**ORIGIN LAB   
(DIAGNOSTIC LABORATORY SYSTEM)**

A Project Report Submitted

in Partial Fulfillment of the Requirement

for the Degree in Master of Computer Applications

By

Kyrshanlang R. Dkhar

University Roll No: SAC MC 15/01

Registration No: 2945 of 2012-2013



Department of Computer Science

St. Anthony’s’s College, Shillong

**CERTIFICATE**

This is to certify that the Project work titled ‘***ORIGIN LAB   
(DIAGNOSTIC LABORATORY SYSTEM)’*** is a bonafide work done by ***Kyrshanlang R. Dkhar, University Roll No : SAC MC 15/01,*** under my guidance during the 5th Semester of the course.

**Medari Tham**

**Internal Guide**

Project seminar was held on ………………. at St. Anthony’s College, Shillong.

**Dr. A. Das**

**Head of the Department** **External Examiner**

**Acknowledgement**

Working on this mini-project was a mammoth task for which a lot of help was required from many people. Fortunately, I have had the fine support of my family, friends and teachers of the department of computer science of St Anthnoy’s College.

My special thanks go to my guide Miss Medari Tham, for without her support I would not have completed my project, and also I would like to thank Sir Anjan Das, Head of Department, Computer Science and all the teachers of the Department for their suggestions and feedback which help me developed my project in a better way.

**Table of Contents**

Contents

[**Introduction** 5](#_Toc493891078)

[Synopsis 6](#_Toc493891079)

[System Study 7](#_Toc493891080)

[User Requirements 10](#_Toc493891081)

[Data Flow Diagram 11](#_Toc493891082)

# **Introduction**

**Diagnostic Laboratory** isconcern with performing test on samples to diagnose a person’s wellbeing, particularly blood, urine, sputum, fungal spores and many more. The Test results are then processed and reports are generated.

This project is a client based live mini project, for Origin Diagnostic Laboratory, Shillong. Located in Jaiaw Langsning, the Origin Diagnostic Center was started in 2017. With the objective of providing quality and timely service to its clients, the Origin Diagnostic Center has incorporated and invested in some of the latest technological equipment to meet the needs of the time. With a team of young and committed professionals at the helm of management, the future of the center is promising.

The Organization currently has 12 staffs including a Medical Consultant, Lab Technician, Data Entry Operator and more.

The Laboratory has several Departments and each Department has a set of Tests that are performed.

* **Departments & Tests performed:**
  + **Biochemistry** – Blood Sugar, Thyroid Test, Liver Function Test, Kidney Function Test, Diabetic Profile.
  + **Microbiology** – Blood Culture, Stool Culture, Urine Culture, Respiratory Culture.
  + **Serology** – Widal Test, Dengue, Malaria, ABO Group
  + **Hematology** – WBC, LYM%, MID%, GRAN%, RBC, HGB.

# Synopsis

**Project Description**

The proposed System is based on the requirements of the Management Team from Origin Diagnostic Laboratory, an application is to be developed that will help in automating the process of generating reports for it’s patients, standardizing the Laboratory Normal Range Values for different Test and increase the work efficiency of the Laboratory.

The Application will be a web based application and will include the following features:

1. Allow the Administrator to Manage Departments and Tests.
2. Provide the Administrators to create and manage Patient’s Details, Patient’s Test Reports and Bills.
3. Allow the Administrator to View Monthly Reports on the Total number of Tests done, total number of Positive test cases and total number of bills settled.
4. Standardized the Normal Range of Each Test for The Region and Laboratory.
5. Provide the Average Result Range, Highest Range and Lowest Range of all Tests.

**Language used**

* PHP, HTML, Javascript, CSS.

**Database**

* MySQL

# System Study

**Existing System**

At Origin Diagnostic Center, Samples are collected from the Collection Center.

Tests are performed in the Laboratory and the results of these tests are note down by the Lab Technician and Staff of the Laboratory.

The Results are confirmed by the medical consultant, entered into a file by a Data Entry Operator and Reports are then generated manually and printed, each report must be manually generated for different types of Tests and the task is an exhaustive time consuming task.

Samples collected include Blood Samples, Urine Samples, Sputum, Stool samples, Throat swabs, Fungal spores…

Test are done either manually, on the spot using Portable Devices or automatically with the help of automated machines (e.g Biochemistry Analyzer, Electrolyte Automated Analyzer, Urine Analyzer, Immunoflorescence Analyzer).

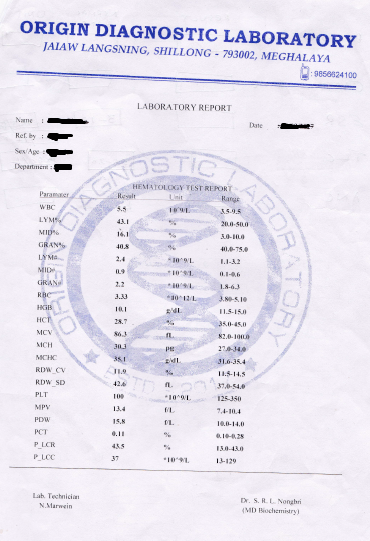


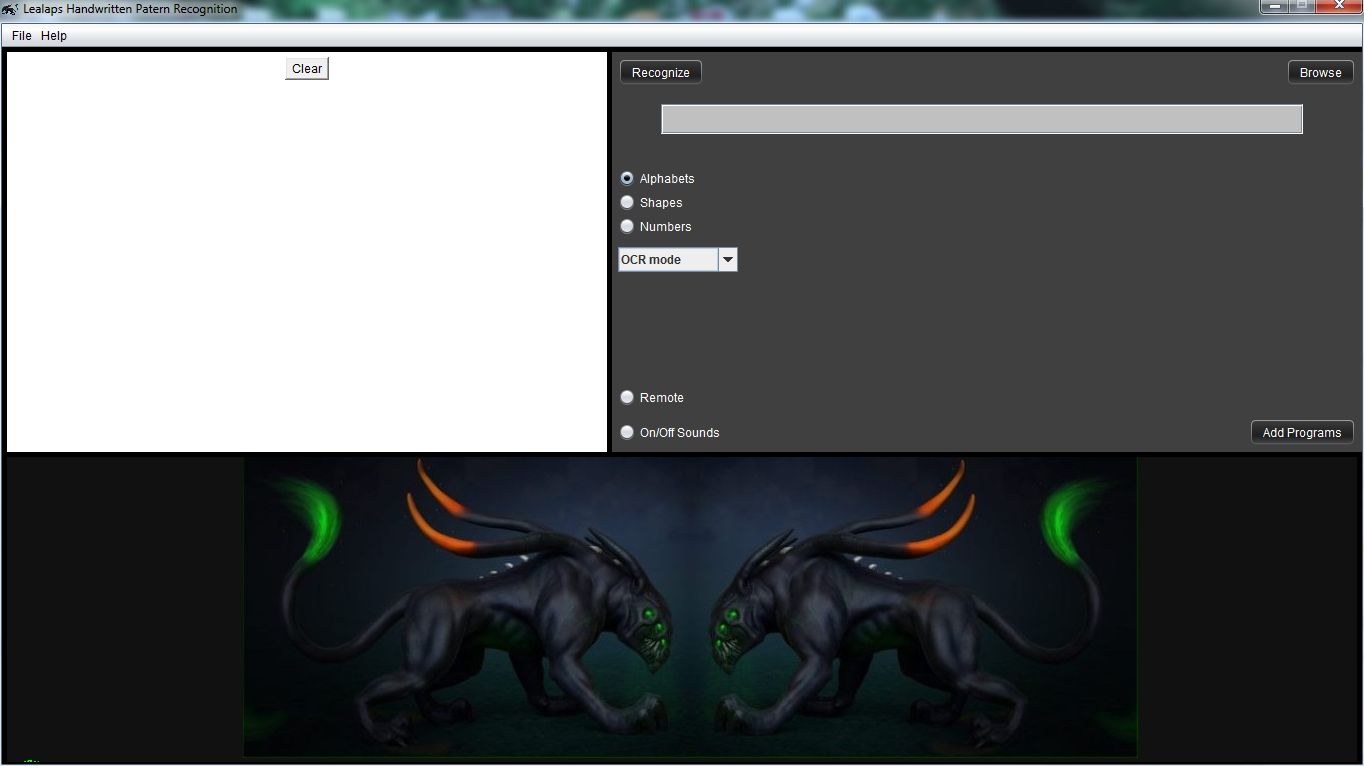
Figure Sample Report

**Drawbacks of the Exsiting System**

1. Report and Bill generation are done manually.
2. Keeping records of patients, tests and reports are difficult.
3. Computing Average/Lowest/Highest Range of a test is exhaustive and time consuming.
4. Computing Standardized Normal Range for each test is difficult.

**Proposed System**

The Proposed System will be an application that will help in automating the process of generating reports for its patients, standardizing the Laboratory Normal Range Values for different Test and increase the work efficiency of the Laboratory.

****

Pic: Proposed System Screen Shot

**Features**

* OCR Mode
  + Shape Recognition
  + Character Recognition
* Application Mode
  + Opening app with pattern drawn in the white box
* Interactive Mode
  + Answering series of questions displayed on the screen via writing in white box
  + Allowing users to set Questions and Answers.
* Audio Output.
* White Box input.
* Image Input ( jpg and png image ).
* Remote Drawing via an android device.

**Algorithm used**

Two popularly known algorithms are used:

1. Chain Code algorithm:

Chain codes are used to represent a boundary by a connected sequence of straight line segments of specified length and direction. The direction of each segment is coded by using a numbering scheme from 0 to 7 in 8 directional and 0 to 3 in a 4 directional chain code. This algorithm have been specially recreated for this project for simplicity from Freemans chain code algorithm in that the traversing from one pixel to another is completely different from the original algorithm. This algorithm will return a vector of size 8.

1. OCRchie algorithm:

### 

# User Requirements

**Software Requirements**

1. Web Browser, WampServer.
2. Window 7 and later.

**Hardware Requirements**

1. Intel Dual Core processor 2.4 GHZ.

2. RAM 1 GB.

3. HDD 1 GB.

**System Flowchart**

Input

Output sound

Is the input valid?

Yes

No

Output

Text

Computing

Algorithms

Is sound

on?

Output sound

Yes

No

Is the system in application mode?

Output application

Yes

No

Is the system in interactive mode?

Is the Ans correct?

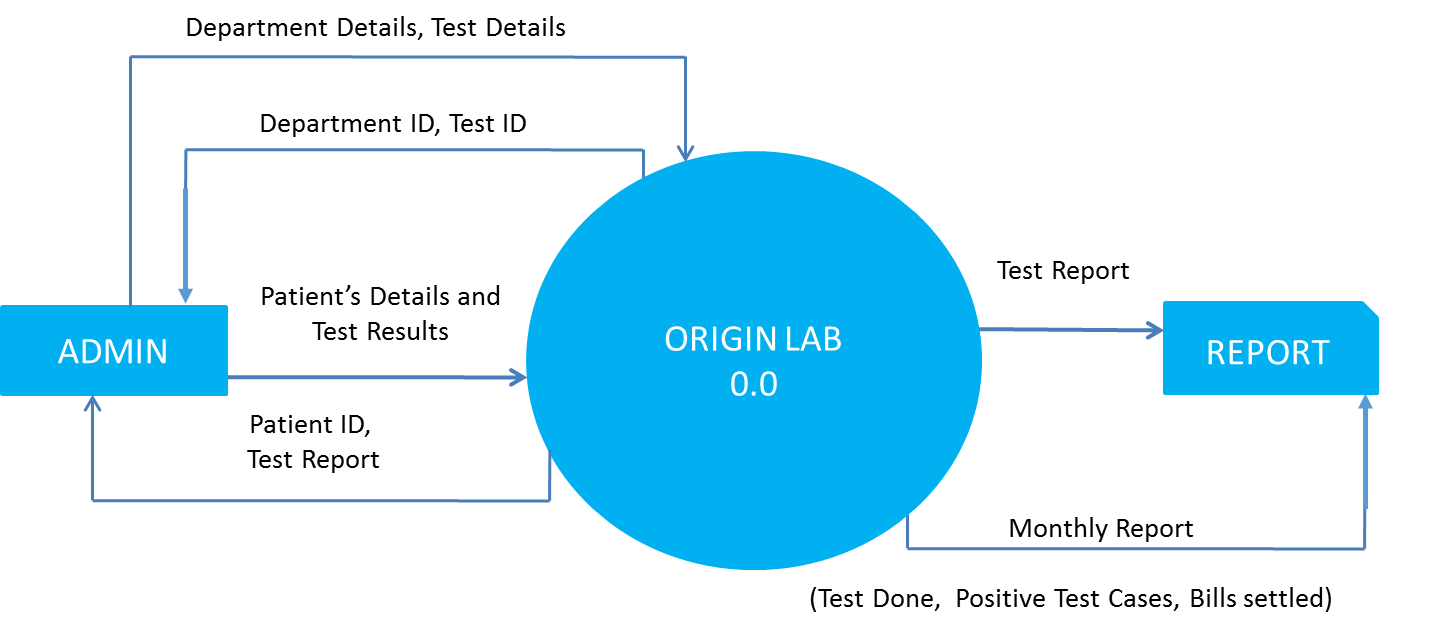
Yes Yes

Output sound and scores

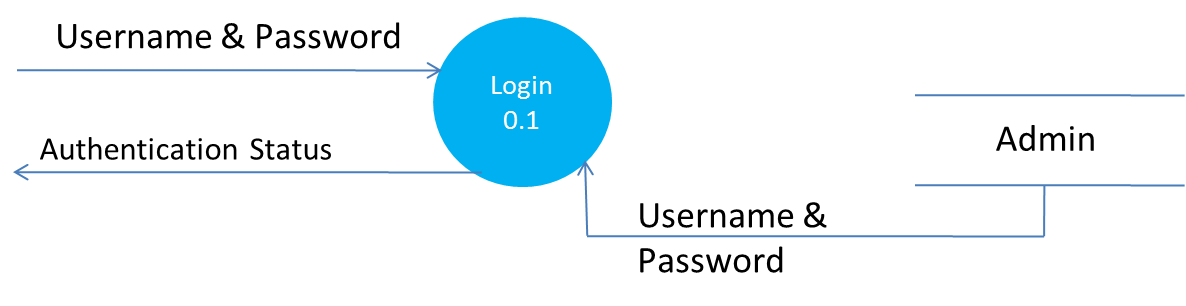
No No

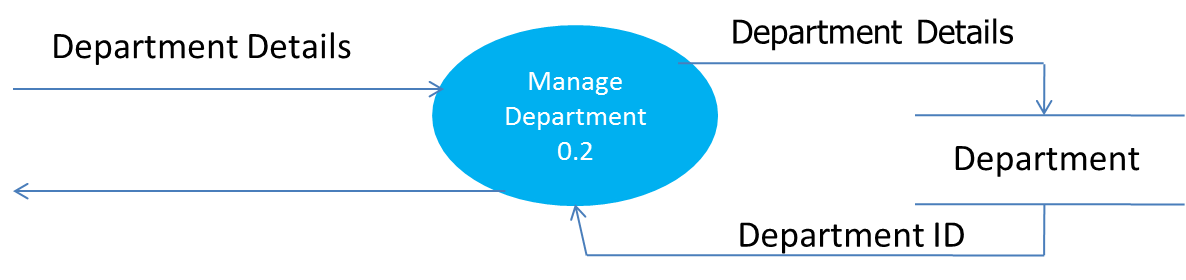
# Data Flow Diagram

**CONTEXT LEVEL DATA FLOW DIAGRAM**

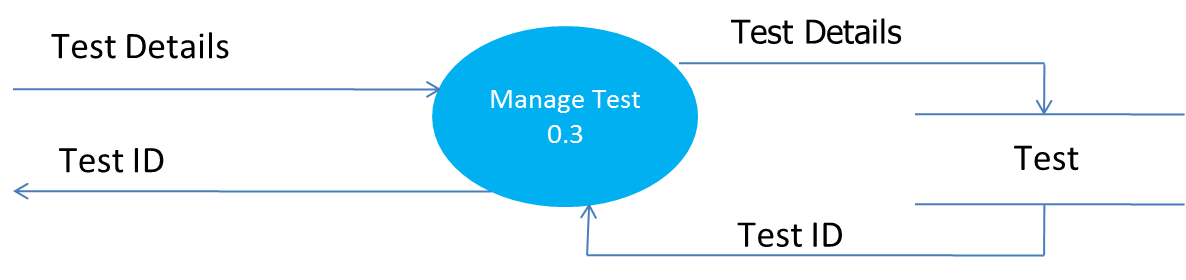


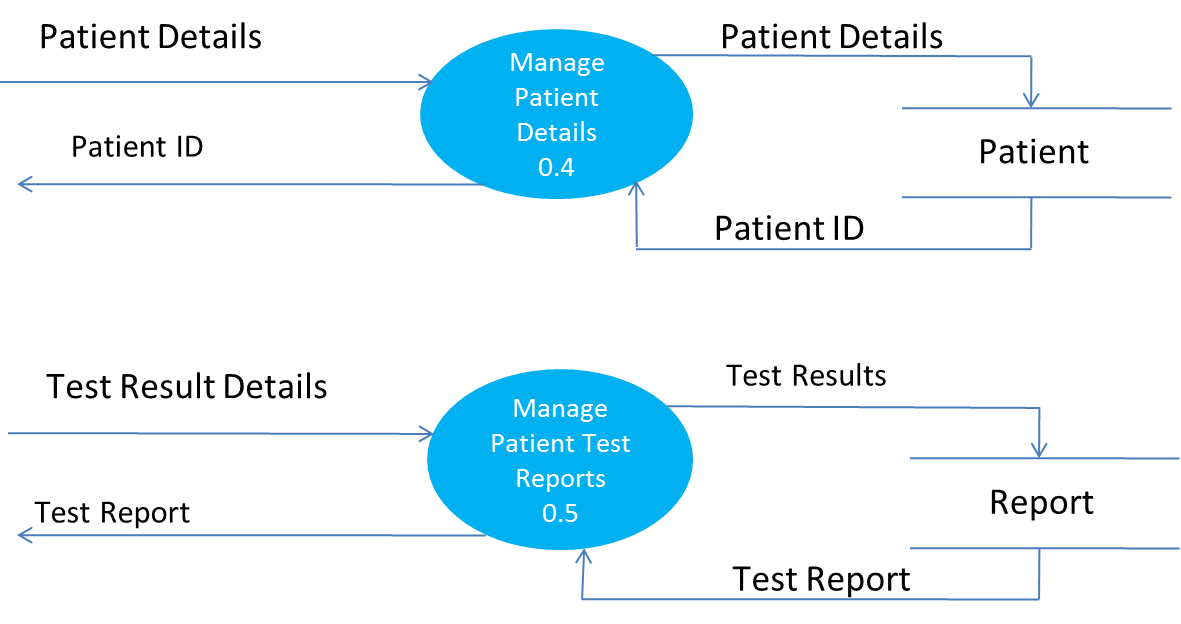
**LEVEL 1 DATA FLOW DIAGRAM**

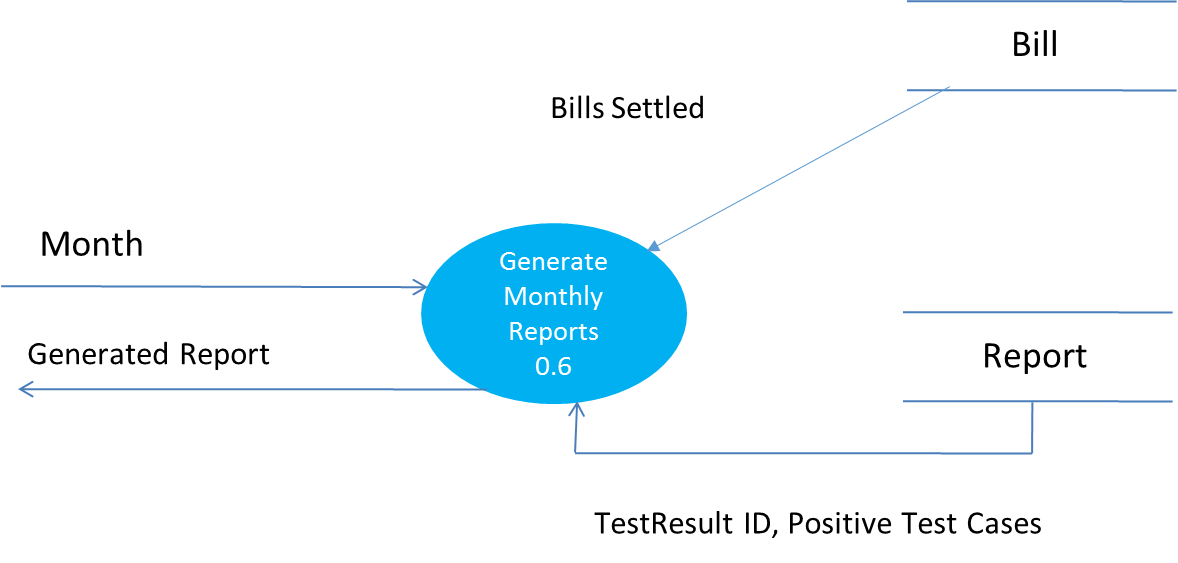


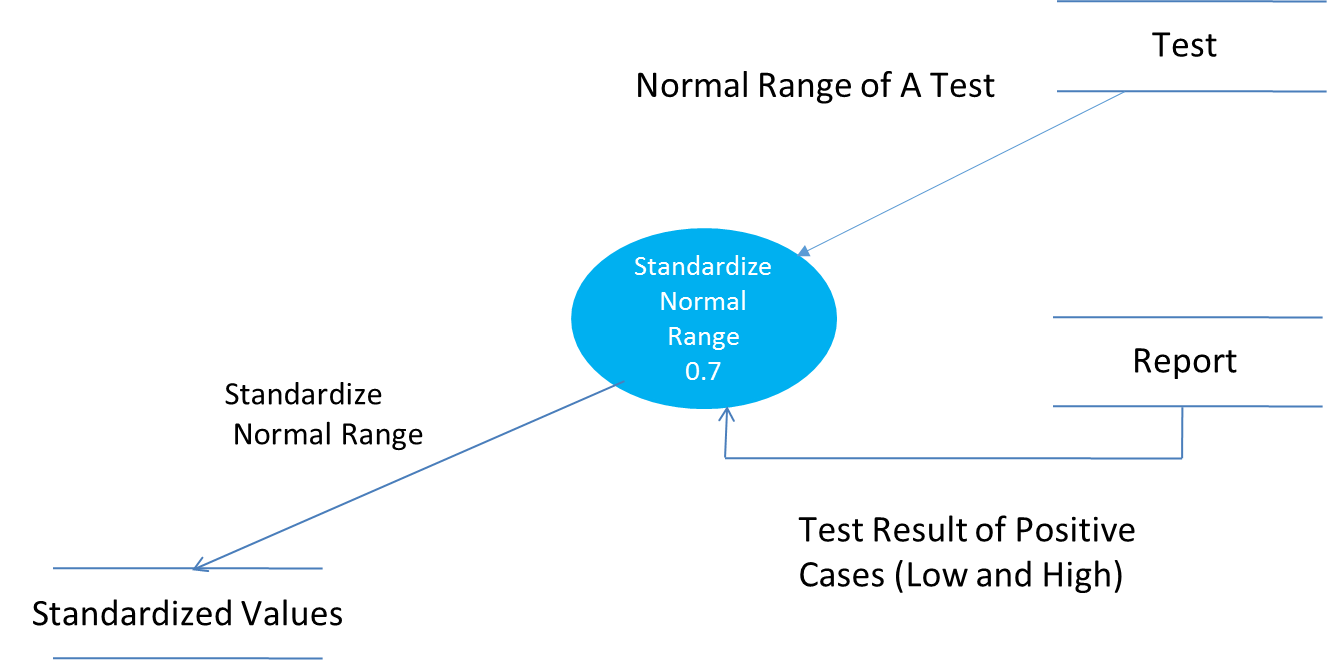


Department ID

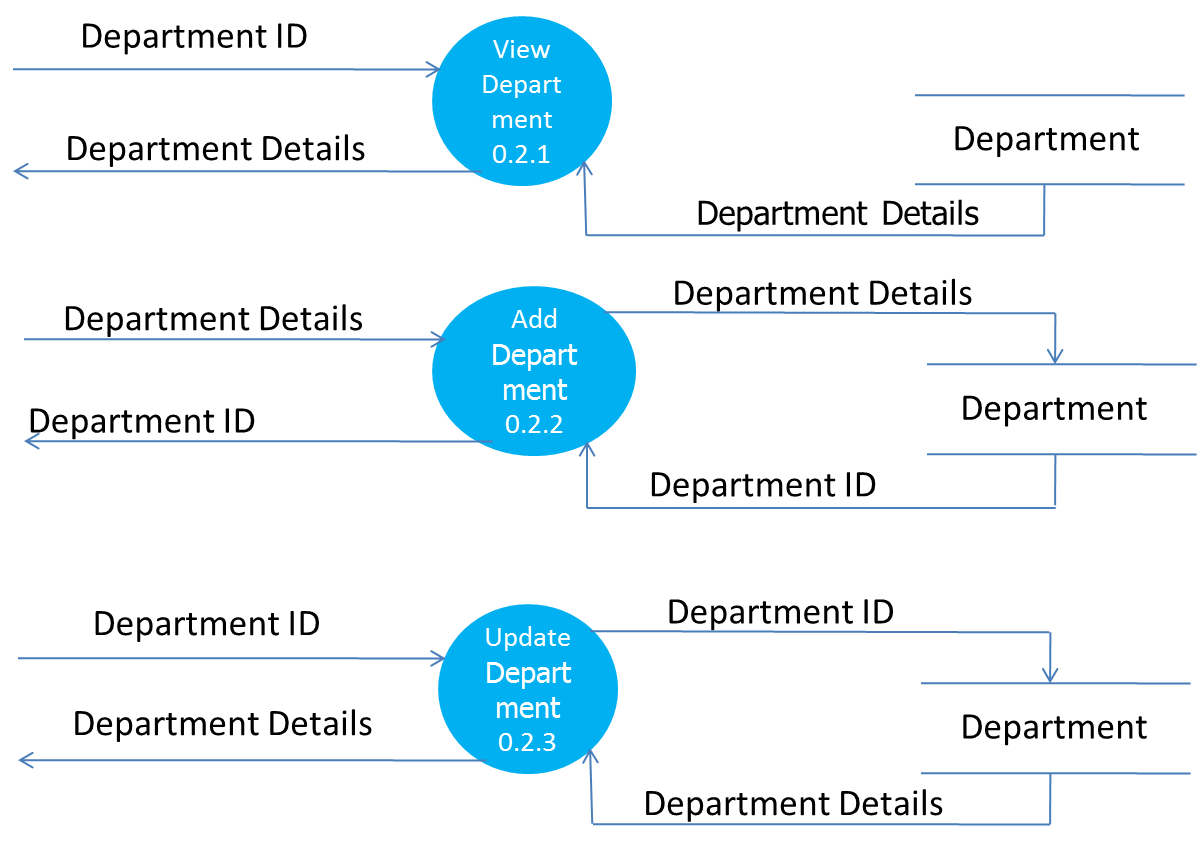


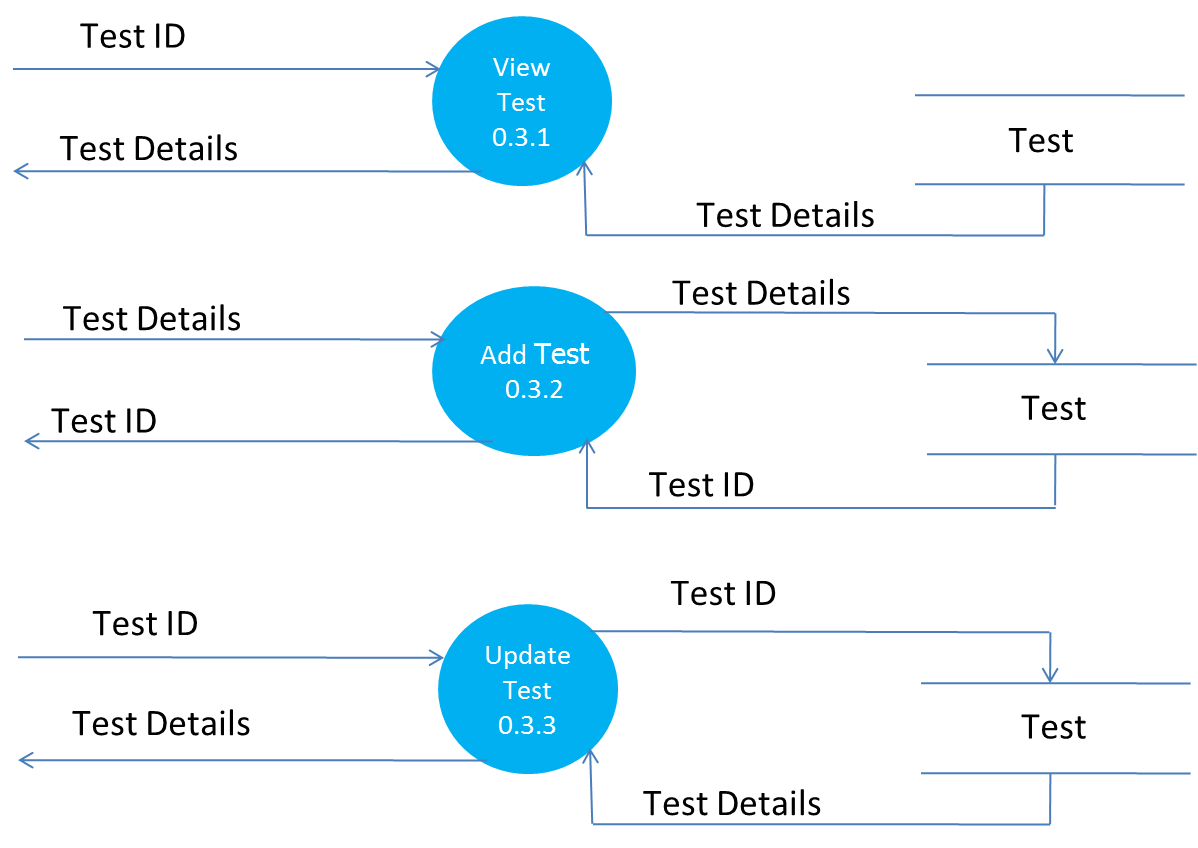


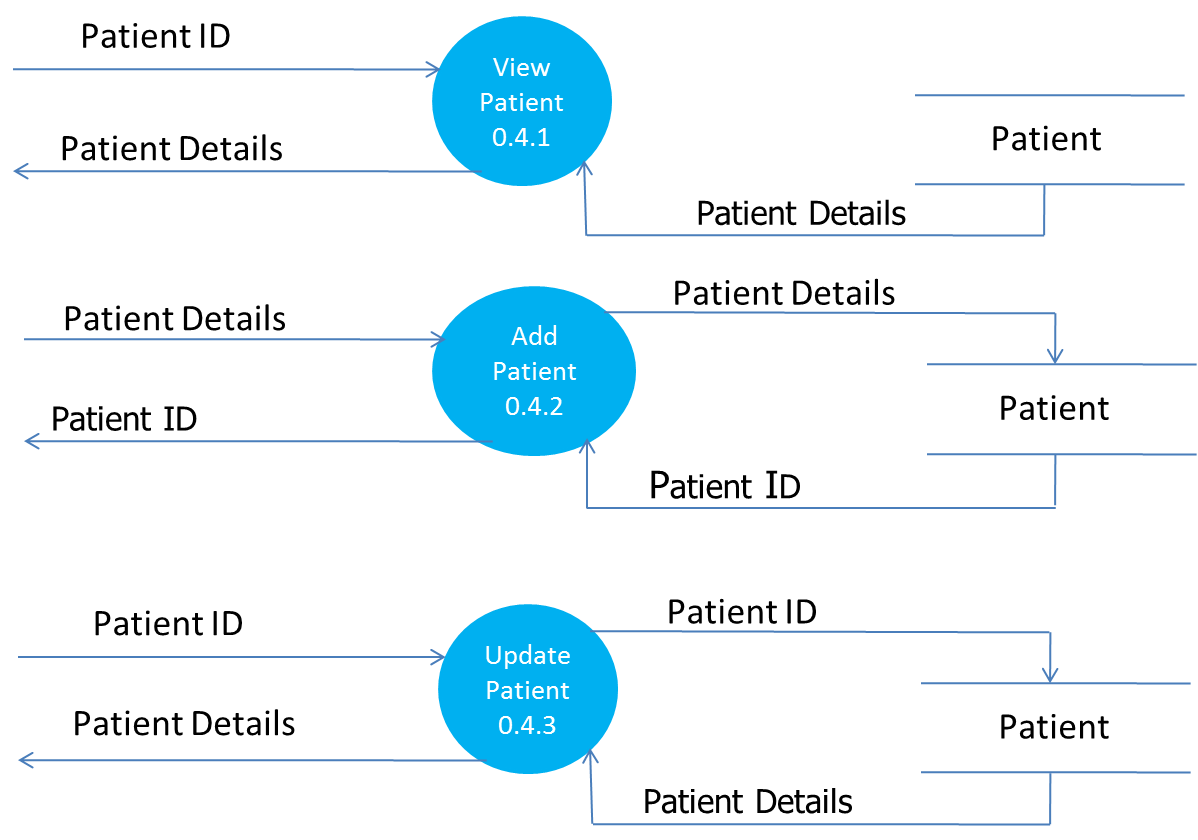


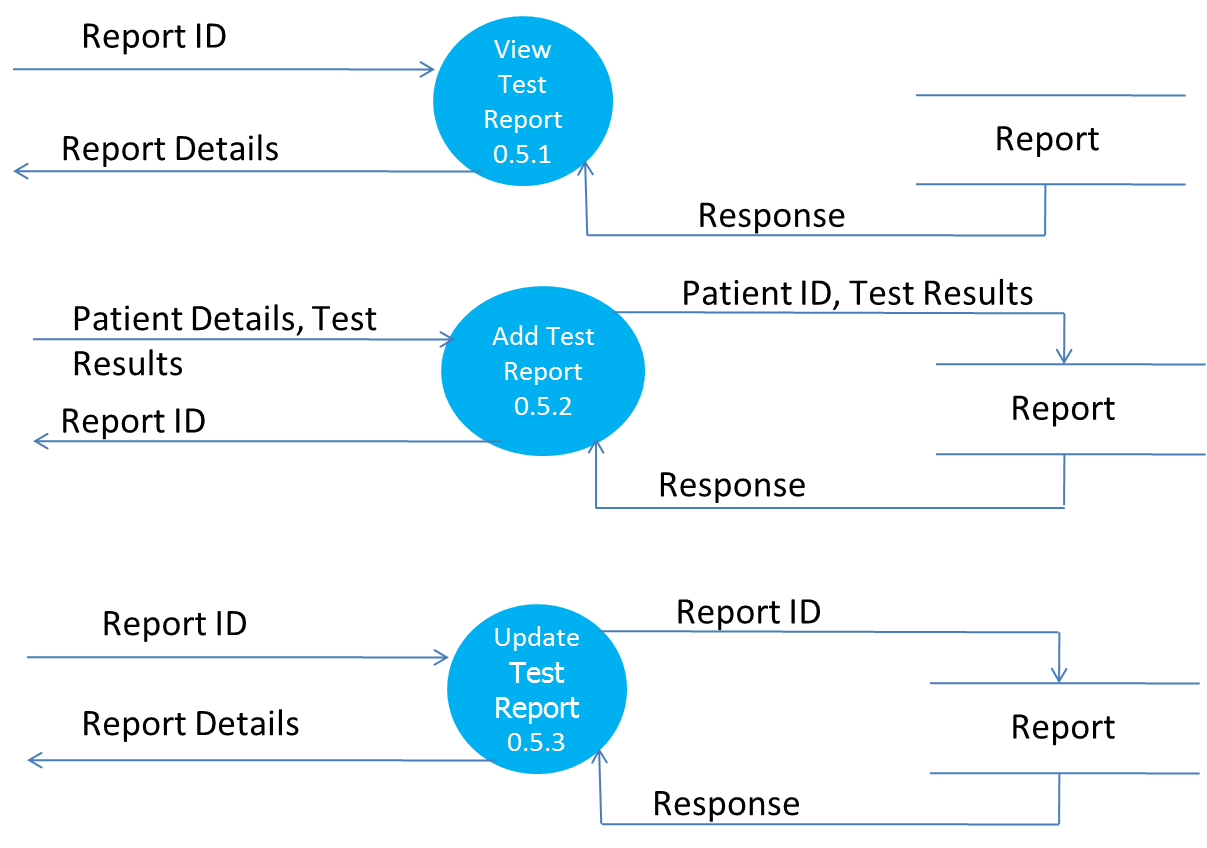


**LEVEL 2 DATA FLOW DIAGRAM**

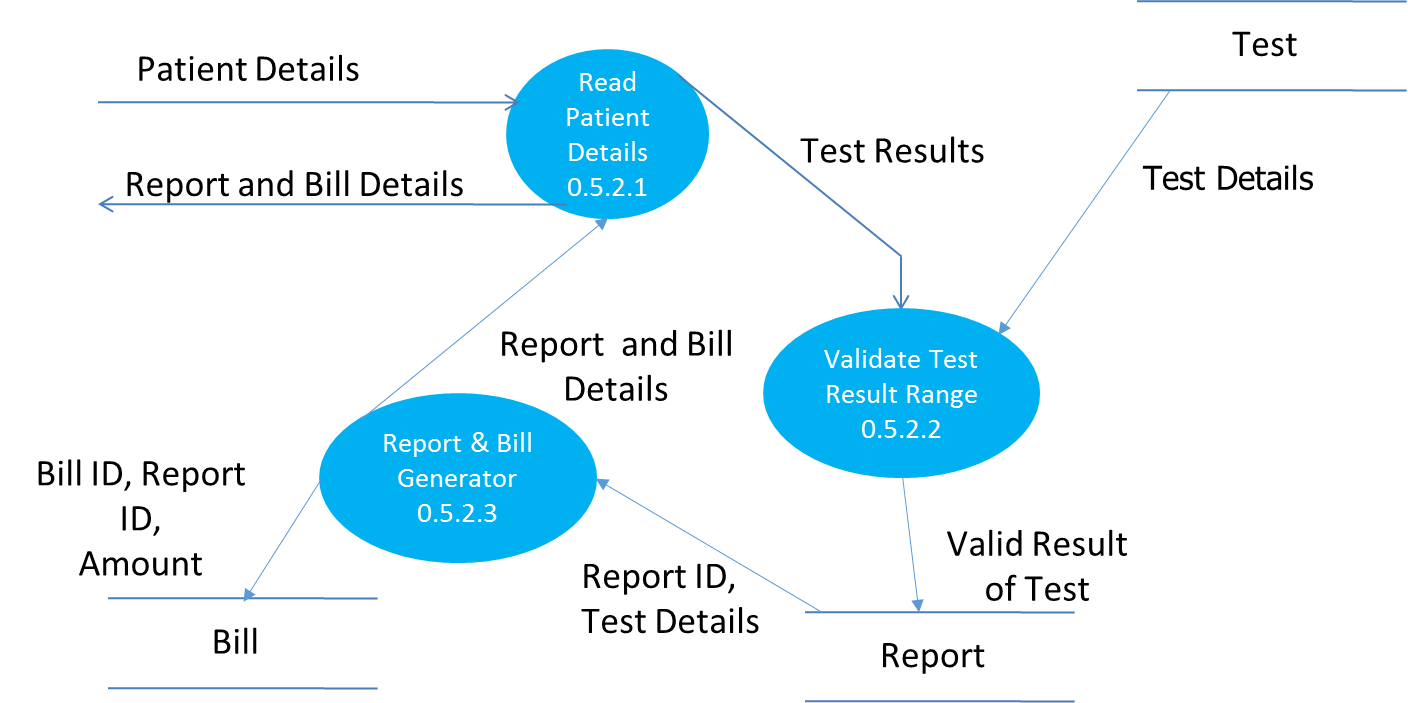




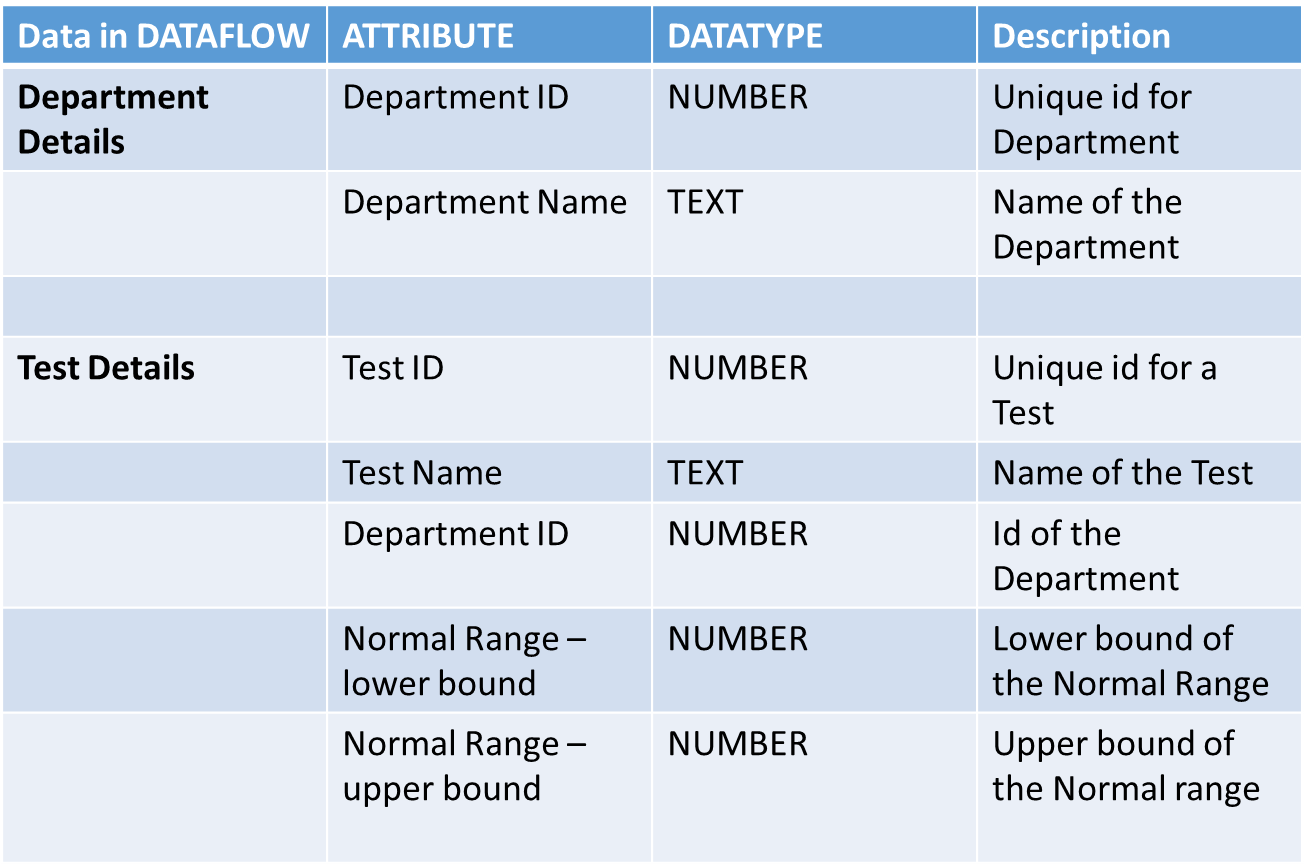
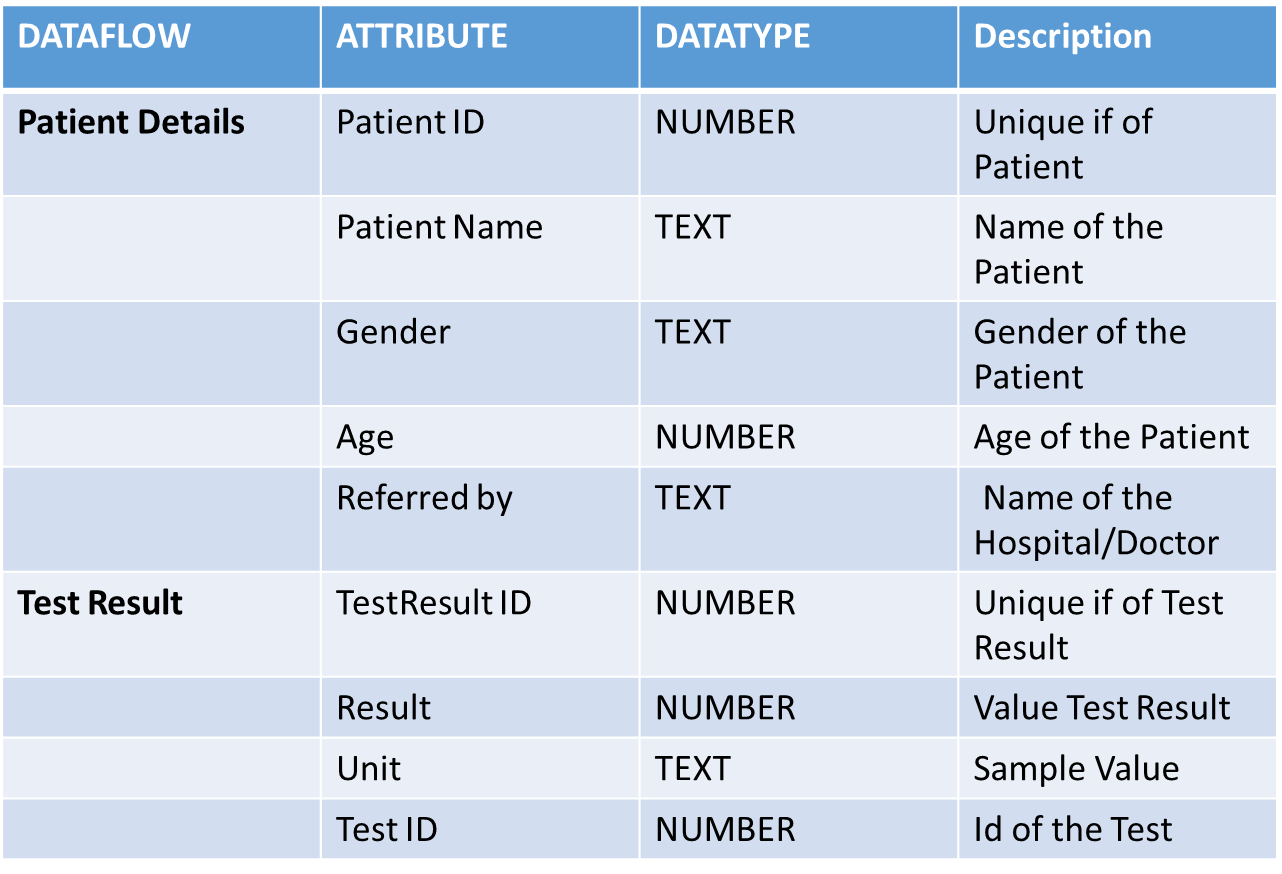


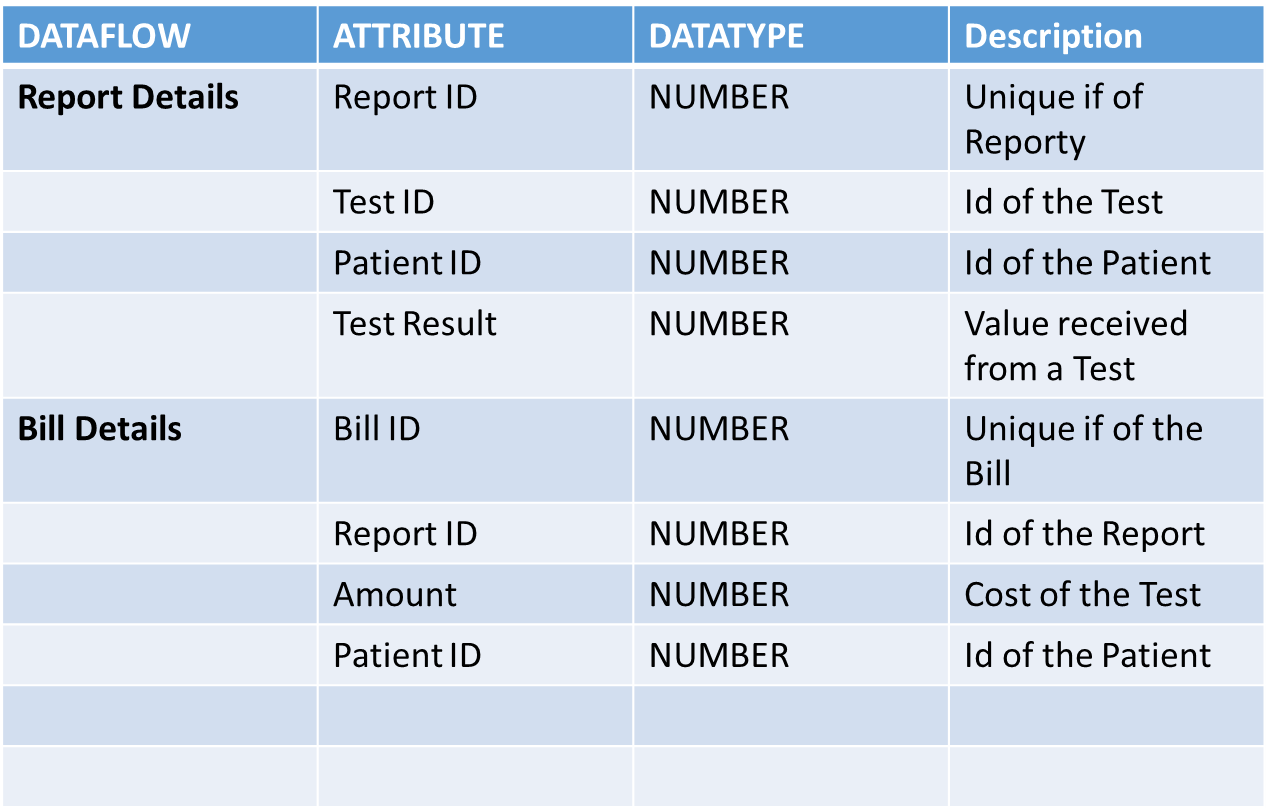


**LEVEL 3 DATA FLOW DIAGRAM**



**DATA DICTIONARY**



****

**Module Design**

**System Modules**

1) Main:-

This is the main module in the project. The painting feature is included in this module. This module directly communicates with all the other modules except with FileData where the communication is indirect.

2) MM:-

This module displays a startup screen and checks all the files required by the system using a batch file. It will display an error message if some files is missing and terminate the program.

3) OCRChie\_ChainCode:-

This Module deals with computes the chain code and OCRchie algorithm of an object and send the Vector of size 33 obtain to the FileData module and the result obtain from FileData is send to the main.

4) NewFileData:-

This module deals with the training sets. The training sets mathematical value is stored in a text file. This module compare the vector obtain from OCRChie\_ChainCode module and the value from the training sets using the Recognition algorithm. The result obtain is send back to OCRChie\_ChainCode.

5) PaintApplications:-

The painting feature is handled by this module where it extends a canvas on which the user draws a pattern.

6) Exe:-

This module is call after OCRChie\_ChainCode. The main work here is to call the corresponding app if the user request for an app whose key have been added in the system.

7) Applications:-

This module allows a user to assign key for a particular app. It also allows a user to update, delete and create new entries of application.

8) Sound:-

After the result is produce it is send to this module where the result is written to a txt file and the test.vbs is called where it reads the txt and produce the sounds by creating a voice object of spVoice.

spVoice is a Microsoft Speech API which brings the text-to-speech (TTS) engine capabilities to applications.

9) Questions:-

This module allows a user to set questions and answers of the interactive mode.

10) Remote1:-

This modules handles the data send by the android device and use Robot class to automatically move the cursor on the screen.

**Android Modules**

1. Drawss:-

The drawing within the android is been done in this module which draws on to a canvas.

1. SendMessage:-

This module sends the coordinates to the system program using UDP protocol.

1. MainActivity:-

This module is the main module where the entire component is added. The Drawss class is added in this module and when a user interact with the canvas the coordinates is sent to the SendMessage module where it in turn sent it over a network to a System program.

**Communications between modules**

MainActivity

SendMessage

Drawss

[ datagram packets ]

Remote1

Applications

PaintApplications

NewFileData

OCRChie\_ChainCode

MM

Exe

Questions

Main

Sounds

**Future Enhancements**

**Conclusion**

**Bibliography**

**Source Code**

**Main:-**

import Paint.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.awt.image.BufferedImage;

import java.io.\*;

import javax.imageio.ImageIO;

import javax.swing.\*;

public class Main extends JFrame implements ActionListener,ItemListener{

Interactive i = new Interactive();

PaintApplications pa = new PaintApplications();

Filechs ff = new Filechs();

public static JComboBox jComboBox2 = new JComboBox();

private JPanel jPanel1 = new JPanel();

private JPanel jPanel2 = new JPanel();

private JButton jButton1 = new JButton("Recognize");

private JButton jButton2 = new JButton("Browse");

private JButton jButton3 = new JButton("Add Programs");

public static JTextField jTextField = new JTextField();

private JRadioButton jRadioButton1 = new JRadioButton("Alphabets");

private JRadioButton jRadioButton2 = new JRadioButton("Shapes");

private JRadioButton jRadioButton3 = new JRadioButton("Numbers");

private JRadioButton jRadioButton4 = new JRadioButton("On/Off Sounds");

private JRadioButton jRadioButton5 = new JRadioButton("Remote");

private ButtonGroup group=new ButtonGroup();

Runtime rt;

private int comboBox=1,sounds=0,remote=0;

String c="";

public int select=1;

Remote1 r;

BufferedImage image = null;

Main()

{

setLayout(new FlowLayout(FlowLayout.LEFT));

setVisible(true);

setSize(1366,768);

setResizable(false);

try

{

BufferedImage icon = ImageIO.read(new File("D:/project/images/icon2.png"));

setIconImage(icon);

}

catch (IOException eew){}

setTitle("Lealaps Handwritten Patern Recognition");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

jButton1.setCursor(new Cursor(Cursor.HAND\_CURSOR));

jButton1.setFocusPainted(false);

jButton1.setBackground(Color.BLACK);

jButton1.setForeground(Color.WHITE);

jButton1.addActionListener(this);

jButton2.setCursor(new Cursor(Cursor.HAND\_CURSOR));

jButton2.setFocusPainted(false);

jButton2.setBackground(Color.BLACK);

jButton2.setForeground(Color.WHITE);

jButton2.addActionListener(this);

jButton2.setToolTipText("Input an image for character recognition");

jButton3.setCursor(new Cursor(Cursor.HAND\_CURSOR));

jButton3.setFocusPainted(false);

jButton3.setBackground(Color.BLACK);

jButton3.setForeground(Color.WHITE);

jButton3.addActionListener(this);

jButton3.setToolTipText("Add a new program to open with paint");

jRadioButton1.setCursor(new Cursor(Cursor.HAND\_CURSOR));

jRadioButton1.setForeground(Color.WHITE);

jRadioButton2.setCursor(new Cursor(Cursor.HAND\_CURSOR));

jRadioButton2.setForeground(Color.WHITE);

jRadioButton3.setCursor(new Cursor(Cursor.HAND\_CURSOR));

jRadioButton3.setForeground(Color.WHITE);

jRadioButton4.setCursor(new Cursor(Cursor.HAND\_CURSOR));

jRadioButton4.setForeground(Color.WHITE);

jRadioButton5.setCursor(new Cursor(Cursor.HAND\_CURSOR));

jRadioButton5.setForeground(Color.WHITE);

group.add(jRadioButton1);

group.add(jRadioButton2);

group.add(jRadioButton3);

jRadioButton1.setSelected(true);

jRadioButton1.addActionListener(this);

jRadioButton2.addActionListener(this);

jRadioButton3.addActionListener(this);

jRadioButton4.addActionListener(this);

jRadioButton5.addActionListener(this);

jComboBox2.setCursor(new Cursor(Cursor.HAND\_CURSOR));

jTextField.setBackground(Color.LIGHT\_GRAY);

jTextField.setFont(new Font("Lucida Console",Font.BOLD,20));

Dimension d=new Dimension(600,400);

pa.setPreferredSize(d);

add(pa);

Dimension d1=new Dimension(750,400);

jPanel1.setPreferredSize(d1);

jPanel1.setBackground(Color.DARK\_GRAY);

getContentPane().setBackground(Color.black);

add(jPanel1);

Dimension d2=new Dimension(1350,350);

jPanel2.setPreferredSize(d2);

jPanel2.setLayout(new BorderLayout());

jPanel2.add(i);

add(jPanel2);

add( jComboBox2);

jComboBox2.setModel(new DefaultComboBoxModel(new String[] { "OCR mode","Application mode","Interactive mode" }));

design();

JMenuBar mb=new JMenuBar();

setJMenuBar(mb);

JMenu file=new JMenu("File");

file.setForeground(Color.BLACK);

JMenuItem item1,item2;

file.add(item1=new JMenuItem("Set Password"));

file.add(item2=new JMenuItem("Exit"));

mb.add(file);

JMenu help=new JMenu("Help");

help.setForeground(Color.BLACK);

JMenuItem item3,item4;

help.add(item3=new JMenuItem("Help Contents"));

help.add(item4=new JMenuItem("About"));

mb.add(help);

jComboBox2.addItemListener(this);

item1.addActionListener(this);

item3.addActionListener(this);

item4.addActionListener(this);

pa.clear=1;

pa.repaint();

}

public final void design()

{

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addGap(0, 0, Short.MAX\_VALUE)

.addComponent(jButton3))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, jPanel1Layout.createSequentialGroup()

.addComponent(jButton1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton2))

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jRadioButton1, javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jRadioButton2, javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jRadioButton3, javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jComboBox2, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED\_SIZE, 120, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jRadioButton4, javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jRadioButton5, javax.swing.GroupLayout.Alignment.LEADING))

.addGap(0, 0, Short.MAX\_VALUE))))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(49, 49, 49)

.addComponent(jTextField, javax.swing.GroupLayout.PREFERRED\_SIZE, 642, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 33, Short.MAX\_VALUE)))

.addContainerGap())

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButton1)

.addComponent(jButton2))

.addGap(18, 18, 18)

.addComponent(jTextField, javax.swing.GroupLayout.PREFERRED\_SIZE, 30, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(35, 35, 35)

.addComponent(jRadioButton1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jRadioButton2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jRadioButton3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jComboBox2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 143, Short.MAX\_VALUE)

.addComponent(jRadioButton5)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButton3)

.addComponent(jRadioButton4))

.addContainerGap())

);

}

public static void main(String args[])

{

MM m = new MM();

int noooop=m.check();

System.out.println("exist: "+noooop);

try{ Thread.sleep(3000); }catch(Exception e){}

if(noooop>=1)

{

JOptionPane.showMessageDialog(null,m.msg,"!!! "+noooop+" ERROR Found !!!",JOptionPane.ERROR\_MESSAGE);

System.exit( 0 );

}

else

{

//\*\* Nimbus \*\*//

m.dispose();

try {

for (UIManager.LookAndFeelInfo info : UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName()))

{

UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException | InstantiationException | IllegalAccessException | UnsupportedLookAndFeelException ex) {}

//\*\* Nimbus \*\*//

Main f=new Main();

}

}

@Override public void itemStateChanged(ItemEvent ie)

{

if(jComboBox2.getSelectedIndex()==0)

{

comboBox=1;

}

if(jComboBox2.getSelectedIndex()==1)

{

comboBox=2;

}

if(jComboBox2.getSelectedIndex()==2)

{

comboBox=3;

}

}

@Override public void actionPerformed(ActionEvent e)

{

String g=e.getActionCommand();

if(e.getSource()==jRadioButton1)

{

select=1;

}

if(e.getSource()==jRadioButton2)

{

select=2;

}

if(e.getSource()==jRadioButton3)

{

select=3;

}

if(e.getSource()==jRadioButton4)

{

if(sounds==0)

sounds=1;

else

sounds=0;

}

if(e.getSource()==jRadioButton5)

{

if(remote==0)

{

remote=1;

try{r = new Remote1();}catch(Exception err){}

}

else

{

r.close();

remote=0;

}

}

if(e.getSource()==jButton1)

{

image=readImage2("D:/Project/paint/100.jpg");

OCRChie\_ChainCode pp=new OCRChie\_ChainCode();

try{

c=pp.CCOCR(image,select).toString();

}catch(Exception exx){}

jTextField.setText(c);

if(sounds==1)

{

Sound.play(c);

}

if(comboBox==2)

{

Exe ex=new Exe();

int jj=ex.exe(c);

if(jj==0)

{

Programme p = new Programme(c);

}

}

if(comboBox==3)

i.rec(c);

}

if("Browse".equals(g))

{

String f=ff.file();

if(!f.equalsIgnoreCase("nullnull"))

{

image=readImage2(f);

OCRChie\_ChainCode pp=new OCRChie\_ChainCode();

try{

c=pp.CCOCR(image,4).toString();

}catch(Exception ee){}

jTextField.setText(c);

if(sounds==1)

{

Sound.play(c);

}

if(comboBox==2)

{

Exe ex=new Exe();

int jj=ex.exe(c);

if(jj==0)

{

Programme p = new Programme(c);

}

}

if(comboBox==3)

i.rec(c);

}

else

if( (f.equalsIgnoreCase("nullnull"))||(f.equalsIgnoreCase("")) )

JOptionPane.showMessageDialog(rootPane,"No image selected");

}

if("Set Password".equals(g))

{

Pass p = new Pass();

}

if("Exit".equals(g))

{

dispose();

}

if("About".equals(g))

{

Menubar menubar = new Menubar(1);

}

if("Help Contents".equals(g))

{

Menubar menubar = new Menubar(2);

}

if(e.getSource()==jButton3)

{

Applications s=new Applications("");

}

}

public static BufferedImage readImage2(String fileLocation)

{

BufferedImage img = null;

try

{

img = ImageIO.read(new File(fileLocation));

}

catch (IOException e)

{}

return img;

}

class Interactive extends JPanel//inner claSS

{

String s[]=new String[1000];

public int cc=0,start=0,lines;

public int r=1,point=0,img=0,first=1;

public Label set = new Label("Set Questions");

BufferedImage img1 = readImage("D:/Project/images/leal.jpg");

BufferedImage img2 = readImage("D:/Project/images/game.jpg");

BufferedImage img3 = readImage("D:/Project/cat/1.png");

BufferedImage img4 = readImage("D:/Project/cat/2.png");

BufferedImage img5 = readImage("D:/Project/cat/3.png");

BufferedImage imgs=readImage("D:/Project/images/leal.jpg");

BufferedImage imgc=null;

@Override public void paintComponent(Graphics g2)

{

Graphics2D g = (Graphics2D)g2;

g.setRenderingHint(RenderingHints.KEY\_ANTIALIASING,RenderingHints.VALUE\_ANTIALIAS\_ON);

if(cc==0)

{

g.drawImage(imgs,0,0,this);

}

else

if(start==1)

{

g.drawImage(imgs,0,0,this);

g.setColor(Color.YELLOW);

g.setFont(new Font("latha",Font.ITALIC,30));

g.drawString("Click here to start !!",500,150);

}

else

{

g.drawImage(imgs,0,0,this);

g.drawImage(imgc,0,0,this);

g.setColor(Color.GREEN);

g.setFont(new Font("latha",Font.ITALIC,30));

g.drawString("Score: "+point,30,290);

g.setColor(Color.blue);

lines=readLines();

if(r<=lines)

{

g.setFont(new Font("Lucida Console",Font.ITALIC,20));

g.drawString(s[r],400,100);

}

else

{

g.setFont(new Font("Lucida Console",Font.BOLD,20));

g.drawString("Well Done !!!",580,100);

g.drawString("Your Score ( "+point+" )",550,130);

}

g.setColor(Color.YELLOW);

g.setFont(new Font("Lucida Console",Font.BOLD,20));

g.drawString("Skip ->",1250,300);

g.drawString("Set Questions",500,300);

}

}

Interactive()

{

setLayout(new BorderLayout());

addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent e) {

super.mouseClicked(e);

int x=e.getX();

int y=e.getY();

System.out.println("x: "+x+" Y: "+y);

if(((x>=500 && x<=670) && (y>=280 && y<=320)) &&(cc==1))

{

Questions q = new Questions();

}

if(((x>=1250 && x<=1340) && (y>=280 && y<=320))&&(cc==1))

{

System.out.println("Skip");

r=r+2;

if(r==5)

img=1;

if(r==13)

img=2;

if(r>lines)

Sound.play("\*");

repaint();

x=-1;

y=-1;

System.out.println("Skip");

}

if(((x>=500 && x<=750) && (y>=130 && y<=160))&&(first==1))

{

first++;

start=0;

r=1;

imgc=img5;

repaint();

}

}

});

jComboBox2.addItemListener(new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(jComboBox2.getSelectedIndex()==2)

{

System.out.println("Comboboxxx "+comboBox);

try{

try (FileReader fr = new FileReader("D:/Project/Question.txt")) {

BufferedReader br = new BufferedReader(fr);

int k=1;

while((s[k]=br.readLine())!=null)

{

k++;

}

}

}

catch(Exception er){}

cc=1;

start=1;

imgs=img2;

repaint();

}

else

{

cc=0;

r=1;

point=0;

first=1;

imgs=img1;

repaint();

}

}

});

}

public void rec(String ss){

if(ss.equalsIgnoreCase(s[r+1]))

{

img=0;

point=point+2;

r=r+2;

Sound.play("#");

if(r==5)

img=1;

if(r==13)

img=2;

imgc=img4;

repaint();

}

else

{

imgc=img3;

Sound.play("!");

repaint();

}

if(r>lines)

Sound.play("\*");

}

public BufferedImage readImage(String fileLocation)

{

BufferedImage img = null;

try

{

img = ImageIO.read(new File(fileLocation));

}

catch (IOException e)

{}

return img;

}

public int readLines()

{

int k=0;

try{

try (FileReader fr = new FileReader("D:/Project/Question.txt")) {

BufferedReader br = new BufferedReader(fr);

while(br.readLine()!=null)

{

k++;

}

}

}catch(Exception e){}

return k;

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class Filechs extends JFrame {

FileDialog fd;

public void init()

{

fd=new FileDialog(this,"Choose a File");

fd.setVisible(true);

}

public String file()

{

init();

String s=fd.getDirectory();

s=s+fd.getFile();

if((s.charAt(s.length()-1)!='g')||(s.charAt(s.length()-2)!='p')&&(s.charAt(s.length()-2)!='n')||(s.charAt(s.length()-3)!='j')&&(s.charAt(s.length()-3)!='p'))

{

JOptionPane.showMessageDialog(rootPane," !! ERROR !!\n The file you have selected is not an image");

s="";

}

return s;

}

}

class Programme extends JFrame implements ActionListener

{

JButton bb1 = new JButton("Yes");

JButton bb2 = new JButton("No");

JLabel la = new JLabel("The key did not match with any application you have added.");

JLabel la2 = new JLabel("Do you wish to assing application to this key?");

String string;

public Programme(String str)

{

string=str;

try

{

BufferedImage icon = ImageIO.read(new File("D:/project/images/icon2.png"));

setIconImage(icon);

}

catch (IOException e){}

setSize(400,200);

la.setForeground(Color.WHITE);

la2.setForeground(Color.WHITE);

bb1.setForeground(Color.WHITE);

bb2.setForeground(Color.WHITE);

bb1.setBackground(Color.BLACK);

bb2.setBackground(Color.BLACK);

add(bb1);

add(bb2);

add(la);

add(la2);

getContentPane().setBackground(Color.DARK\_GRAY);

setLocation(600,100);

setLayout(new FlowLayout());

setVisible(true);

setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

bb1.addActionListener(this);

bb2.addActionListener(this);

}

@Override

public void actionPerformed(ActionEvent e) {

if(e.getSource()==bb1)

{

Applications p = new Applications(string);

dispose();

}

if(e.getSource()==bb2)

{

dispose();

}

}

}

class Menubar extends JFrame implements MouseListener

{

BufferedImage img = readImage("D:/Project/images/gang.jpg");

BufferedImage img1 = readImage("D:/Project/images/help.jpg");

Panel p = new Panel();

JLabel jLabel1 = new JLabel("Browse");

JLabel jLabel2 = new JLabel("Recognize");

JLabel jLabel3 = new JLabel("Alphabet");

JLabel jLabel4 = new JLabel("Shape");

JLabel jLabel5 = new JLabel("Number");

JLabel jLabel6 = new JLabel("Modes");

JLabel jLabel7 = new JLabel("Add Programs");

JLabel jLabel8 = new JLabel("Sounds");

JLabel jLabel9 = new JLabel("White Box");

String contents="hggggggggggggggggggggggggggg";

int ii,no=0,mm=150;

String draw[] = new String[1000];

Menubar(int i)

{

try

{

BufferedImage icon = ImageIO.read(new File("D:/project/images/icon2.png"));

setIconImage(icon);

}

catch (IOException e){}

this.ii = i;

UI(ii);

if(i==1)

{

setSize(700,680);

}

else

if(i==2)

{

Dimension d = new Dimension(280,690);

Dimension d1 = new Dimension(600,690);

setLayout(new FlowLayout(FlowLayout.LEFT));

p.setPreferredSize(d);

p.setBackground(Color.GRAY);

add(p);

setSize(1200,740);

}

jLabel1.setForeground(Color.WHITE);

jLabel2.setForeground(Color.WHITE);

jLabel3.setForeground(Color.WHITE);

jLabel4.setForeground(Color.WHITE);

jLabel5.setForeground(Color.WHITE);

jLabel6.setForeground(Color.WHITE);

jLabel7.setForeground(Color.WHITE);

jLabel8.setForeground(Color.WHITE);

jLabel9.setForeground(Color.WHITE);

jLabel1.addMouseListener(this);

jLabel2.addMouseListener(this);

jLabel3.addMouseListener(this);

jLabel4.addMouseListener(this);

jLabel5.addMouseListener(this);

jLabel6.addMouseListener(this);

jLabel7.addMouseListener(this);

jLabel8.addMouseListener(this);

jLabel9.addMouseListener(this);

setLocation(100,50);

setVisible(true);

setResizable(false);

}

public void UI(int k)

{

if(k==2)

{

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(p);

p.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(46, 46, 46)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel9)

.addComponent(jLabel7)

.addComponent(jLabel6)

.addComponent(jLabel5)

.addComponent(jLabel8)

.addComponent(jLabel4)

.addComponent(jLabel3)

.addComponent(jLabel1)

.addComponent(jLabel2))

.addGap(0, 75, Short.MAX\_VALUE))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel4)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel8)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel5)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel6)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel7)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel9)

.addContainerGap(115, Short.MAX\_VALUE))

);

}

}

@Override public void paint(Graphics g)

{

Graphics2D g2=(Graphics2D) g;

g2.setColor(Color.YELLOW);

g2.fillRect(0,0,getWidth(),getHeight());

if(ii==1)

{

g2.drawImage(img,30,40,this);

}

else

g2.drawImage(img1,300,mm,this);

if(ii==6)

{

int k = 0;

g2.setColor(Color.BLACK);

for(int i=0;i<no;i++)

{

g2.setFont(new Font("Lucida Console",Font.BOLD,16));

g2.drawString(draw[i],330,50+k);

k=k+15;

}

no=0;

}

}

public BufferedImage readImage(String fileLocation)

{

BufferedImage img=null;

try

{

img = ImageIO.read(new File(fileLocation));

}

catch (IOException e)

{}

return img;

}

@Override

public void mouseClicked(MouseEvent e) {

String dir="";

mm=150;

if(e.getSource()==jLabel1)

{

dir="D:/Project/help/Browse.txt";

img1=readImage("D:/Project/images/Browse.jpg");

}

if(e.getSource()==jLabel2)

{

dir="D:/Project/help/Recognize.txt";

img1=readImage("D:/Project/images/recognize.jpg");

}

if(e.getSource()==jLabel3)

{

dir=("D:/Project/help/Alphabet.txt");

img1=readImage("D:/Project/images/alphabet.jpg");

}

if(e.getSource()==jLabel4)

{

dir=("D:/Project/help/Shape.txt");

img1=readImage("D:/Project/images/Shape.jpg");

}

if(e.getSource()==jLabel5)

{

dir=("D:/Project/help/Number.txt");

img1=readImage("D:/Project/images/number.jpg");

}

if(e.getSource()==jLabel6)

{

dir=("D:/Project/help/Modes.txt");

mm=300;

img1=readImage("D:/Project/images/modes.jpg");

}

if(e.getSource()==jLabel7)

{

dir=("D:/Project/help/Programes.txt");

img1=readImage("D:/Project/images/programe.jpg");

}

if(e.getSource()==jLabel8)

{

dir=("D:/Project/help/Sounds.txt");

img1=readImage("D:/Project/images/sounds.jpg");

}

if(e.getSource()==jLabel9)

{

dir=("D:/Project/help/White Box.txt");

img1=readImage("D:/Project/images/white box.jpg");

mm=200;

}

try{

try (FileReader fr = new FileReader(dir)) {

BufferedReader br = new BufferedReader(fr);

String s;

while((s=br.readLine())!=null)

{

draw[no]=s;

no++;

ii=6;

}

repaint();

}

}

catch(Exception ee){}

}

@Override

public void mousePressed(MouseEvent e) {}

@Override

public void mouseReleased(MouseEvent e) {}

@Override

public void mouseEntered(MouseEvent e) {

jLabel1.setCursor(new Cursor(Cursor.HAND\_CURSOR));

jLabel2.setCursor(new Cursor(Cursor.HAND\_CURSOR));

jLabel3.setCursor(new Cursor(Cursor.HAND\_CURSOR));

jLabel4.setCursor(new Cursor(Cursor.HAND\_CURSOR));

jLabel5.setCursor(new Cursor(Cursor.HAND\_CURSOR));

jLabel6.setCursor(new Cursor(Cursor.HAND\_CURSOR));

jLabel7.setCursor(new Cursor(Cursor.HAND\_CURSOR));

jLabel8.setCursor(new Cursor(Cursor.HAND\_CURSOR));

jLabel9.setCursor(new Cursor(Cursor.HAND\_CURSOR));

}

@Override

public void mouseExited(MouseEvent e) {}

}

class Pass extends JFrame implements ActionListener{

JLabel jLabel = new JLabel("Enter your new password");

JLabel jLabel2 = new JLabel("Enter Password:");

JPasswordField jTextField2=new JPasswordField(30);

JPasswordField jTextField=new JPasswordField(30);

JButton b = new JButton("Done");

int i;

Pass(){

i= check();

setSize(500,200);

setLayout(new FlowLayout(FlowLayout.CENTER));

add(jLabel2);

add(jTextField2);

add(jLabel);

add(jTextField);

add(b);

setVisible(true);

b.addActionListener(this);

if(i==0)

{

jLabel2.setVisible(false);

jTextField2.setVisible(false);

}

}

public int check(){

String s=null;

int i=1;

try{

FileReader fr = new FileReader("D:/project/password.txt");

BufferedReader br= new BufferedReader(fr);

s=br.readLine();

}catch(Exception e){}

if (s==null)

i=0;

return i;

}

@Override

public void actionPerformed(ActionEvent e) {

try{

FileReader fr = new FileReader("D:/project/password.txt");

BufferedReader br = new BufferedReader(fr);

String pass=jTextField.getText();

String old=jTextField2.getText();

if(i==0)

{

FileWriter fw = new FileWriter("D:/project/password.txt");

try(BufferedWriter bw = new BufferedWriter(fw)){bw.write(pass);};

dispose();

}

else

{

String s=br.readLine();

if(s.equals(old))

{

FileWriter fw = new FileWriter("D:/project/password.txt");

try(BufferedWriter bw = new BufferedWriter(fw)){bw.write(pass);};

JOptionPane.showMessageDialog(rootPane,"Correct");

dispose();

}

else

JOptionPane.showMessageDialog(rootPane,"Incorrect Password entered");

}

}catch(Exception eee){}

}

}

**MM:-**

package Paint;

import java.awt.\*;

import java.awt.image.BufferedImage;

import java.io.BufferedReader;

import java.io.File;

import java.io.FileReader;

import java.io.IOException;

import javax.imageio.ImageIO;

import javax.swing.ImageIcon;

import javax.swing.JFrame;

import javax.swing.JPanel;

public class MM extends JFrame{

BufferedImage img = null;

int save=0;

public GraphicsDevice vc;

public static String msg="";

Display dis = new Display();

public MM()

{

GraphicsEnvironment e = GraphicsEnvironment.getLocalGraphicsEnvironment();

vc = e.getDefaultScreenDevice();

setUndecorated(true);

setIgnoreRepaint(true);

setResizable(false);

add(dis);

setVisible(true);

vc.setFullScreenWindow(this);

}

public int check(){

int i=0;

String s;

try{

Runtime.getRuntime().exec("D:/project/check.bat");

Thread.sleep(1000);

FileReader fr = new FileReader("D:/Project/exist.txt");

BufferedReader br = new BufferedReader(fr);

while((s=br.readLine())!=null)

{

if (i!=0)

{

msg=msg+s;

msg=msg+"\n";

}

i++;

}

if(i>1)

save=i-1;

}catch(Exception e){}

return save;

}

public static void main(String a[])

{

new MM();

}

class Display extends JPanel

{

@Override public void paint(Graphics g)

{

g.setFont(new Font("Harlow Solid Italic",Font.PLAIN,32));

try

{

BufferedImage i = ImageIO.read(new File("D:/project/images/le.jpg"));

g.drawImage(i,0,0,this);

}

catch (IOException e){}

try

{

BufferedImage imgs = ImageIO.read(new File("D:/project/images/Lealaps.jpg"));

g.drawImage(imgs,190,110,this);

}

catch (IOException e){}

g.setColor(Color.ORANGE);

g.drawString("Lealaps Handwritten Patern Recognition",350,80);

}

}

public BufferedImage readImage(String fileLocation)

{

BufferedImage img=null;

try

{

img = ImageIO.read(new File(fileLocation));

}

catch (IOException e)

{}

return img;

}

}

**OCRChie\_ChainCode:-**

package Paint;

import java.awt.image.BufferedImage;

import java.io.IOException;

import Paint.Imagess;

public class OCRChie\_ChainCode

{

public static String c="",pre\_c="";

public static StringBuffer sentence=new StringBuffer("");

public static int chain[]=new int[10000];//this will store the chain code

public static int a[][]=new int[2][10000];//this will store the pixel coordinates

public static int h,w,k=1;//to store the 1st pixel

public static int dir;//store the final direction

public static int predir;//store the direction and pass to dir

public static int Rheight,Rwidth;

public int sel;

public int xgreatest=0,xsmallest=1000,ygreatest=0,ysmallest=1000,Sheight,Swidth,xbound,ybound;

public int xg,xs,yg,ys;

public double van[]=new double[27];

public double melv[]=new double[27];

double normalize[]=new double[27];

double sumOCR=0;

public static int limit[][]=new int[10000][10000];

int flagg=0,grtt;

public OCRChie\_ChainCode()

{}

public static void neighbour(int dir,int a,int b)

{

switch(dir)

{

case 0:

{

b++;

}

break;

case 1:

{

a--;

b++;

}

break;

case 2:

{

a--;

}

break;

case 3:

{

a--;

b--;

}

break;

case 4:

{

b--;

}

break;

case 5:

{

a++;

b--;

}

break;

case 6:

{

a++;

}

break;

case 7:

{

a++;

b++;

}

break;

}

Rheight=a;

Rwidth=b;

}

public int findFirstPixel(BufferedImage image,int p)throws IOException

{

int width = image.getWidth();

int height = image.getHeight();

int cou=0,exist=0,grt=1,n=0;

if(flagg==1)

{

grt=grtt+8;

grtt=grt;

}

else

{

if(p!=0)/\*calculate the previous greatest x axis except when checking for the 1st time that is why(p!=0)\*/

{

for(int m=0;m<k;m++)

{

if(grt<=a[1][m])

{

grt=a[1][m];

grtt=grt;

}

}

}

}

for (int i =grt+1; i<=width; i++)

{

for (int j =height-1; j>0; j--)

{

try{

int color = image.getRGB(i,j);

int red = (color & 0x00ff0000) >> 16;

int green = (color & 0x0000ff00) >> 8;

int blue = color & 0x000000ff;

if(red <50 || green<50 || blue<50 )//black

{

w=i;

h=j;

cou=1;

}

if(i>width-2)

{

n=1;

break;

}

if(cou==1)

{

break;

}

}

catch(Exception e){}

}

if(cou==1)

{

break;

}

}

a[0][0]=h;//the array will have the value of

a[1][0]=w;//the 1st pixel store

return n;

}

public StringBuffer CCOCR(BufferedImage image,int select) throws IOException {

sentence.setLength(0);

int width = image.getWidth();

int height = image.getHeight();

this.sel=select;

int aa,bb,mel=0,check=0,p=0;

do

{

check=findFirstPixel(image,p);

p++;

k=1;

mel=0;

dir=6;

if(check!=1)

{

/\*

\* check with other RGB module here in the segment im using for instead of while(flag!=1)

\*/

for(int i=1;i<(height\*width);i++) //keep on checking for neighbours

{

aa=h;//pixel obtain stored in aa and bb and also aa and bb will be

bb=w;// the coordinates of the pixel to be pass to the neighbour method

for (int j =0;j<8; j++)//traverse cycle

{

OCRChie\_ChainCode.neighbour(dir,aa,bb);//calling the neighbour function and passing the direction

// and the pixel value as parameters

try

{

int color = image.getRGB(Rwidth,Rheight);

int red = (color & 0x00ff0000) >> 16;

int green = (color & 0x0000ff00) >> 8;

int blue = color & 0x000000ff;

if(red <50 || green<50 || blue<50)

{

h=Rheight;

w=Rwidth;

predir=dir;

}//end of if red <50..............

}

catch(Exception e){}

if(dir==7)//increment the direction and also if the direction reach 7 set direction to 0

{

dir=0;

}

else

{

dir=dir+1;

}

}//end of for traverse cycle

chain[mel]=predir;

mel++;

a[0][k]=h;

a[1][k]=w;

image.setRGB(w,h,-1238236);// marking the border with red pixels

k++;

if( predir >= 4)

{

dir=predir-4;

}

else

if(predir<4)

{

dir=predir+4;

}

if( (a[0][0]==h)&&(a[1][0]==w) )

{

break;

}

}//end of for checking the neighbours

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Cropping an object and adjusting the x and y axis \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

for(int i=0;i<k;i++)

{

if(a[0][i]>ygreatest)

ygreatest=a[0][i];

if(a[0][i]<ysmallest)

ysmallest=a[0][i];

if(a[1][i]>xgreatest)

xgreatest=a[1][i];

if(a[1][i]<xsmallest)

xsmallest=a[1][i];

}

Sheight=ygreatest-ysmallest;

Swidth=xgreatest-xsmallest;

int mh=Sheight%5;

int mw=Swidth%5;

if(mh==4)

{

ysmallest=ysmallest-1;

}

if(mh==3)

{

ysmallest=ysmallest-1;

ygreatest=ygreatest+1;

}

if(mh==2)

{

ysmallest=ysmallest-2;

ygreatest=ygreatest+1;

}

if(mh==1)

{

ysmallest=ysmallest-2;

ygreatest=ygreatest+2;

}

if(mh==0)

{

ysmallest=ysmallest-3;

ygreatest=ygreatest+2;

}

if(mw==4)

{

xsmallest=xsmallest-1;

}

if(mw==3)

{

xsmallest=xsmallest-1;

xgreatest=xgreatest+1;

}

if(mw==2)

{

xsmallest=xsmallest-2;

xgreatest=xgreatest+1;

}

if(mw==1)

{

xsmallest=xsmallest-2;

xgreatest=xgreatest+2;

}

if(mw==0)

{

xsmallest=xsmallest-3;

xgreatest=xgreatest+2;

}

xsmallest=xsmallest-3;

xgreatest=xgreatest+2;

ysmallest=ysmallest-3;

ygreatest=ygreatest+2;

Sheight=ygreatest-ysmallest;

Swidth=xgreatest-xsmallest;

mh=Sheight%5;

mw=Swidth%5;

xbound=Swidth/5;

ybound=Sheight/5;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

for(int i=0;i<k;i++)

{

for(int j=0;j<k;j++)

limit[i][j]=-10;

}

for(int i=0;i<k;i++)

{

limit[a[1][i]][i]=a[0][i];

}

float c0=0,c1=0,c2=0,c3=0,c4=0,c5=0,c6=0,c7=0;

for(int kkk=0;kkk<mel;kkk++)

{

if(chain[kkk]==0)

c0++;

else

if(chain[kkk]==1)

c1++;

else

if(chain[kkk]==2)

c2++;

else

if(chain[kkk]==3)

c3++;

else

if(chain[kkk]==4)

c4++;

else

if(chain[kkk]==5)

c5++;

else

if(chain[kkk]==6)

c6++;

else

if(chain[kkk]==7)

c7++;

}

float sum=c0+c1+c2+c3+c4+c5+c6+c7

double a[]=new double[35];

a[0]=c0/sum;

a[1]=c1/sum;

a[2]=c2/sum;

a[3]=c3/sum;

a[4]=c4/sum;

a[5]=c5/sum;

a[6]=c6/sum;

a[7]=c7/sum;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* OCRchie \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

for (int i =1; i<=width; i++)

{

for (int j =height-1; j>0; j--)

{

try{

int color = image.getRGB(i,j);

int red = (color & 0x00ff0000) >> 16;

int green = (color & 0x0000ff00) >> 8;

int blue = color & 0x000000ff;

if(red <50 || green<50 || blue<50 )//black

{

w=i;

h=j;

int sml=1000,grt=-1;

for(int ie=0;ie<k;ie++)

{

if(limit[i][ie]>=0)

{

if(limit[i][ie]>=grt)

grt=limit[i][ie];

if(limit[i][ie]<=sml)

sml=limit[i][ie];

}

}

if((j>=sml)&&(j<=grt))

{

if( (i>=xsmallest && i<xsmallest+xbound) && (j>=ysmallest && j<ysmallest+ybound) )

{

van[0]=van[0]+1;

}

else

if( (i>=xsmallest+xbound && i<xsmallest+(xbound\*2)) && (j>=ysmallest && j<ysmallest+ybound) )

{

van[1]=van[1]+1;

}

else

if( (i>=xsmallest+(xbound\*2) && i<xsmallest+(xbound\*3)) && (j>=ysmallest && j<ysmallest+ybound) )

{

van[2]=van[2]+1;

}

else

if( (i>=xsmallest+(xbound\*3) && i<xsmallest+(xbound\*4)) && (j>=ysmallest && j<ysmallest+ybound) )

{

van[3]=van[3]+1;

}

else

if( (i>=xsmallest+(xbound\*4) && i<xsmallest+(xbound\*5)) && (j>=ysmallest && j<ysmallest+ybound) )

{

van[4]=van[4]+1;

}

else

if( (i>=xsmallest && i<xsmallest+xbound) && (j>=ysmallest+ybound && j<ysmallest+(ybound\*2)) )

{

van[5]=van[5]+1;

}

else

if( (i>=xsmallest+xbound && i<xsmallest+(xbound\*2)) && (j>=ysmallest+ybound && j<ysmallest+(ybound\*2)) )

{

van[6]=van[6]+1;

}

else

if( (i>=xsmallest+(xbound\*2) && i<xsmallest+(xbound\*3)) && (j>=ysmallest+ybound && j<ysmallest+(ybound\*2)) )

{

van[7]=van[7]+1;

}

else

if( (i>=xsmallest+(xbound\*3) && i<xsmallest+(xbound\*4)) && (j>=ysmallest+ybound && j<ysmallest+(ybound\*2)) )

{

van[8]=van[8]+1;

}

else

if( (i>=xsmallest+(xbound\*4) && i<xsmallest+(xbound\*5)) && (j>=ysmallest+ybound && j<ysmallest+(ybound\*2)) )

{

van[9]=van[9]+1;

}

else

if( (i>=xsmallest && i<xsmallest+xbound) && (j>=ysmallest+(ybound\*2) && j<ysmallest+(ybound\*3)) )

{

van[10]=van[10]+1;

}

else

if( (i>=xsmallest+xbound && i<xsmallest+(xbound\*2)) && (j>=ysmallest+(ybound\*2) && j<ysmallest+(ybound\*3)) )

{

van[11]=van[11]+1;

}

else

if( (i>=xsmallest+(xbound\*2) && i<xsmallest+(xbound\*3)) && (j>=ysmallest+(ybound\*2) && j<ysmallest+(ybound\*3)) )

{

van[12]=van[12]+1;

}

else

if( (i>=xsmallest+(xbound\*3) && i<xsmallest+(xbound\*4)) && (j>=ysmallest+(ybound\*2) && j<ysmallest+(ybound\*3)) )

{

van[13]=van[13]+1;

}

else

if( (i>=xsmallest+(xbound\*4) && i<xsmallest+(xbound\*5)) && (j>=ysmallest+(ybound\*2) && j<ysmallest+(ybound\*3)) )

{

van[14]=van[14]+1;

}

else

if( (i>=xsmallest && i<xsmallest+xbound) && (j>=ysmallest+(ybound\*3) && j<ysmallest+(ybound\*4)) )

{

van[15]=van[15]+1;

}

else

if( (i>=xsmallest+xbound && i<xsmallest+(xbound\*2)) && (j>=ysmallest+(ybound\*3) && j<ysmallest+(ybound\*4)) )

{

van[16]=van[16]+1;

}

else

if( (i>=xsmallest+(xbound\*2) && i<xsmallest+(xbound\*3)) && (j>=ysmallest+(ybound\*3) && j<ysmallest+(ybound\*4)) )

{

van[17]=van[17]+1;

}

else

if( (i>=xsmallest+(xbound\*3) && i<xsmallest+(xbound\*4)) && (j>=ysmallest+(ybound\*3) && j<ysmallest+(ybound\*4)) )

{

van[18]=van[18]+1;

}

else

if( (i>=xsmallest+(xbound\*4) && i<xsmallest+(xbound\*5)) && (j>=ysmallest+(ybound\*3) && j<ysmallest+(ybound\*4)) )

{

van[19]=van[19]+1;

}

else

if( (i>=xsmallest && i<xsmallest+xbound) && (j>=ysmallest+(ybound\*4) && j<ysmallest+(ybound\*5)) )

{

van[20]=van[20]+1;

}

else

if( (i>=xsmallest+xbound && i<xsmallest+(xbound\*2)) && (j>=ysmallest+(ybound\*4) && j<ysmallest+(ybound\*5)) )

{

van[21]=van[21]+1;

}

else

if( (i>=xsmallest+(xbound\*2) && i<xsmallest+(xbound\*3)) && (j>=ysmallest+(ybound\*4) && j<ysmallest+(ybound\*5)) )

{

van[22]=van[22]+1;

}

else

if( (i>=xsmallest+(xbound\*3) && i<xsmallest+(xbound\*4)) && (j>=ysmallest+(ybound\*4) && j<ysmallest+(ybound\*5)) )

{

van[23]=van[23]+1;

}

else

if( (i>=xsmallest+(xbound\*4) && i<xsmallest+(xbound\*5)) && (j>=ysmallest+(ybound\*4) && j<ysmallest+(ybound\*5)) )

{

van[24]=van[24]+1;

}

int mid=( (ygreatest-ysmallest)/2 )+ysmallest;

if(j<mid)

{

van[25]=van[25]+1;

}

else

if(j>=mid)

{

van[26]=van[26]+1;

}

}

}

}

catch(Exception e){}

}

}

for(int i=0;i<27;i++)

{

melv[i]=van[i];

}

sumOCR=0;

for(int i=0;i<27;i++)

{

if((i!=25)&&(i!=26))

{

sumOCR=sumOCR+van[i];

}

van[i]=0;

}

for(int i=0;i<27;i++)

{

normalize[i]=melv[i]/sumOCR;

}

yg=ygreatest;

ys=ysmallest;

xg=xgreatest;

xs=xsmallest;

ygreatest=0;

ysmallest=1000;

xgreatest=0;

xsmallest=1000;

int k=0;

for(int i=8;i<35;i++)

{

a[i]=normalize[k];

k++;

}

NewFileData t=new NewFileData();

c=t.init(a,sel);

if(sum>40) /\* If the object have less than 60 borderpixel no result will be produce \*/

{

if( (flagg==1) && (!c.equals("l")) )

{

flagg=0;

}

if( (flagg==1) && (c.equals("l")) )

{

c="i";

flagg=0;

}

if(c.equals("l"))

{

String cd=i(image);

c=cd;

}

if( ( pre\_c.equals("l")||( pre\_c.equals("I") ) )&&( c.equals("!") ) )

{

sentence.setCharAt((sentence.length())-1,'k');

}

else

{

pre\_c=c;

sentence.append(c);

}

}

else

{

flagg=1;

}

}//End of if check!=1

}while(check!=1);

return sentence;

}//end of constructor

public String i(BufferedImage im)

{

String s="";

int no=0;

for(int i=xs;i<=xg;i++)

{

for(int j=0;j<ys;j++)

{

try{

int color = im.getRGB(i,j);

int red = (color & 0x00ff0000) >> 16;

int green = (color & 0x0000ff00) >> 8;

int blue = color & 0x000000ff;

if(red <50 || green<50 || blue<50 )//black

{

no++;

im.setRGB(i,j,-14438067);

}

}

catch(Exception e){}

}

}

if(no>20)

s="i";

else

s="l";

return s;

}

}

**NewFileData:-**

package Paint;

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.IOException;

public class NewFileData

{

public double aa[]=new double[40];

String c;

int select;

String file\_name="D:/Project/TrainingSet.txt";

public NewFileData()

{

}

public String init(double a[],int select)

{

this.aa=a;

this.select=select;

{

if(this.select==1)

{

file\_name="D:/Project/TrainingSet\_duplicate.txt";

}

if(this.select==2)

{

file\_name="D:/Project/TrainingSets\_shape.txt";

}

if(this.select==3)

{

file\_name="D:/Project/Training\_Sets\_no.txt";

}

if(this.select==4)

{

file\_name="D:/Project/Training\_Sets\_B\_duplicate.txt";

}

try

{

ReadFiles file=new ReadFiles(file\_name,aa,this.select);

c=file.openFile();

}

catch(IOException e)

{

System.out.println(e.getMessage());

}

}

return c;

}

}

class ReadFiles

{

public String path;

double bb[];

double x[]=new double[40];

double r,pre\_r;

char character,match;

String characters="",matchs="";

int sel;

String s;

public ReadFiles()

{

}

public ReadFiles(String file\_path,double a[],int select)

{

bb=a;

path=file\_path;

sel=select;

}

public String openFile()throws IOException

{

FileReader fr=new FileReader(path);

try (BufferedReader textReader = new BufferedReader(fr)) {

int no\_of\_lines=readLines();

String[] textData=new String[no\_of\_lines];

String g="";

StringBuffer tt=new StringBuffer(890);

double sum=0;

int k=0;

for(int i=0;i<no\_of\_lines;i++)

{

textData[i]=textReader.readLine();

for(int j=0;j<textData[i].length();j++)

{

if(textData[i].charAt(j)!=' ')

{

if(k==35)

{

if(sel==2)

{

characters=" ";

while(j!=(textData[i].length()))

{

characters=characters+textData[i].charAt(j);

j++;

}

}

else

{

character=textData[i].charAt(j);

}

break;

}

else

{

g=g+textData[i].charAt(j);

}

}

else

{

if(!" ".equals(g))

{

x[k]=Double.parseDouble(g);

k++;

}

g=" ";

}

}

Compares fj=new Compares();

pre\_r=r;

r=fj.init(bb,x,i);

if(r!=pre\_r)

{

match=character;

matchs=characters;

}

k=0;

}

System.out.println("The Score is:"+r);

System.out.println("The Alphabet is:"+match);

if(sel==2)

{

s=matchs.toString();

}

else

{

s=String.valueOf(match);

}

}

return s;

}

public int readLines()throws IOException

{

FileReader file\_to\_read=new FileReader(path);

int no\_of\_lines;

try (BufferedReader bf = new BufferedReader(file\_to\_read)) {

String aLine;

no\_of\_lines = 0;

while((aLine=bf.readLine())!=null)

{

no\_of\_lines++;

}

}

return no\_of\_lines;

}

}

class Compares

{

public static double result[]=new double[40];

public static double sums,pre,smallest;

public double init(double a[],double x[],int i)

{

for(int h=0;h<35;h++)

{

if((h!=33)&&(h!=34))

{

result[h]=a[h]-x[h];

if(result[h]<0)

{

result[h]=((result[h]\*-2)+result[h]);

}

}

else

{

result[h]=a[h];

}

sums=sums+result[h];

}

if(i==0)

{

pre=sums;

}

else

{

if(pre<sums)

{

sums=pre;

pre=sums;

}

else

{

pre=sums;

}

}

smallest=sums;

sums=0;

return smallest;

}

}

**PaintApplications:-**

package Paint;

import java.awt.\*;

import java.awt.event.\*;

import java.awt.image.BufferedImage;

import java.io.File;

import javax.imageio.ImageIO;

import javax.swing.JPanel;

public class PaintApplications extends JPanel implements ActionListener{

public static int x=-1,y=-1,xx=-1,yy=-1,rotate=0,flag=0, clear=0,save=0;

public BufferedImage img=new BufferedImage(620,410,BufferedImage.TYPE\_INT\_ARGB);

Button b1 = new Button("Clear");

public PaintApplications()

{

setLayout(new FlowLayout());

add(b1);

setVisible(true);

b1.addActionListener(this);

addMouseMotionListener(new MouseMotionAdapter() {

@Override

public void mouseDragged(MouseEvent e) {

super.mouseDragged(e);

x = e.getX();

y = e.getY();

if((y<390)&&(x<590)&&(y>10)&&(x>10))

{

repaint();

}

}

});

addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent e) {

xx=-1;

yy=-1;

}

@Override

public void mouseExited(MouseEvent e) {

super.mouseExited(e);

save=1;

repaint();

}

@Override

public void mouseReleased(MouseEvent e) {

super.mouseReleased(e);

x=e.getX();

y=e.getY();

if((y<390)&&(x<590)&&(y>10)&&(x>10))

{

repaint();

}

xx=-1;

yy=-1;

flag=1;

}

});

}

@Override

public void paint(Graphics g){

if(save!=1)

{

Graphics2D g2 = (Graphics2D)g;

Graphics2D g3 = img.createGraphics();

g2.setRenderingHint(RenderingHints.KEY\_ANTIALIASING,RenderingHints.VALUE\_ANTIALIAS\_ON);

g3.setRenderingHint(RenderingHints.KEY\_ANTIALIASING,RenderingHints.VALUE\_ANTIALIAS\_ON);

g2.setStroke(new BasicStroke(7));

g3.setStroke(new BasicStroke(7));

g2.setColor(Color.BLACK);

g3.setColor(Color.BLACK);

if(clear==1)

{

g2.setColor(Color.WHITE);

g3.setColor(Color.WHITE);

g2.fillRect(0,0,3000,600);

g3.fillRect(0,0,3000,600);

clear=0;

xx=-1;

yy=-1;

}

else

{

if((xx!=-1)&&(yy!=-1))

{

g2.drawLine(xx,yy,x,y);

g3.drawLine(xx,yy,x,y);

xx=x;

yy=y;

}

else

{

xx=x;

yy=y;

if(flag==1)

{

xx=-1;

yy=-1;

flag=0;

}

}

}

}

if(save==1)

{

try{

ImageIO.write(img,"PNG",new File("D:/Project/paint/100.jpg"));

}catch(Exception e){}

save=0;

System.out.println("Saved");

}

}

@Override

public void actionPerformed(ActionEvent e) {

if(e.getSource()==b1)

{

clear=1;

repaint();

}

}

}

**Exe:-**

package Paint;

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.IOException;

public final class Exe {

String text[]=new String[50];

String key="";

String dir="";

int ret=0;

public Exe()

{

}

public int exe(String open)

{

try{

FileReader fr = new FileReader("D:/Project/melvin.txt");

BufferedReader br = new BufferedReader(fr);

int no=readLines();

String s;

int k=0;

for(int i=0;i<no;i++)

{

text[i]=br.readLine();

System.out.println("text "+text[i]);

for(int j=0;j<text[i].length();j++)

{

if(text[i].charAt(j)!=' ' && k<1)

{

key=key+text[i].charAt(j);

}

else

{

k=1;

dir=dir+text[i].charAt(j);

}

}

System.out.println("Key: "+key);

System.out.println("Dir: "+dir);

if(open.equalsIgnoreCase(key))

{

try

{

Runtime.getRuntime().exec(dir);

}

catch(Exception e){System.out.println("ERROR: "+e);}

ret=1;

}

key="";

dir="";

k=0;

}

}

catch(Exception e){System.out.println(e);}

return ret;

}

public int readLines()throws IOException

{

FileReader file\_to\_read=new FileReader("D:/Project/melvin.txt");

int no\_of\_lines;

try (BufferedReader bf = new BufferedReader(file\_to\_read)) {

String aLine;

no\_of\_lines = 0;

while((aLine=bf.readLine())!=null)

{

no\_of\_lines++;

}

}

return no\_of\_lines;

}

}

**Applications:-**

package Paint;

import java.awt.Color;

import java.awt.FileDialog;

import java.awt.TextArea;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.image.BufferedImage;

import java.io.\*;

import javax.imageio.ImageIO;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JTextField;

public class Applications extends JFrame implements ActionListener{

public JButton jButton1 = new JButton("Done");

public JButton jButton2 = new JButton("Browse..");

public JButton jButton3 = new JButton("Update");

public JButton jButton4 = new JButton("OK");

public JTextField jTextField1=new JTextField(100);

public JTextField jTextField2=new JTextField(100);

public JLabel jLabel1=new JLabel("Key:");

public JLabel jLabel2=new JLabel("Path:");

public JLabel jLabel4=new JLabel("Key");

public JLabel jLabel3=new JLabel("Programs");

public TextArea textArea1=new TextArea();

private Filech ff=new Filech();

public String str;

public String sources;

public final void inits()

{

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(43, 43, 43)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel1)

.addComponent(jLabel2))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(jTextField1, javax.swing.GroupLayout.DEFAULT\_SIZE, 232, Short.MAX\_VALUE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED))

.addGroup(layout.createSequentialGroup()

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, 198, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(36, 36, 36)))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jButton2)

.addComponent(jButton1)))

.addComponent(textArea1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(layout.createSequentialGroup()

.addGap(44, 44, 44)

.addComponent(jLabel4)

.addGap(147, 147, 147)

.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED\_SIZE, 62, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 0, Short.MAX\_VALUE)))

.addContainerGap())

.addGroup(layout.createSequentialGroup()

.addGap(26, 26, 26)

.addComponent(jButton3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton4)

.addGap(61, 61, 61))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel2)

.addComponent(jButton2))

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel1)

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton1))

.addGap(43, 43, 43)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel4)

.addComponent(jLabel3))

.addGap(1, 1, 1)

.addComponent(textArea1, javax.swing.GroupLayout.PREFERRED\_SIZE, 215, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButton3)

.addComponent(jButton4))

.addContainerGap(22, Short.MAX\_VALUE))

);

}

public Applications(String st)

{

jTextField2.setText(st);

inits();

try

{

BufferedImage icon = ImageIO.read(new File("D:/project/lealaps.jpg"));

setIconImage(icon);

}

catch (IOException e){}

getContentPane().setBackground(Color.GRAY);

setVisible(true);

setLocation(600,20);

setSize(500,500);

setDefaultCloseOperation(JFrame.HIDE\_ON\_CLOSE);

jButton1.addActionListener(this);

jButton2.addActionListener(this);

jButton3.addActionListener(this);

jButton4.addActionListener(this);

jButton4.setVisible(false);

try{

try (FileReader fr = new FileReader("D:/Project/melvin.txt")) {

BufferedReader br = new BufferedReader(fr);

String s;

while((s=br.readLine())!=null)

{

System.out.println("s: "+s);

textArea1.append(s+"\n");

}

}

}

catch(Exception e){System.out.println(e);}

jTextField1.disable();

textArea1.disable();

}

@Override public void actionPerformed(ActionEvent ae)

{

if(ae.getSource()==jButton1)

{

String s=jTextField2.getText();

String h=jTextField1.getText();

if(s.length()!=0 && h.length()!=0)

{

int i=1;

sources=s+" "+str;

try{

FileWriter f1 = new FileWriter("D:/Project/Melvin.txt",true);

try (BufferedWriter br = new BufferedWriter(f1)) {

br.write(sources);

br.newLine();

}

}catch(Exception e){}

dispose();

}

}

if(ae.getSource()==jButton2)

{

str=ff.file();

jTextField1.setText(str);

}

if(ae.getSource()==jButton3)

{

jButton4.setVisible(true);

textArea1.enable();

}

if(ae.getSource()==jButton4)

{

String b=textArea1.getText();

try{

FileWriter f1 = new FileWriter("D:/Project/Melvin.txt");

try (BufferedWriter br = new BufferedWriter(f1)) {

br.write(b);

br.newLine();

}

}catch(Exception e){}

dispose();

}

}

}

class Filech extends JFrame

{

FileDialog fd;

public void init()

{

fd=new FileDialog(this,"Choose a File");

fd.setVisible(true);

}

public String file()

{

init();

String s=fd.getDirectory();

s=s+fd.getFile();

return s;

}

}

**Sound:-**

package Paint;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileWriter;

import javax.sound.sampled.AudioInputStream;

import javax.sound.sampled.AudioSystem;

import javax.sound.sampled.Clip;

import javax.swing.JFrame;

public class Sound extends JFrame{

public static int p=0;

public static Clip clip;

public Sound(String c)

{

String sss="";

if(c=="\*")

sss="D:/Project/sounds/complete.wav";

else

if(c=="#")

sss="D:/Project/sounds/correct.wav";

else

if(c=="!")

sss="D:/Project/sounds/wrong.wav";

else

{

try

{

FileWriter fw = new FileWriter("D:/Project/sound.txt");

try( BufferedWriter bw = new BufferedWriter(fw);){

bw.write(c);

}

Thread.sleep(500);

Runtime.getRuntime().exec("cmd.exe /c start test.vbs",null,new File("D:/Project/"));

}catch(Exception e){}

sss="";

}

try{

if(!sss.equals(""))

{

AudioInputStream audio=AudioSystem.getAudioInputStream(new File(sss).getAbsoluteFile());

clip=AudioSystem.getClip();

clip.open(audio);

clip.start();

Thread.sleep(000);

}

}

catch(Exception e){

System.out.println(e);}

}

public static void play(String c)

{

Sound s = new Sound(c);

}

}

**Questions:-**

package Paint;

import java.awt.FlowLayout;

import java.awt.TextArea;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.FileReader;

import java.io.FileWriter;

import javax.swing.\*;

public class Questions extends JFrame implements ActionListener

{

JLabel jLabel = new JLabel("Enter your password");

JButton jButton = new JButton("OK");

JPasswordField jTextField= new JPasswordField(20);

public Questions()

{

setSize(500,100);

setLayout(new FlowLayout());

add(jLabel);

add(jTextField);

add(jButton);

setVisible(true);

setLocation(500,400);

setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

jButton.addActionListener(this);

}

@Override

public void actionPerformed(ActionEvent e) {

if(e.getSource()==jButton)

{

String enter=jTextField.getText();

int pass=password(enter);

if(pass==1)

{

Questions.PrepareQuestions pq=new Questions.PrepareQuestions();

dispose();

}

else

JOptionPane.showMessageDialog(rootPane,"Wrong");

}

}

public int password(String enter)

{

int password=0;

try{

try (FileReader fr = new FileReader("D:/project/password.txt")) {

BufferedReader br = new BufferedReader(fr);

String s=br.readLine();

if(s.equals(enter))

password=1;

}

}

catch(Exception ee){}

return password;

}

class PrepareQuestions extends JFrame implements ActionListener

{

Questions.Q q = new Questions.Q();

JButton jButton = new JButton("OK");

JButton jButton2 = new JButton("Clear Previous Questions");

JButton jButton3 = new JButton("Show Questions");

JLabel jLabel = new JLabel("Enter Question here:");

JLabel jLabel2 = new JLabel("Enter Answer here:");

JTextField jTextField= new JTextField(20);

JTextField jTextField2= new JTextField(20);

PrepareQuestions()

{

design();

setResizable(false);

setVisible(true);

setLocation(500,400);

setSize(520,250);

setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

jButton.addActionListener(this);

jButton2.addActionListener(this);

jButton3.addActionListener(this);

}

public void design()

{

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addGap(0, 0, Short.MAX\_VALUE)

.addComponent(jButton2))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton))

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, layout.createSequentialGroup()

.addGap(20, 20, 20)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel2)

.addGap(18, 18, 18)

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, 345, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jTextField))))

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, layout.createSequentialGroup()

.addGap(185, 185, 185)

.addComponent(jButton3)))

.addGap(0, 13, Short.MAX\_VALUE)))

.addContainerGap())

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(17, 17, 17)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel)

.addComponent(jTextField, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel2)

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addComponent(jButton)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jButton2)

.addGap(15, 15, 15)

.addComponent(jButton3)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

}

public void write()

{

try{

FileReader fr = new FileReader("D:/Project/Question.txt");

BufferedReader br=new BufferedReader(fr);

FileWriter fw = new FileWriter("D:/Project/Question.txt",true);

try(BufferedWriter bw=new BufferedWriter(fw)){

String que=jTextField.getText();

String ans=jTextField2.getText();

if(que.length()!=0 && ans.length()!=0)

{

System.out.println(que+" "+ans);

if(br.readLine()!=null)

bw.newLine();

bw.write(que);

bw.newLine();

bw.write(ans);

jTextField.setText("");

jTextField2.setText("");

}

que=null;

ans=null;

}

}catch(Exception e){}

}

public void showQ()

{

int i=1;

try{

FileReader fr = new FileReader("D:/Project/Question.txt");

BufferedReader br=new BufferedReader(fr);

String sss,s="";

while((sss=br.readLine())!=null)

{

if(i%2==0)

{

s=s+"Ans) "+sss+"\n";

}

else

{

s=s+"Que) "+sss+"\n";

}

i++;

}

q.textArea.setText(s);

q.init()

}catch(Exception e){}

}

public void clear()

{

try{

FileWriter fw = new FileWriter("D:/Project/Question.txt");

BufferedWriter bw=new BufferedWriter(fw);

bw.write("");

}catch(Exception e){}

}

@Override

public void actionPerformed(ActionEvent e) {

if(e.getSource().equals(jButton))

{

write();

}

if(e.getSource().equals(jButton2))

{

clear();

}

if(e.getSource().equals(jButton3))

{

showQ();

}

}

}

class Q extends JFrame

{

TextArea textArea = new TextArea();

public void init()

{

setSize(500,500);

setLocation(870,0);

add(textArea);

setVisible(true);

setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

}

}

}

**Remote1:-**

import java.awt.Robot;

import java.awt.event.InputEvent;

import java.net.DatagramPacket;

import java.net.DatagramSocket;

import java.net.InetAddress;

import java.util.Timer;

import java.util.TimerTask;

public class Remote1 implements Runnable {

private static Robot robot;

public static int release=0,m=0;;

private static String message;

public static int flag=0;

DatagramSocket serverSocket;

Thread t;

Remote1(){

//JFrame.

t=new Thread(this,"");

try{robot = new Robot();}catch(Exception e){}

t.start();

}

public void close(){

Main.jTextField.setText("Connection Terminated");

serverSocket.close();

m=0;

}

public void receive()throws Exception{

int x=0,y=0;

String xy="";

InetAddress a =InetAddress.getLocalHost();

String IPP = a.toString();

String IP="";

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

{

if(IPP.charAt(i)=='/')

break;

IP=IP+IPP.charAt(i);

}

Main.jTextField.setText("connect your remote device to host '"+IP+"'");

serverSocket = new DatagramSocket(9876);

byte[] receiveData = new byte[1024];

while(true){

DatagramPacket receivePacket = new DatagramPacket(receiveData,receiveData.length);

serverSocket.receive(receivePacket);

if(m==0)

Main.jTextField.setText("connected");

m=1;

message = new String(receivePacket.getData(),0,receivePacket.getLength());

System.out.println(message);

if(message.equals("recognize"))

{

robot.mouseMove(662,70);

robot.mousePress(InputEvent.BUTTON1\_MASK);

Thread.sleep(300);

robot.mouseRelease(InputEvent.BUTTON1\_MASK);

System.out.println("Recognize");

}

else

if(message.equals("clear"))

{

robot.mouseMove(300,70);

robot.mousePress(InputEvent.BUTTON1\_MASK);

Thread.sleep(300);

robot.mouseRelease(InputEvent.BUTTON1\_MASK);

Thread.sleep(300);

robot.mouseMove(662,150);

System.out.println("clear");

}

else

if(message.equals("up"))

{

robot.mouseRelease(InputEvent.BUTTON1\_MASK);

}

else

{

robot.mousePress(InputEvent.BUTTON1\_MASK);

for(int i=0;i<=message.length()-1;i++)

{

if(message.charAt(i)==',')

{

x=Integer.parseInt(xy);

xy="";

}

else

xy=xy+message.charAt(i);

}

y=Integer.parseInt(xy);

xy="";

robot.mouseMove(x,y);

}

}

}

public static void main(String args[])throws Exception{

Remote1 r = new Remote1();

}

@Override

public void run() {

try{

receive();

}

catch(Exception err){}

}

}