionListener(new ActionListener() {

   public void actionPerformed(ActionEvent e) {

     if(textCover.isSelected())
        coverState = true;

     else
        coverState = false;

     coverText();
   }
  });

  // shuffle option
  shuffle = new JCheckBox("Auto shuffle");

  shuffle.addActionListener(new ActionListener() {

   public void actionPerformed(ActionEvent e) {
```

```
    if(shuffle.isSelected())
        autoShuffle = true;

    else
        autoShuffle = false;
 }
});

// prepare buttons
buttons = new JButton[ (KeyMap[0].length()+KeyMap[1].length()+KeyMap[2].length()) ];

int keyNumber = 0, panelNumber = 1;

for(String Key : KeyMap) {

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

      buttons[keyNumber] = new JButton( ""+Key.charAt(i) );

      buttons[keyNumber].addActionListener(new ActionListener(){

         public void actionPerformed(ActionEvent e) {

            if(capsState)
               resultString += e.getActionCommand().trim().toUpperCase();


            else
               resultString += e.getActionCommand().trim().toLowerCase();

            coverText();
```

```
            if(autoShuffle)
                shuffle();
          }
       });
       panels[panelNumber].add(buttons[keyNumber]);

       keyNumber++;
    }
    panelNumber++;
}

// backspace key
backSpace = new JButton("Backspace");
backSpace.addActionListener(new ActionListener(){

    public void actionPerformed(ActionEvent e) {

       if(resultString.length() > 0) {

          resultString = resultString.substring(0, resultString.length()-1);
          coverText();
       }
    }
});

// clear key
clear = new JButton("Clear");
clear.addActionListener(new ActionListener(){

    public void actionPerformed(ActionEvent e) {
```

```
      resultString = "";
      resultField.setText(resultString);
    }
});

// copy key
copy = new JButton("Copy");
copy.addActionListener(new ActionListener(){

   public void actionPerformed(ActionEvent e) {

      /* copying data in another thread to avoid program freeze */
      new Thread( new Runnable() {

         public void run() {

      /* copy data to system clipboard */
       StringSelection selection = new StringSelection(resultString);
       Toolkit.getDefaultToolkit().getSystemClipboard().setContents(selection, null);

       try {
          Thread.sleep(10000); // wait for 10 seconds
       }
       catch(InterruptedException ex) {

       }

       /* remove system clipboard contents */
       selection = new StringSelection(""); // no contents
       Toolkit.getDefaultToolkit().getSystemClipboard().setContents(selection, null);
```

```
        Toolkit.getDefaultToolkit().beep(); /* signal for clipboard cleanup */

          }
       } ).start();

    }
});

//  space key
space = new JButton("Space");
space.addActionListener(new ActionListener(){

    public void actionPerformed(ActionEvent e) {

        resultString += " ";
        coverText();
    }
});

// capslock key
capsLock = new JButton("Capslock");
capsLock.addActionListener(new ActionListener(){

    public void actionPerformed(ActionEvent e) {

        if(!capsState) {
            for(JButton key : buttons) {

                String title = key.getText();

                if(Character.isLowerCase(title.charAt(0)))
```

```
                    key.setText(title.toUpperCase());
                }
              }

            else {
            for(JButton key : buttons) {

                    String title = key.getText();

                    if(Character.isUpperCase(title.charAt(0)))
                      key.setText(title.toLowerCase());
                }
            }
               capsState = !capsState;

        }
      });

      panels[4].add(textCover);
      panels[4].add(shuffle);
      panels[4].add(capsLock);
      panels[4].add(backSpace);
      panels[4].add(space);
      panels[4].add(clear);
      panels[4].add(copy);


}

/* hides the text based on cover state */
private void coverText(){
```

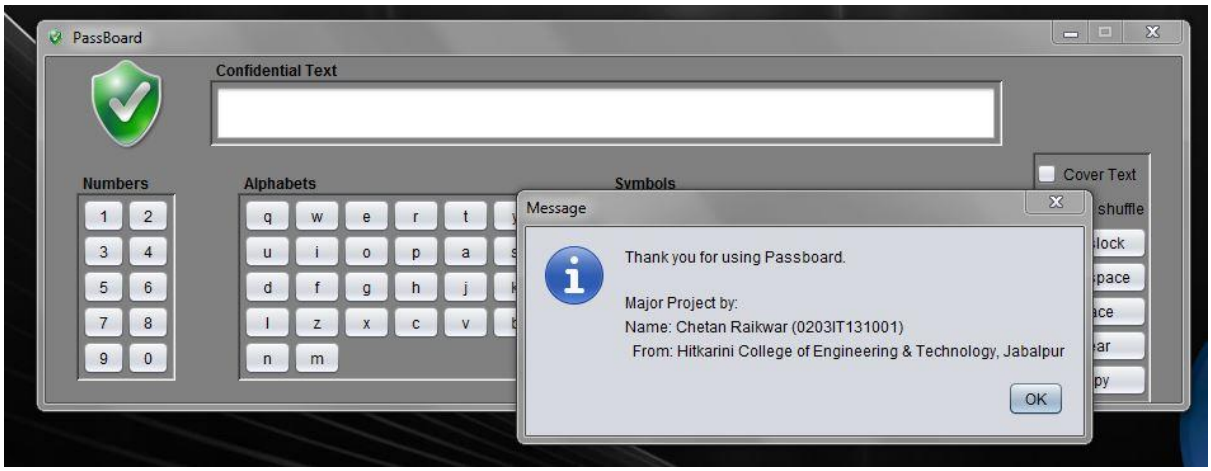
```
    if(coverState){

        String starText = "";
        for(int i=0; i<resultString.length(); i++)
            starText += "*";

        resultField.setText(starText);
    }

    else
        resultField.setText(resultString);
}

/* shuffles the keys in groups */
private void shuffle(){

    ArrayList<Character> List = new ArrayList<>();

    for(int mainCount = 0; mainCount < KeyMap.length; mainCount++ ) {

        for(int i=0; i< KeyMap[mainCount].length(); i++)
            List.add(KeyMap[mainCount].charAt(i));

        Collections.shuffle(List);

       String newMap = "";

        for(Character c : List)
            newMap += c;
```

```
            KeyMap[mainCount] = newMap;
            List.clear();
        }

        int keyNumber = 0;

        for(String Key : KeyMap) {
            for(int i=0; i<Key.length(); i++)
                buttons[keyNumber++].setText( ""+Key.charAt(i) );

        }
    }
}

class PassBoard extends JFrame {

    public PassBoard(String title) {

        addWindowListener(new WindowAdapter(){

            public void windowClosing(WindowEvent e) {

                JOptionPane.showMessageDialog(null, "Thank you for using Passboard. \n\nMajor Project by:\n"
                        + "Name: Chetan Raikwar (0203IT131001)\n   From: Hitkarini College of Engineering & Technology, Jabalpur");
            }
        });

        add(new KeyPad());
        setIconImage(new ImageIcon("F:\\green.png").getImage());
        setTitle(title);
```

```
        setSize(900, 300);
        setLocationByPlatform(true);
        setResizable(false);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }
}

class Solution {

    public static void main(String[] args) {

        EventQueue.invokeLater(new Runnable(){

            public void run() {

                EventQueue.invokeLater(new Runnable() {

                    public void run() {

                        new PassBoard("PassBoard");
                    }
                });

            }
        });

    }
}
```

Program Images


PassBoard
Confidential Text
chetan
Numbers
Alphabets
Symbols
Cover Text
Auto shuffle
Capslock
Backspace
Space
Clear
Copy


PassBoard
Confidential Text
Numbers
Alphabets
Symbols
Cover Text
Message
Thank you for using Passboard.
Major Project by:
Name: Chetan Raikwar (0203IT131001)
From: Hitkarini College of Engineering & Technology, Jabalpur
OK

References

- Icon from free icons packages (google images).
- Paintbrush for minor editing.
- Core Java Volume 1, for GUI designing.
- Netbeans for development.
- Articles about Keylogging types and techniques by kaspersky:

  https://www.kaspersky.co.in/resource-center/definitions/keylogger

  https://www.kaspersky.com/blog/keylogger/1573/

  (These articles above lead to development of my project)