Pimpri Chinchwad Education Trust’s

Pimpri Chinchwad College of Engineering

Department of Computer Engineering

STQA Mini Project

Quiz Desktop App

September 04, 2020

Members :

1. Aishwarya Patil (BECOB228)
2. Swapnajit Patil(BECOB229)
3. Sachin Rokade(BECOB240)
4. Vijay Sharma (BECOB249)

Guide:

Prof. Shrikant Kokate

Index

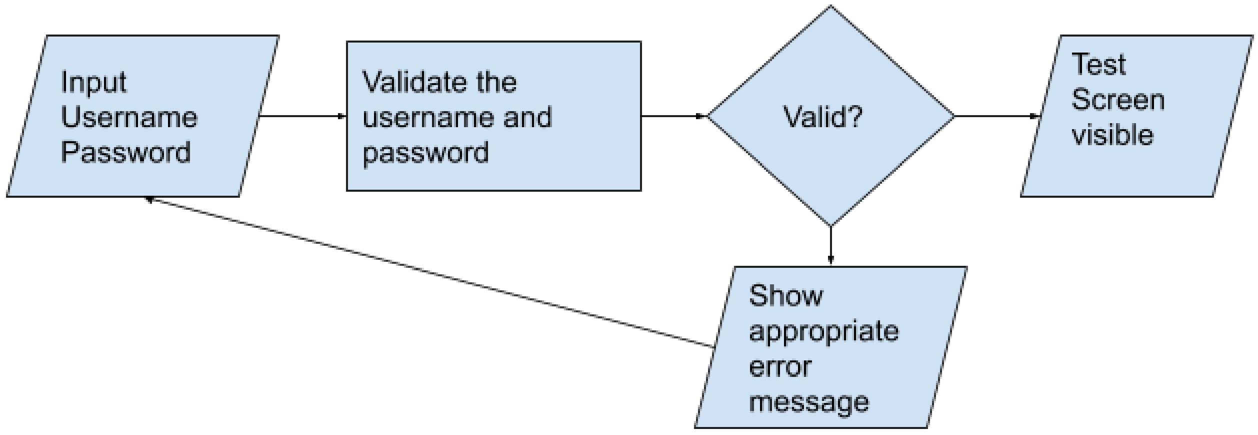
Introduction

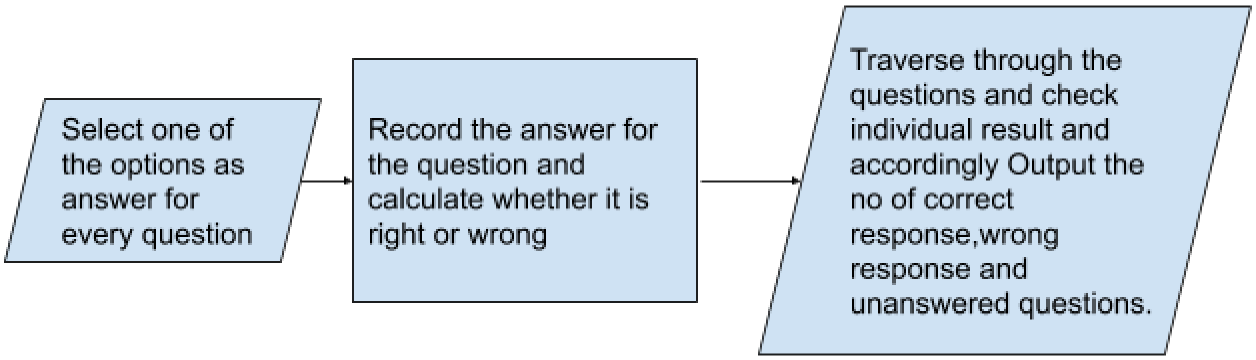
We have developed a desktop application to take quizzes on any topic. The application starts with a login form to enter into the quiz. Once entered, the quiz starts in full screen mode with a timer. Participant selects an answer and clicks the next button. Once answered , he can't navigate back to the previous question. User has to answer all the questions within time. Either the user has answered all questions or the timer is over, the result screen shows the analysis of performance. User clicks the exit button to come out of full screen mode. This application is useful in conducting online tests from remote places. The quiz can be given at your own convenience at any time and place. The full screen mode makes it impossible for the participant to switch to other apps.

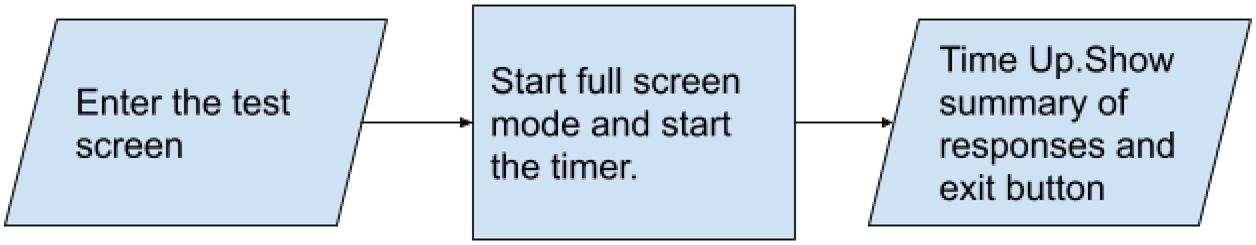
Functional Requirements

Functional requirements define the basic system behaviour. Essentially, they are what the system does or must not do, and can be thought of in terms of how the system responds to inputs.

- The main function of the application is to record participant's answers and show the correct result.
- The other functional requirements include the authorization of valid participants to take the quiz. Invalid participants should not be allowed to see the test page.
- The quiz should be auto submitted if the timer is over.
- The participant should not be able to exit the full screen mode.
- The close button on the login screen and exit button on result screen should close the app.


Input Username Password
Validate the username and password
Valid?
Test Screen visible
Show appropriate error message


Select one of the options as answer for every question
Record the answer for the question and calculate whether it is right or wrong
Traverse through the questions and check individual result and accordingly Output the no of correct response,wrong response and unanswered questions.


Enter the test screen
Start full screen mode and start the timer.
Time Up.Show summary of responses and exit button

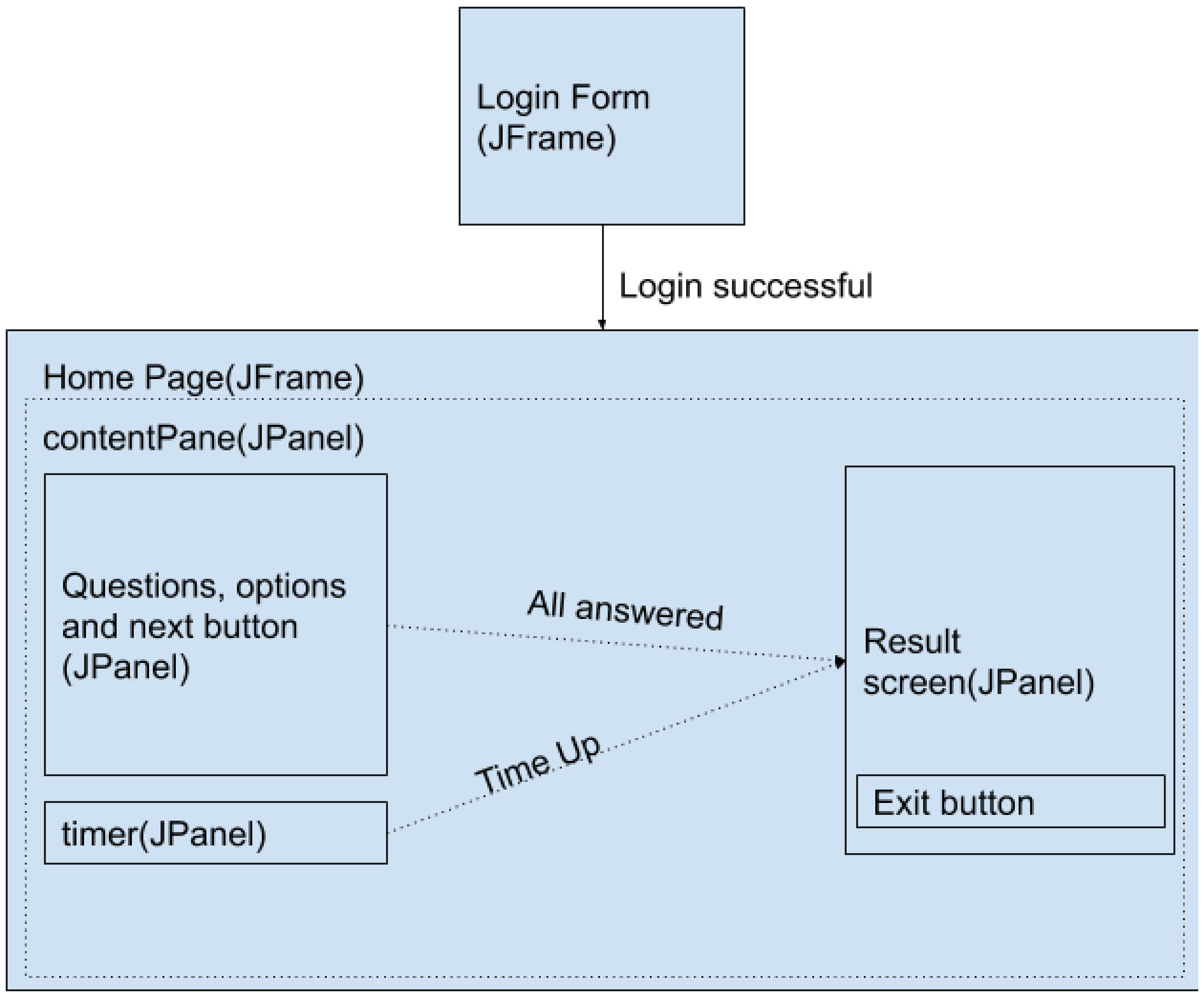
Non Functional Requirements

While *functional* requirements define what the system does or must not do, *non-functional* requirements specify **how** the system should do it.

- The background color should be consistent across all screens.
- The button should be highlighted on hovering.

- Appropriate error messages or instructions should be shown, for example, enter username, enter password, invalid credentials, all fields are mandatory to fill.
- Close button to abort the application.
- Font size,style,weight,family should be consistent across all screens.
- Timer is bright enough to be visible.
- Result and exit button should be visible immediately after the quiz. No need to search various options and menus.
- Tests are loaded quickly as soon as credentials are validated.

Design (BLOCK DIAGRAM)



Source Code

Question.java

```
package com.vijay.apps;

public class Question {

    private String question;

    private  String option1,option2,option3,option4;

    private  int correctAnswer;

    private  int selectedAnswer;

    private  int result;

    public void setSelectedAnswer(int s) {

            selectedAnswer=s;

            calculateResult();

    }

    public String getQuestion()

    {

            return question;

    }

    public String get1()

    {

            return option1;

    }

    public String get2()

    {

            return option2;

    }
```

```
public String get3()
  {
          return option3;
  }
public String get4()
  {
          return option4;
  }

  public int getResult() {
          return result;
  }
   void calculateResult(){

  this.result = this.selectedAnswer == this.correctAnswer ? 1 : 0;


   }

  public Question(String  question, int correctAnswer,
                 String  option1,
                 String  option2,
                 String  option3,
                 String  option4
                 ) {
          // TODO Auto-generated constructor stub
           this.correctAnswer=correctAnswer;
```

```
            this.option1=option1;
            this.option2=option2;
            this.option3=option3;
            this.option4=option4;
            this.question=question;
           this.result=-1;
   }
}
```

LoginForm.java

```
package com.vijay.apps;
import java.awt.BorderLayout;
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.border.LineBorder;
import java.awt.SystemColor;
import java.awt.Color;
import javax.swing.JTextField;
import javax.swing.SwingConstants;
import javax.swing.JPasswordField;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import java.awt.Font;
import java.awt.Image;
import javax.swing.ImageIcon;
import java.awt.Rectangle;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.FocusAdapter;
import java.awt.event.FocusEvent;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.event.ActionEvent;

public class LoginForm extends JFrame {

    private JPanel contentPane;
    private JTextField txtUsername;
    private JPasswordField pwdPassword;
    private JPanel panel;
    private JLabel loginBtn;
    private JPanel panel_1;
    private JPanel panel_2;
    private JLabel message;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
            EventQueue.invokeLater(new Runnable() {
```

```
                        public void run() {
                                try {
                                        LoginForm frame = new LoginForm();
                                        frame.setVisible(true);
                                } catch (Exception e) {
                                        e.printStackTrace();
                                }
                        }
                });
        }

        /**
         * Create the frame.
         */
        public LoginForm() throws NullPointerException {
                setBackground(Color.WHITE);
                setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                setBounds(100, 100, 450, 606);
                contentPane = new JPanel();
                contentPane.setBorder(new LineBorder(SystemColor.textHighlight, 2, true));
                contentPane.setForeground(new Color(255, 255, 255));
                contentPane.setBackground(new Color(135, 206, 250));
                setContentPane(contentPane);
                contentPane.setLayout(null);

                JLabel closeBtn = new JLabel("X");
                closeBtn.setHorizontalAlignment(SwingConstants.CENTER);

                closeBtn.setFont(new Font("Tahoma", Font.BOLD, 20));

                closeBtn.setForeground(new Color(255, 255, 255));

                closeBtn.setBounds(398, 13, 40, 40);

                closeBtn.addMouseListener(new MouseAdapter() {

                        public void mouseClicked(MouseEvent arg0) {

                                LoginForm.this.dispose();

                        }

                        public void mouseEntered(MouseEvent arg0) {

                                closeBtn.setForeground(Color.RED);

                        }

                        public void mouseExited(MouseEvent arg0) {

                                closeBtn.setForeground(Color.WHITE);

                        }

                });

                contentPane.add(closeBtn);

                JLabel lblNewLabel_1 = new JLabel("");

                lblNewLabel_1.setHorizontalAlignment(SwingConstants.CENTER);

                lblNewLabel_1.setIcon(new ImageIcon(new
        ImageIcon(LoginForm.class.getResource("/res/user-male.png")).getImage().getScaledInstance(
        200, 200, Image.SCALE_SMOOTH)));

                lblNewLabel_1.setBounds(130, 96, 200, 200);

                contentPane.add(lblNewLabel_1);

                panel = new JPanel();

                panel.setBorder(new LineBorder(new Color(30, 144, 255), 2, true));

                panel.setBackground(SystemColor.textHighlight);

                panel.setBounds(130, 441, 200, 40);
                contentPane.add(panel);
                panel.setLayout(null);

                loginBtn = new JLabel("Login");
                loginBtn.setForeground(new Color(255, 255, 255));
                loginBtn.setFont(new Font("Tahoma", Font.BOLD, 18));
                loginBtn.setHorizontalAlignment(SwingConstants.CENTER);
                loginBtn.setBounds(0, 0, 200, 40);
                loginBtn.addMouseListener(new MouseAdapter() {
                        public void mouseClicked(MouseEvent arg0) {
                                if(txtUsername.getText().equals("admin") &&
        pwdPassword.getText().equals("admin")) {
                                        message.setText("Login successful");
                                        HomePage hm=new HomePage();
                                hm.setVisible(true);
                                        LoginForm.this.dispose();
                                }else {
                                        message.setText("Login failed");
                                }
                        }
        });
                panel.add(loginBtn);

                panel_1 = new JPanel();
                panel_1.setLayout(null);
                panel_1.setBorder(new LineBorder(new Color(30, 144, 255), 2, true));
                panel_1.setBackground(SystemColor.textHighlight);
```

```
panel_1.setBounds(130, 323, 200, 40);
contentPane.add(panel_1);
txtUsername = new JTextField();
txtUsername.setBorder(null);
txtUsername.setBounds(1, 1, 198, 38);
panel_1.add(txtUsername);
txtUsername.setText("Username");
txtUsername.setHorizontalAlignment(SwingConstants.CENTER);
txtUsername.setColumns(10);
txtUsername.addFocusListener(new FocusAdapter() {
        public void focusGained(FocusEvent arg0) {
                message.setText("");
                if(txtUsername.getText().equals("Username")) {
                        txtUsername.setText("");
                }else {
                        txtUsername.selectAll();
                }


        }
        @Override
        public void focusLost(FocusEvent e) {
                if(txtUsername.getText().equals("")) {
                        txtUsername.setText("Username");
                }
        }
});
```

```
panel_2 = new JPanel();
panel_2.setLayout(null);
panel_2.setBorder(new LineBorder(new Color(30, 144, 255), 2, true));
panel_2.setBackground(SystemColor.textHighlight);
panel_2.setBounds(130, 376, 200, 40);
contentPane.add(panel_2);

pwdPassword = new JPasswordField();
pwdPassword.setBorder(null);
pwdPassword.setBounds(1, 1, 198, 38);
panel_2.add(pwdPassword);
pwdPassword.setHorizontalAlignment(SwingConstants.CENTER);
pwdPassword.setText("Password");
pwdPassword.setEchoChar((char)0);

message = new JLabel("");
message.setForeground(new Color(255, 0, 0));
message.setBounds(130, 416, 200, 16);
contentPane.add(message);
pwdPassword.addFocusListener(new FocusAdapter() {
        public void focusGained(FocusEvent arg0) {
                message.setText("");

                if(pwdPassword.getText().equals("Password")) {
                        pwdPassword.setEchoChar('●');
                        pwdPassword.setText("");
```

```
                    }else {
                              pwdPassword.selectAll();
                    }
          }
          @Override
          public void focusLost(FocusEvent e) {
                    if(pwdPassword.getText().equals("")) {
                              pwdPassword.setText("Password");
                              pwdPassword.setEchoChar((char)0);
                    }
          }
     });
     setUndecorated(true);
  setLocationRelativeTo(null);
  }
}
```

HomePage.java

```
package com.vijay.apps;


import java.awt.BorderLayout;
import java.awt.EventQueue;
import java.awt.GraphicsDevice;
import java.awt.GraphicsEnvironment;
import java.awt.SystemColor;
import java.awt.Toolkit;
import java.util.ArrayList;
import java.util.List;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.text.SimpleDateFormat;
import javax.swing.ButtonGroup;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import java.awt.Color;
import java.awt.Dimension;

import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.border.LineBorder;
import javax.swing.JRadioButton;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.Font;
import java.awt.FlowLayout;
import java.awt.Frame;
import javax.swing.SwingConstants;
import javax.swing.Timer;
import javax.swing.border.MatteBorder;
import javax.swing.JButton;
```

```
public class HomePage extends JFrame {


    private JPanel contentPane;
private static List<Question> questions;
int currentQuestion;
private Timer timer;
private Dimension screenSize;
private long duration = 15000;
    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    HomePage frame = new HomePage();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
    */
   public HomePage()  throws NullPointerException,IndexOutOfBoundsException{


              questions=new ArrayList<Question>() ;
                        questions.add(new Question("How to kill an activity in Android?",3
,"finish()","finishActivity(int requestCode)","A & B" ,"kill()"));


                        questions.add(    new Question("What is android view group?",1 ,"Collection of
views and other child views","Base class of building blocks ","Layouts" ,"None of the Above ")
                                  );
                        questions.add(new Question("How many threads are there in asyncTask in
android?",1 ,"Only one" , "two","AsyncTask doesn't have tread" ,"None")
                                            );
                        questions.add(new Question("What is the use of content provider in android?",3
,"A - To send the data from an application to another application", "To store the data in a
database","C - To share the data between applications" ,"None of the Above "));
                        questions.add(new Question("How to get current location in android?",3 ,"Using
with GPRS", "Using location provider ","A & B" ,"SQlite"));


                        currentQuestion=0;


                         GraphicsEnvironment graphics =
                                         GraphicsEnvironment.getLocalGraphicsEnvironment();
                                         GraphicsDevice device = graphics.getDefaultScreenDevice();
                                         screenSize = Toolkit.getDefaultToolkit().getScreenSize();


                        this.setSize(screenSize.width, screenSize.height);
             contentPane = new JPanel();
             contentPane.setBackground(new Color(135, 206, 250));
             contentPane.setSize(screenSize.width, screenSize.height);
             setContentPane(contentPane);
             ButtonGroup btngrp=new ButtonGroup();
             contentPane.setLayout(null);
             JPanel panel = new JPanel();
             panel.setLocation(20, 10);
             panel.setSize(screenSize.width-40, screenSize.height/10);
             panel.setBorder(new LineBorder(new Color(0, 0, 139), 2, true));
             panel.setBackground(new Color(255, 255, 255));
             contentPane.add(panel);
             panel.setLayout(null);
             JLabel question = new JLabel("",JLabel.CENTER);
             question.setBounds(12, 7, 1856, 88);
             question.setFont(new Font("Segoe UI", Font.PLAIN, 40));
             question.setText(questions.get(currentQuestion).getQuestion());
             panel.add(question);
             JPanel options = new JPanel();
             options.setLocation(20, 128);
             options.setSize(screenSize.width-40, (screenSize.height/10)*7);
             options.setBorder(new LineBorder(new Color(0, 0, 139), 2, true));
             options.setBackground(new Color(255, 255, 255));
             contentPane.add(options);
             options.setLayout(null);
             JRadioButton op1 = new JRadioButton("");
             op1.setBounds(8, 9, 1864, 189);
             op1.setFont(new Font("Segoe UI", Font.PLAIN, 40));
```

```
op1.setBackground(new Color(255, 255, 255));
op1.setText(questions.get(currentQuestion).get1());
options.add(op1);

JRadioButton op2 = new JRadioButton("");
op2.setBounds(8, 170, 1864, 189);
op2.setFont(new Font("Segoe UI", Font.PLAIN, 40));
op2.setBackground(Color.WHITE);
op2.setText(questions.get(currentQuestion).get2());
options.add(op2);

JRadioButton op3 = new JRadioButton("");
op3.setBounds(8, 364, 1864, 189);
op3.setFont(new Font("Segoe UI", Font.PLAIN, 40));
op3.setBackground(Color.WHITE);
op3.setText(questions.get(currentQuestion).get3());
options.add(op3);

JRadioButton op4 = new JRadioButton("");
op4.setBounds(8, 558, 1864, 189);
op4.setFont(new Font("Segoe UI", Font.PLAIN, 40));
op4.setBackground(Color.WHITE);
op4.setText(questions.get(currentQuestion).get4());
options.add(op4);

btngrp.add(op1);
btngrp.add(op2);
btngrp.add(op3);
btngrp.add(op4);

JPanel panel_2 = new JPanel();
panel_2.setForeground(new Color(255, 255, 255));
panel_2.setLocation(1640, 894);
panel_2.setSize(300-40, 100);
panel_2.setBorder(new MatteBorder(1, 4, 4, 1, (Color) new Color(0, 0, 0)));
panel_2.setBackground(SystemColor.textHighlight);
panel_2.addMouseListener(new MouseAdapter() {
	@Override
	public void mouseClicked(MouseEvent e) {
		if(op1.isSelected()) {
			questions.get(currentQuestion).setSelectedAnswer(1);

		}else if(op2.isSelected()) {
			questions.get(currentQuestion).setSelectedAnswer(2);
		}else if (op3.isSelected()) {
			questions.get(currentQuestion).setSelectedAnswer(3);
		}else if(op4.isSelected()) {
			questions.get(currentQuestion).setSelectedAnswer(4);
		}
		currentQuestion++;
		btngrp.clearSelection();
		if(currentQuestion==questions.size()) {
			timer.stop();
```

```
                                        showResult();
return;

                              }
                              question.setText(questions.get(currentQuestion).getQuestion());
                              op1.setText(questions.get(currentQuestion).get1());
                              op2.setText(questions.get(currentQuestion).get2());
                              op3.setText(questions.get(currentQuestion).get3());
                              op4.setText(questions.get(currentQuestion).get4());
                    }
                    @Override
                    public void mouseEntered(MouseEvent arg0) {
                              panel_2.setBackground(Color.BLACK);
                    }
                    @Override
                    public void mouseExited(MouseEvent e) {
                              panel_2.setBackground(SystemColor.textHighlight);
                    }
                    @Override
                    public void mousePressed(MouseEvent e) {
                              panel_2.setBackground(Color.GRAY);
                    }
                    @Override
                    public void mouseReleased(MouseEvent e) {
                              panel_2.setBackground(SystemColor.textHighlight);
```

```
            }
    });
    contentPane.add(panel_2);
    panel_2.setLayout(null);
    JLabel next = new JLabel("Next",JLabel.CENTER);
    next.setBounds(12, 7, 236, 80);
    next.setForeground(new Color(255, 255, 255));
    next.setFont(new Font("Tahoma", Font.BOLD, 45));
    panel_2.add(next);
   JPanel panel_1 = new JPanel();
   panel_1.setBorder(new MatteBorder(1, 4, 4, 1, (Color) new Color(0, 0, 0)));
   panel_1.setBackground(Color.YELLOW);
   panel_1.setBounds(20, 901, 260, 100);
   contentPane.add(panel_1);
   panel_1.setLayout(null);

   JLabel lblNewLabel = new JLabel("New label");
   lblNewLabel.setFont(new Font("Calibri Light", Font.PLAIN, 40));
   lblNewLabel.setBounds(12, 13, 236, 74);
   panel_1.add(lblNewLabel);
   lblNewLabel.setBackground(Color.YELLOW);
   lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
   lblNewLabel.setText(duration/1000+" seconds");
   timer = new Timer(1000, new ActionListener() {
   @Override
   public void actionPerformed(ActionEvent e) {
```

```
                    duration-=1000;
                  lblNewLabel.setText(duration/1000+" seconds");
                  if(duration==0) {
                    showResult();
                  }
          }
          });
          if (!timer.isRunning()) {
          timer.start();
          }
          setUndecorated(true);
           setResizable(false);
    }
    void showResult(){
           int c=0,w=0,t=0,u=0;
           for(Question q : questions) {
                    t++;
                    if(q.getResult()==1)
                             c++;
                    else if (q.getResult()==0)
                             w++;
                    else
                             u++;
           }

           setContentPane(panel(c,w,u ));
```

```
        repaint();
        revalidate();
    }
    public JPanel panel(int c,int w,int u) {

        JPanel panel = new JPanel();
        panel.setBounds(0, 0, 444, 271);
        panel.setBackground(new Color(255, 255, 255));
        getContentPane().add(panel);
        panel.setSize(screenSize.width, screenSize.height);
        panel.setLayout(null);

        JLabel correct = new JLabel();
        correct.setHorizontalAlignment(SwingConstants.CENTER);
        correct.setVerticalAlignment(SwingConstants.TOP);
        correct.setFont(new Font("Segoe UI", Font.PLAIN, 45));
        correct.setBounds(0, 300, 1920, 97);
        correct.setText("Correct : "+c);
        panel.add(correct);

        JLabel wrong = new JLabel();
        wrong.setVerticalAlignment(SwingConstants.TOP);
        wrong.setHorizontalAlignment(SwingConstants.CENTER);
        wrong.setFont(new Font("Segoe UI", Font.PLAIN, 45));
        wrong.setBounds(0, 422, 1920, 97);
        wrong.setText("Wrong : "+w);
```

```
panel.add(wrong);

JLabel unanswered = new JLabel();
unanswered.setVerticalAlignment(SwingConstants.TOP);
unanswered.setHorizontalAlignment(SwingConstants.CENTER);
unanswered.setFont(new Font("Segoe UI", Font.PLAIN, 45));
unanswered.setBounds(0, 532, 1920, 97);
unanswered.setText("Unanswered : "+u);
panel.add(unanswered);

JButton btnNewButton = new JButton("Exit");
btnNewButton.setBackground(new Color(255, 140, 0));
btnNewButton.setFont(new Font("Segoe UI", Font.PLAIN, 45));
btnNewButton.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {
                HomePage.this.dispose();
        }
});
btnNewButton.setBounds((screenSize.width/2)-150, 666, 300, 100);
panel.add(btnNewButton);

JLabel lblNewLabel_1 = new JLabel("Thank You !!");
lblNewLabel_1.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_1.setFont(new Font("Script MT Bold", Font.PLAIN, 85));
lblNewLabel_1.setBounds(658, 13, 618, 274);
```

```
        panel.add(lblNewLabel_1);

        return panel;

}
}
```

Output

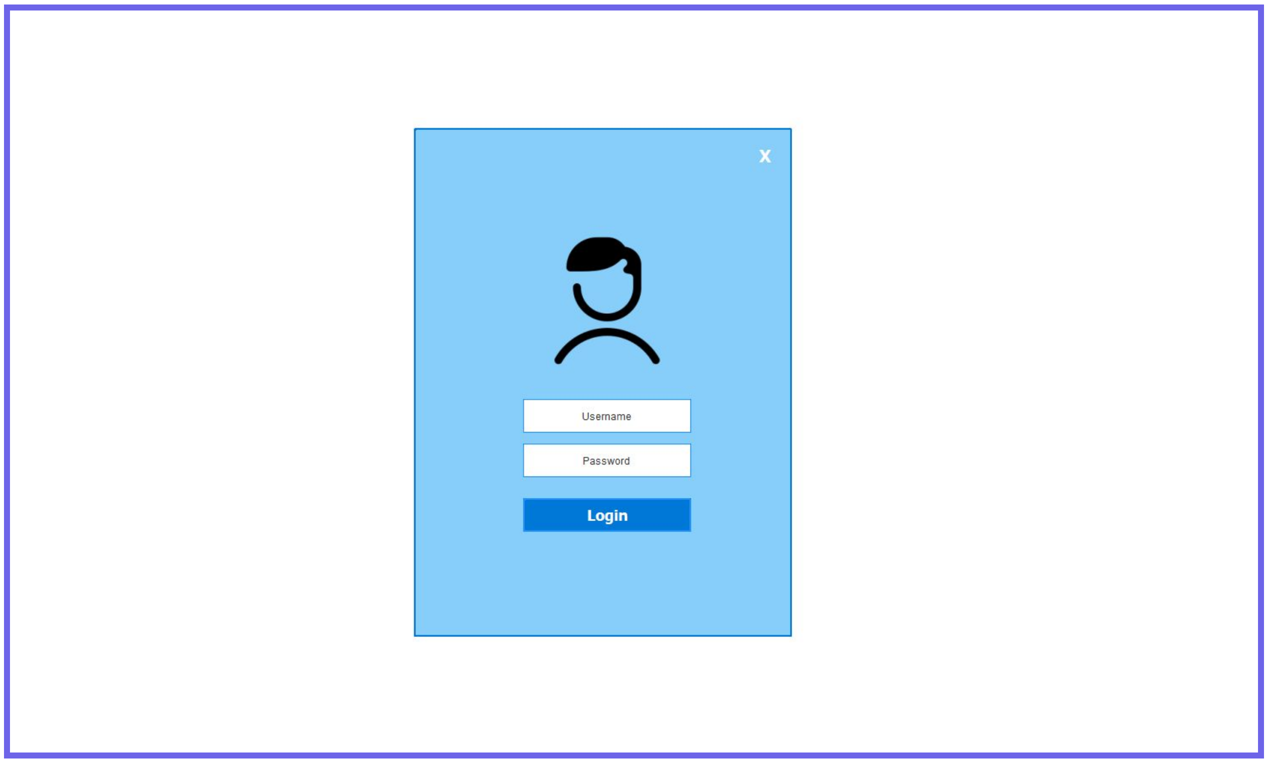
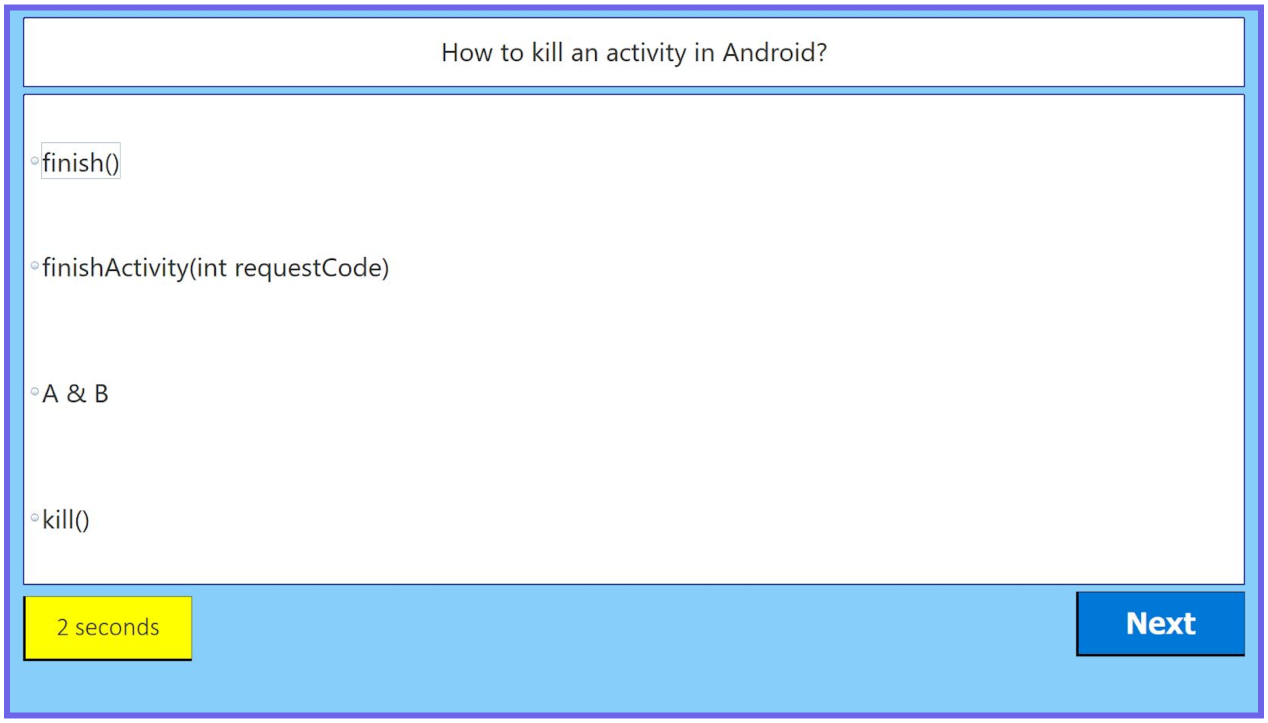


Fig 1 : Login Screen


How to kill an activity in Android?

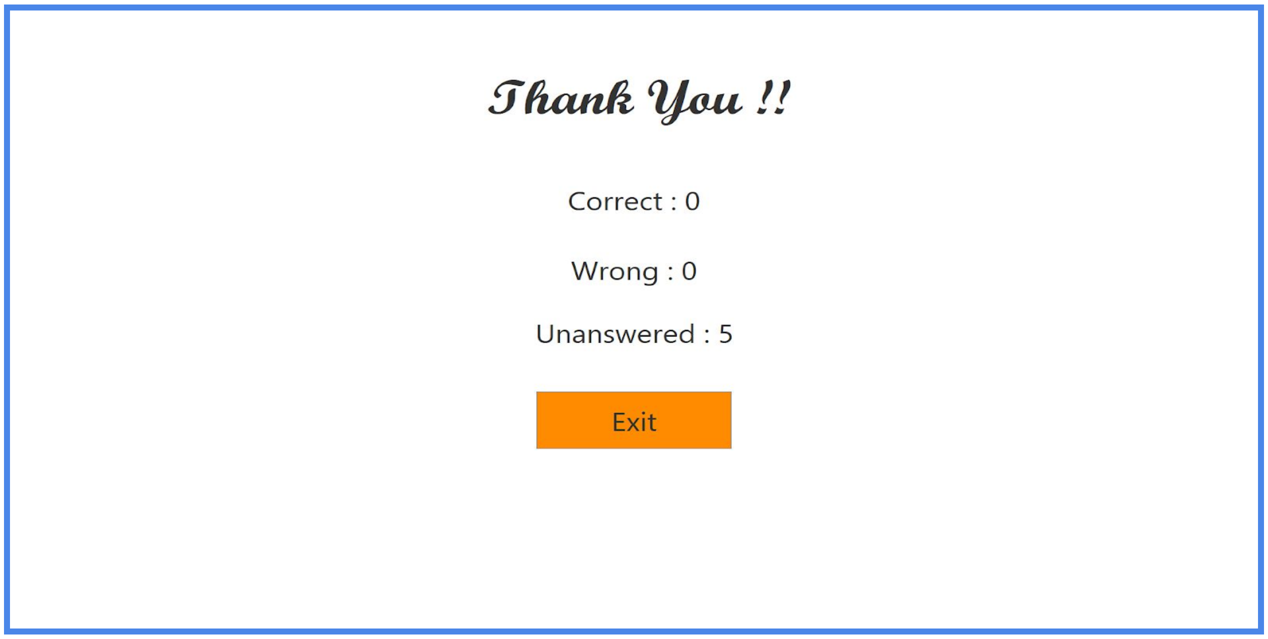
finish()

finishActivity(int requestCode)

A & B

kill()

Fig 2 : Home Screen


Thank You !!

Correct : 0

Wrong : 0

Unanswered : 5

Fig 3 : Result Screen

Test Plan

Introduction

This application is used to take time based quiz/exam/test on any topic and show summary of responses given. Participants are required to provide valid credentials in order to start the test. Switching from the app is not allowed during the test.

Intended Audience

This test plan is made for system testing of this application. The test plan will be referred by

- Development manager and development team
- Customers (schools, colleges and recruiting companies)
- Senior management

Intended Users

The application will be used by candidates of the exam who are allowed to do so. The candidates may include students, in case of schools and colleges , employees in case of companies.

Test Scope

- Unit testing
- Integration testing
- Regression testing
- Alpha testing
- User interface testing
- Functionality testing.

Out of Scope

The following types of testing are out of scope for this application

- Beta testing

- Smoke testing
- Recovery testing
- Security testing
- Stress testing
- Performance testing
- Usability testing

Test Objectives (Targeted defects, test cases, scenarios)

- Targeted number of defects : 500
- Targeted number of test scenario : 100
- Targeted number of test cases : 2000
- Number of iterations required : 5
- Test scenario writing per day : 10
- Test case writing per day : 50
- Test case execution per day : 100
- Coverage of requirements
  - P1 = 100%
  - P2 = 50%
  - P3 = 10%

Assumptions

Following assumptions are made during testing:

- Users know how to use computers and understand english language.
- JDK/JRE is present on the computer
- Users are right handed
- Testing is done on Windows 10 OS, 4GB , 1TB, i3 2GHz intel processor ,Radeon graphics. Any other machine capable of running java programs is acceptable.
- User has the credentials to login
- Users are aware of objective type exams,timers and the next button to move onto the next question.

Risk analysis

The following threats are identified to divert the functioning of the application:

- Alt + F4 can close the window
- Alt + TAB can switch between screens
- Windows Home button may show the taskbar.
- Abrupt power supply may terminate the application.
- Users may use any other device to search the answers.
- Users may not understand english
- Left handed users may find it uncomfortable to use.

Workflow

The project manager communicates the availability of new builds along with the records of reviews and unit testing to the test manager.The application is installed on the machine as defined in prerequisites. Test manager checks the entry criteria for testing. If the application passes the entry criteria, it is taken for iteration testing. Test cases are stored in the configuration library. Test cases are executed as per the sequence mentioned and results are added in the test case as actual results. Difference between expected result and actual result is taken for evaluation. All differences between expected results and actual results are reproduced. Video clips are created for the defects. Defects are added in defect management tools.

Test Design

The application will be tested as per steps described in Table A.1. Verification activities will be covered in the quality plan. This is a system test plan and other levels of testing are covered by separate plans at each level.

Table A.1 Levels of testing

Testing level	Responsibility
Unit testing	Developers
Integration testing	Developers
System testing	Testers

Test Data Definition

Test data is defined using the following techniques

- Equivalence partitioning
- Error guessing

Roles and Responsibilities

Table A.2 Roles and responsibilities

Responsibility/ Role	Test manager	Senior tester	Tester	Customer
Test planning	X			
Test scenario writing		X		
Test case writing		X	X	
Test data definition			X	
Test execution			X	
Query resolution				X

Test Schedule and Resources

Assuming that the project starts on 4th September 2020.

1. Test planning—Sep 4th, 2020
2. Test scenario writing, review and updation—Sep 4th, 2020
3. Test case writing, review and updation—Sep 5th, 2020
4. Test case execution 1st iteration—Sep 5th, 2020
5. Test case execution 2nd iteration—Sep 6th, 2020

6. Test execution 3rd iteration—Sep 7th, 2020

Test Data Management

Test plan, test scenario and test cases will be under configuration management control and kept on centralised server. Test reports will be stored on the same server. Backup will be taken on tape, once per week.

Test Environment

- Windows 10 OS
- 4GB RAM 1TB Hard Disk
- I3 intel processor 2GHz
- JDK/JRE installed

Test Entry, Exit, Suspension, Resumption Criteria

Entry

- All review comments are closed
- All unit testing defects are closed
- Application can be installed and launched

Exit

Test cases are completed and all defects found in testing are closed, retested and found to be closed.

Suspension

Tests will be suspended in the following circumstances,

- Test cases written do not find sufficient number of defects
- Test case writing and test data definition are not completed
- Application cannot be installed
- Defects found in 1st/2nd/3rd iteration are more than 500

Resumption

- All defects are reviewed and corrective/preventive actions are taken

- All review comments are closed
- All unit testing defects are fixed, restested and found to be closed
- Application can be installed

Test Scenario

Common for all scenarios :

The application can be launched by running the LoginForm java class which extends the JFrame class of swing package. A window of 450X606 opens with a close button on the top right corner, an icon of male user, a textbox to enter username, a textbox to enter password, label to print error message and a button to login. The button is 40X40 in size. Icon is 200X200.The textboxes and login button is 200X40 with horizontal center alignment with a blue border of 2dp. The cursor is by default in the username field and can be switched to the password field using the TAB key on the keyboard. If some text is already present in the textboxes then it is automatically selected on gaining focus. Hovering over the close button turns it into red.

Scenario 1: Positive scenario where login is successful

Cursor is present in the username field on application launch. User enters “admin”.Press TAB key. Cursor moves to the password field and the user enters “admin”. Click on the login button and a new window with loaded questions opens in full screen mode.

Scenario 2: Negative scenario where either username or password is empty

There can be two combinations where equivalence partitioning can be applied. One of the two fields is taken as empty each time which may represent equivalence class

Scenario 3: Negative scenario where both username and password is empty

Scenario 4: Negative scenario where either username or password is wrong

There can be two combinations where equivalence partitioning can be applied. One of the two fields is taken as wrong each time which may represent equivalence class

Scenario 5: Negative scenario both username and password is wrong

Scenario 6: Positive scenario close button terminates the application

Common for all scenarios :

After successful login, the test screen with preloaded questions and options is launched in full screen mode. It is a JFrame unit with resolution equal to screen resolution(1920X1080). The default icon, app name, minimize button,resize button and close buttons are not visible. The taskbar is also hidden. There is a panel with a label with question writing in it with a padding of 10dp from all four sides and a blue border of 2dp. Below it is a panel with the same padding and border containing four radio buttons packed in a button group to show all four options. At the left bottom corner there is a timer with yellow color. At the right bottom corner, there is a next button with blue background and white caption with shadow effect. The next button turns black on hovering.User may or may not choose to select any radio button Clicking on the next button will refresh the screen with the next question and corresponding options. Timer decreases itself every second.Either the time is up or all questions are attempted, the frame is repainted with a new panel containing a label with “Thank you” text in cursive handwriting and correct, wrong and unanswered questions with their count next to it. Below everything there is an exit button in orange color. On clicking it turns grey and terminates the application.Everything is horizontally centered.

Scenario 1: Positive scenario where user selects an option and clicks the next button. Question is updated.

By default no option is selected and the user may select one of the options and click the next button. The next question is visible.

Scenario 2: Negative scenario where timer is not working

Scenario 3: Negative scenario where next button is not working

Scenario 4: Negative scenario where ALT+F4 or ALT+TAB or Home key is pressed

Application is terminated or screens are switched.

Scenario 5: Negative scenario where selected option is not recorded

There can be four combinations where equivalence partitioning can be applied. One of the four options is selected each time which may represent equivalence class

Scenario 6: Negative scenario where correct option is selected but result is marked as wrong

There can be four combinations where equivalence partitioning can be applied. One of the four options is selected each time which may represent equivalence class

Scenario 7: Negative scenario where wrong option is selected but result is marked as correct

There can be four combinations where equivalence partitioning can be applied. One of the four options is selected each time which may represent equivalence class

Scenario 8: Positive scenario where wrong option is selected and result is marked as wrong

There can be four combinations where equivalence partitioning can be applied. One of the four options is selected each time which may represent equivalence class

Scenario 9: Positive scenario where correct option is selected and result is marked as correct

There can be four combinations where equivalence partitioning can be applied. One of the four options is selected each time which may represent equivalence class

Scenario 10: Negative scenario where repainting the result panel may overlap with existing components

Scenario 11: Negative scenario where all counts of correct,wrong and unanswered responses is wrong

There can be one combination where equivalence partitioning can be applied.

Scenario 12: Negative scenario where count of any one(correct,wrong and unanswered responses) is wrong

There can be three combinations where equivalence partitioning can be applied.

Scenario 13: Negative scenario where count of any two(correct,wrong and unanswered responses) is wrong

There can be three combinations where equivalence partitioning can be applied.

Scenario 14: Positive scenario where count of correct,wrong and unanswered responses is correct

There can be one combination where equivalence partitioning can be applied.

Scenario 15: Positive scenario where exit button terminates the app.

Test Cases

Test Case 1

- Test Precondition : application must be installed on computer and must be working
- Type Of Testing : Smoke testing
- Valid/invalid condition : valid condition
- Objective : to check whether the application is working or not after starting it by double clicking.
- Expected Results : application started and login screen is shown.

Test Case 2

- Test Precondition : username and password is entered correctly and clicked on the login button.
- Type Of Testing : Functional testing
- Valid/invalid condition : valid condition
- Objective : user should be redirected to the home page to take the test.
- Expected Results : user is redirected to homepage and is able to give the test.

Test Case 3

- Test Precondition : Wrong details are entered on the login page and clicked on the login button.
- Type Of Testing : Functional testing

- Valid/invalid condition : invalid condition
- Objective : only users with correct details can log in.
- Expected Results : user should be able to see an error message

Test Case 4

- Test Precondition : Details(username & password) are not entered on the login page and clicked on the login button.
- Type Of Testing : Functional testing
- Valid/invalid condition : invalid condition
- Objective : only users with correct details can log in (empty should be invalid).
- Expected Results : user should be able to see an error message

Test Case 5

- Test Precondition : all questions are attempted
- Type Of Testing : Functional testing
- Valid/invalid condition : valid condition
- Objective : result screen is displayed.
- Expected Results : to display the result to the user.

Test Case 6

- Test Precondition : user selects an option and clicks the next button. Question is updated.
- Type Of Testing : Functional testing
- Valid/invalid condition : valid condition
- Objective : to display the next question.
- Expected Results : next question will be displayed on screen.

Test Case 7

- Test Precondition : user selects an option and clicks the next button. Question is updated.
- Type Of Testing : Functional testing
- Valid/invalid condition : valid condition
- Objective : to display the next question.
- Expected Results : next question will be displayed on screen.

Test Case 8

- Test Precondition : ALT+F4 or ALT+TAB or Home key is pressed by the user while giving the test.
- Type Of Testing : Functional testing
- Valid/invalid condition : invalid condition
- Objective : to terminate the test
- Expected Results : Application is terminated or screens are switched.

Test Case 9

- Test Precondition : User clicks the exit button on result screen
- Type Of Testing : Functional testing
- Valid/invalid condition : valid condition
- Objective : to terminate the application
- Expected Results : Application is terminated.

Test Case 10

- Test Precondition : User clicks the close button on login screen
- Type Of Testing : Functional testing
- Valid/invalid condition : valid condition
- Objective : to terminate the application
- Expected Results : Application is terminated.

Junit Testing

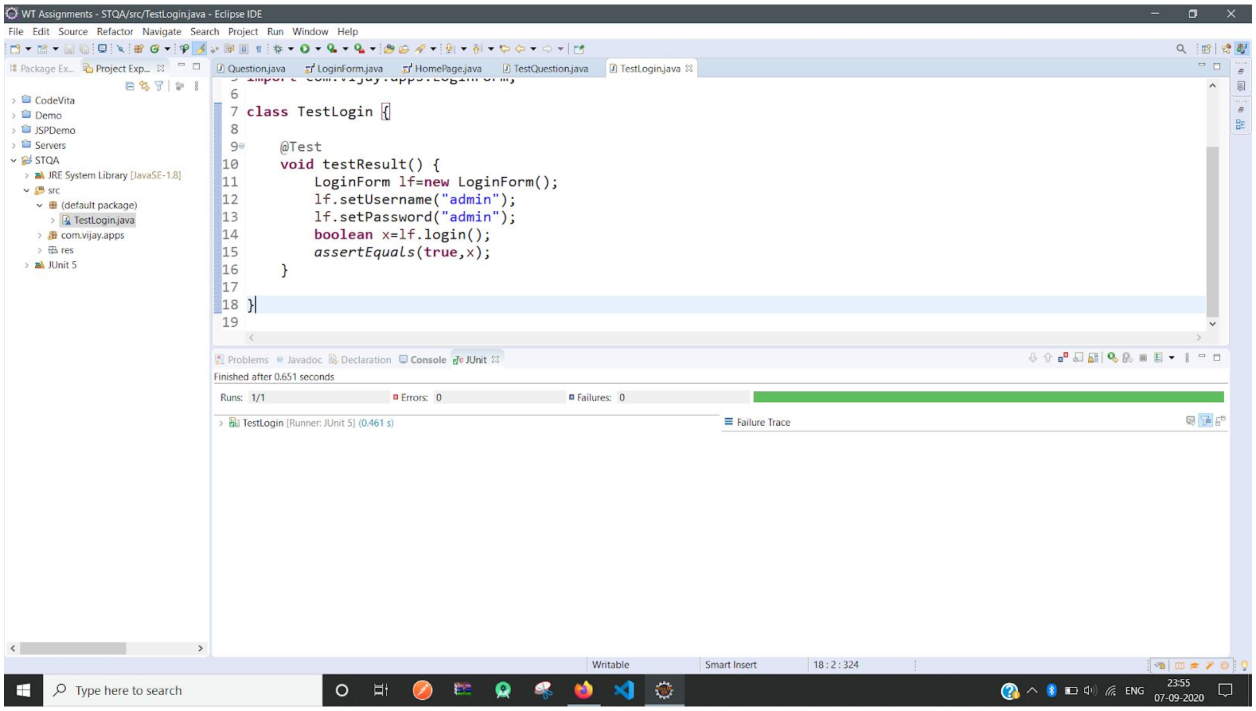
TestQuestion.java

```
package com.vijay.apps;
import static org.junit.jupiter.api.Assertions.*;
import org.junit.jupiter.api.Test;
class TestQuestion {
   @Test
   void testResult() {
          Question q=new Question("How to kill an activity in Android?",3
                            ,"finish()","finishActivity(int requestCode)","A & B" ,"kill()");
          q.setSelectedAnswer(3);
          assertEquals(1, q.getResult());
   }
}
```

TestLogin.java

```
import static org.junit.jupiter.api.Assertions.*;
import org.junit.jupiter.api.Test;
import com.vijay.apps.LoginForm;
class TestLogin {
   @Test
   void testResult() {
          LoginForm lf=new LoginForm();
          lf.setUsername("admin");
          lf.setPassword("admin");
          boolean x=lf.login();
```

```
            assertEquals(true,x);
    }
}
```



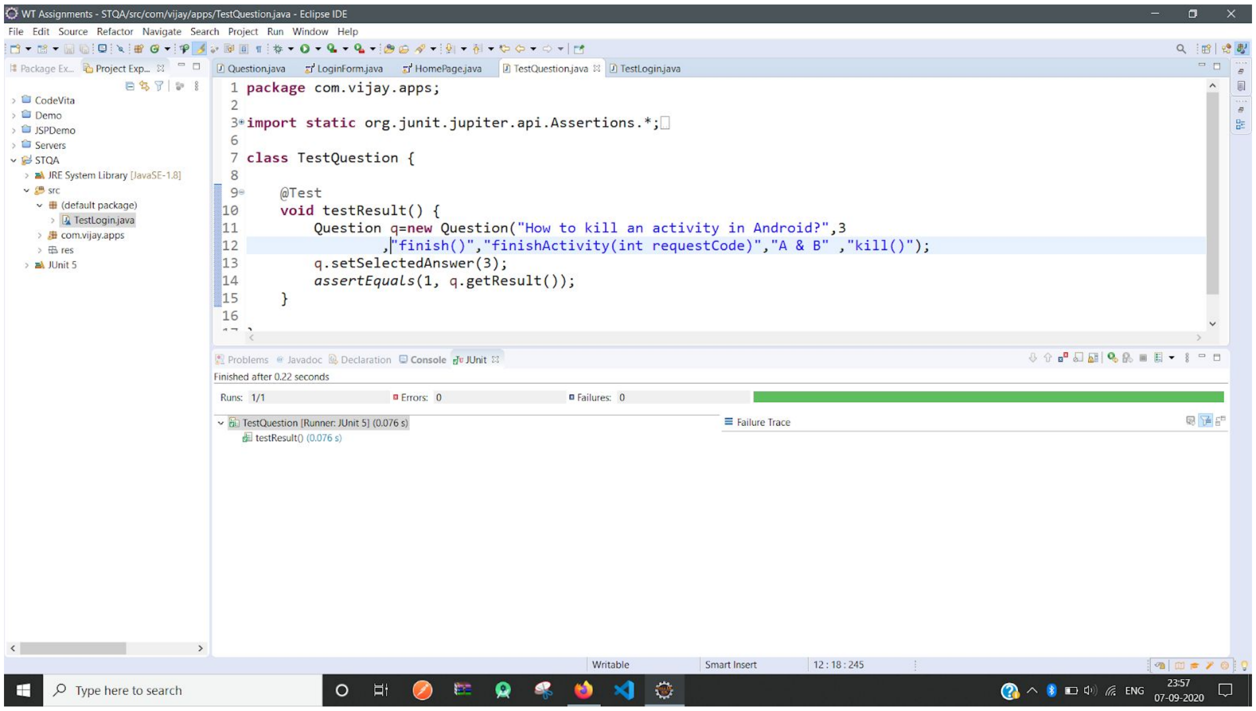
WT Assignments - STQA/src/TestLogin.java - Eclipse IDE

```java
class TestLogin {

    @Test
    void testResult() {
        LoginForm lf=new LoginForm();
        lf.setUsername("admin");
        lf.setPassword("admin");
        boolean x=lf.login();
        assertEquals(true,x);
    }

}
```

```
Finished after 0.651 seconds
Runs: 1/1    Errors: 0    Failures: 0
TestLogin [Runner: JUnit 5] (0.461 s)
```



WT Assignments - STQA/src/com/vijay/apps/TestQuestion.java - Eclipse IDE

```java
package com.vijay.apps;

import static org.junit.jupiter.api.Assertions.*;

class TestQuestion {

    @Test
    void testResult() {
        Question q=new Question("How to kill an activity in Android?",3
                ,"finish()","finishActivity(int requestCode)","A & B" ,"kill()");
        q.setSelectedAnswer(3);
        assertEquals(1, q.getResult());
    }

```

```
Finished after 0.22 seconds
Runs: 1/1    Errors: 0    Failures: 0
TestQuestion [Runner: JUnit 5] (0.076 s)
    testResult() (0.076 s)
```