Module Code & Module Title

CS4001NA Programming

Assessment Weightage &Type

50 % Individual Coursework

Year and Semester

2019-20 Autumn

Student Name: Samir Mishra

London Met ID: 19031857

College ID: NP05CP4A190145

Assignment Due Date: 05/06/2020

Assignment Submission Date: 05/06/2020

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Contents

Table of Figures:

Table of Tables:

Introduction

Java

Java is a high-level object-oriented programming language developed by James Gosling at Sun Microsystems. it is a platform independent programming language which can run in different platform without any problems. it can create software for multiple platforms. it is a very widely used programming language most commonly it is used in android and server sites. Java is designed to have a look and feel of the famous programming language C++, but it is easier than it and used to create complete applications that may run on a single computer or can be distributed for others. (Oracle, 2020)

Why is Java popular?

It is very difficult to provide single reason for this question because it has many characters such as:

Simple: Java syntax is like others old programming languages like C++, so it is easier to learn for programmers. If we understood the basic concept of OOP of java then it would be easy to master.

Object-Oriented: Java is an object-oriented programming language. Everything in java is an object. Java can be easily extended since it is based on the Object model. Some of the basic concept of OOP's are

- Object
- Class
- Inheritance
- Polymorphism
- Abstraction
- Encapsulation

Robust: Robust tries to detect error prone codes by checking compile time error, syntax error and runtime error. Java also has improved memory management and mishandled exceptions because it has automatic Garbage Collector and Exception Handling.

Platform Independent: Java is a platform independent language. Java is guaranteed to be written to be write once and run anywhere language. When we compile java code it is compiled into bytecode and this bytecode is platform independent, we can run it in more than one platform.

Secure: When it comes to security java stands in top position as the program written in java runs in java runtime environment with almost no interaction with Operating System. So, we can say that it is very secure.

Portable: Compiled bytecode of java program is easy to carry and run in many devices and Operating Systems. Hence it is very portable.

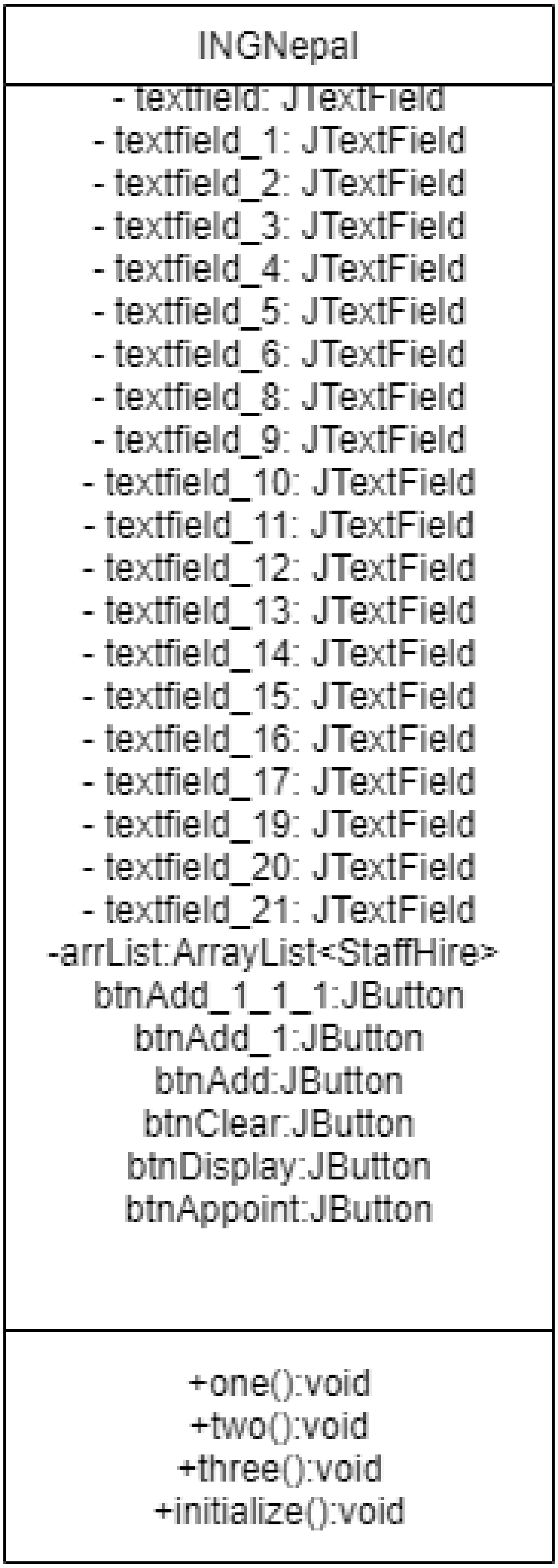
High Performance: As we know that java is an interpreted language the program created with java gives us very fast and high performance.

Java Swing

Java Swing is a graphical user interface (GUI) package which includes various widgets. This package is the part of Java Foundation Classes (JFC) and includes different packages which are needed to create a GUI like buttons, images, trees, sliders, toolbars etc. Swing package is also platform independent like java which can be created in one platform and run in different platforms. (Oracle, 2018)

Class Diagram

A class diagram is a static diagram which is used to represent the static view of an application and for the construction of executable code of that software or application. It describes the structure of that program showing its attributes, operators and relationship among objects.



INGNepal

- textfield: JTextField
- textfield_1: JTextField
- textfield_2: JTextField
- textfield_3: JTextField
- textfield_4: JTextField
- textfield_5: JTextField
- textfield_6: JTextField
- textfield_8: JTextField
- textfield_9: JTextField
- textfield_10: JTextField
- textfield_11: JTextField
- textfield_12: JTextField
- textfield_13: JTextField
- textfield_14: JTextField
- textfield_15: JTextField
- textfield_16: JTextField
- textfield_17: JTextField
- textfield_19: JTextField
- textfield_20: JTextField
- textfield_21: JTextField

-arrList:ArrayList<StaffHire>
btnAdd_1_1_1:JButton
btnAdd_1:JButton
btnAdd:JButton
btnClear:JButton
btnDisplay:JButton
btnAppoint:JButton

+one():void
+two():void
+three():void
+initialize():void

Pseudocode

```
FUNCTION INGNepal():

   CALL initialize();

END


FUNCTION main():

   INSTANCE of INGNepal

END


FUNCTION one():

   CREATE JLabel "FullTimeStaffHire"

   SET bounds to 300, 12, 129, 15

   ADD label to frame


   CREATE JLabel "Vacancy number"

   SET bounds to 39, 79, 141, 24

   ADD label to frame


   CREATE JTextField

   SET bounds to 175, 82, 141, 19

   ADD textfield to frame


   CREATE JLabel "Job Type"
```

```
SET bounds to 39, 144, 70, 15
ADD label to frame

CREATE JTextField
SET bounds to 175, 142, 141, 19
ADD textfield to frame

CREATE JLabel "Designation"
SET bounds to 39, 197, 109, 15
ADD label to frame

CREATE JTextField
SET bounds to 175, 195, 141, 19
ADD textfield to frame

CREATE JLabel "Working Hour"
SET bounds to 39, 250, 129, 15
ADD label to frame

CREATE JTextField
SET bounds to 175, 248, 141, 19
ADD textfield to frame
```

```
CREATE JLabel "salary"

SET bounds to 39, 298, 70, 15

ADD label to frame


CREATE JTextField

SET bounds to 175, 296, 141, 19

ADD textfield to frame


CREATE a JButton "Add"

SET bounds to 39, 349, 277, 25

ADD button to frame

ADD actionListener to the button

    FUNCTION actionPerformed(ActionEvent e):

        ADDS the user data to list.



CREATE JLabel "staff name"

SET bounds to 39, 298, 70, 15

ADD label to frame


CREATE JTextField

SET bounds to 578, 82, 141, 19

ADD textfield to frame
```

```
CREATE JLabel "joining date"

SET bounds to 436, 144, 96, 15

ADD label to frame


CREATE JTextField

SET bounds to 578, 142, 141, 19

ADD textfield to frame


CREATE JLabel "Qualification"

SET bounds to 436, 250, 109, 15

ADD label to frame


CREATE JTextField

SET bounds to 578, 248, 141, 19

ADD textfield to frame


CREATE JLabel "appointed by"

SET bounds to 436, 298, 129, 15

ADD label to frame


CREATE JTextField

SET bounds to 578, 296, 141, 19
```

```
    ADD textfield to frame


    CREATE a JButton "Appoint"
    SET bounds to 436, 349, 283, 25
    ADD button to frame
    ADD actionListener to the button
        FUNCTION actionPerformed(ActionEvent e):
            SHOW a information if user is verified and appointed.



END


FUNCTION two():
    CREATE JLabel "part time staff hire"
    SET bounds to 300, 425, 134, 15
    ADD label to frame


    CREATE JLabel "vacancy number"
    SET bounds to 39, 485, 141, 24
    ADD label to frame


    CREATE JTextField
    SET bounds to 175, 488, 141, 19
```

```
ADD textfield to frame

CREATE JLabel "Job Type"
SET bounds to 39, 555, 70, 15
ADD label to frame

CREATE JTextField
SET bounds to 175, 553, 141, 19
ADD textfield to frame

CREATE JLabel "Designation"
SET bounds to 39, 608, 109, 15
ADD label to frame

CREATE JTextField
SET bounds to 175, 606, 141, 19
ADD textfield to frame

CREATE JLabel "Working Hour"
SET bounds to 39, 652, 129, 24
ADD label to frame

CREATE JTextField
```

```
SET bounds to 175, 655, 141, 19

ADD textfield to frame


CREATE JLabel "shifts"

SET bounds to 39, 708, 70, 15

ADD label to frame


CREATE JTextField

SET bounds to 175, 706, 141, 19

ADD textfield to frame


CREATE JLabel "wages per hour"

SET bounds to 39, 752, 129, 24

ADD label to frame


CREATE JTextField

SET bounds to 175, 757, 141, 19

ADD textfield to frame


CREATE JLabel "staff name"

SET bounds to 436, 490, 96, 15

ADD label to frame
```

```
CREATE JTextField

SET bounds to 578, 488, 141, 19

ADD textfield to frame


CREATE JLabel "joining date"

SET bounds to 436, 555, 96, 15

ADD label to frame


CREATE JTextField

SET bounds to 578, 553, 141, 19

ADD textfield to frame


CREATE JLabel "Qualification"

SET bounds to 436, 250, 109, 15

ADD label to frame


CREATE JTextField

SET bounds to 578, 655, 141, 19

ADD textfield to frame


CREATE JLabel "appointed by"

SET bounds to 436, 722, 129, 15

ADD label to frame
```

```
    CREATE JTextField

    SET bounds to 578, 720, 141, 19

    ADD textfield to frame


    CREATE a JButton "Add"

    SET bounds to 39, 809, 277, 25

    ADD button to frame

    ADD actionListener to the button

        FUNCTION actionPerformed(ActionEvent e):

            ADDS the user data to list.


    CREATE a JButton "Appoint"

    SET bounds to 436, 809, 283, 25

    ADD button to frame

    ADD actionListener to the button

        FUNCTION actionPerformed(ActionEvent e):

            SHOW a information if user is verified and appointed.


END


FUNCTION three():

    CREATE JLabel "staff Termination"
```

```
SET bounds to 300, 890, 134, 15

ADD label to frame


CREATE JLabel "vacancy number"

SET bounds to 175, 917, 141, 24

ADD label to frame


CREATE JTextField

SET bounds to 310, 920, 175, 19

ADD textfield to frame


CREATE a JButton "Terminate"

SET bounds to 175, 963, 313, 25

ADD button to frame

ADD actionListener to the button

    FUNCTION actionPerformed(ActionEvent e):

        TERMINATE a part time staff


CREATE a JButton "display"

SET bounds to 199, 1000, 117, 25

ADD button to frame

ADD actionListener to the button

    FUNCTION actionPerformed(ActionEvent e):
            PRINTS all the staffs


    CREATE a JButton "clear"

    SET bounds to 339, 1000, 117, 25

    ADD button to frame

    ADD actionListener to the button

        FUNCTION actionPerformed(ActionEvent e):

            CLEAR the form


END


FUNCTION initialize():

    CREATE JFRAME "Hiring System"

    SET size bound 100, 100, 769, 1116

    SET default close operation to EXIT_ON_CLOSE

    SET null layout

    CALL one();

    CALL two();

    CALL three();

END
```

Method Description

INGNepal

- one(): This method contains all the GUI components for adding and appointing full time staff.

- two(): This method contains all the GUI components for adding and appointing part time staff.

- three(): This method contains GUI components for terminating staff, clearing the form and displaying all the staffs.

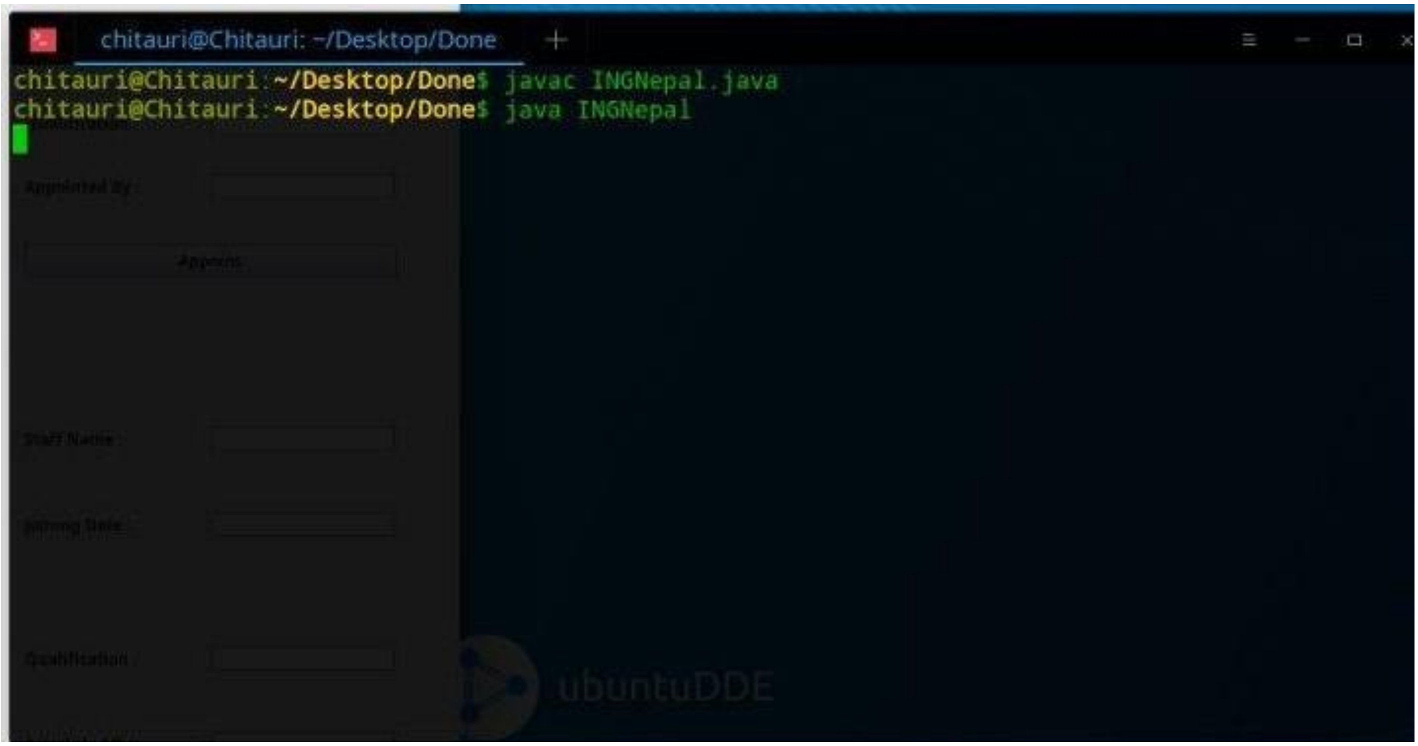
- Initialize(): In this method the name of the application is set, width and height of the program is set, operation on clicking the close button is set, the layout of the application is set and method one(), two() and three() is called.

Testing

Test 1:

Table 1: opening GUI using command

Objective	To open GUI using terminal.
Action	Typed javac INGNepal.java to compile and typed java INGNepal to open GUI.
Excepted result	File must be compiled and GUI must be opened.
Actual result	Files were compiled and GUI was opened.
Conclusion	Test successful.



```
chitauri@Chitauri:~/Desktop/Done$ javac INGNepal.java
chitauri@Chitauri:~/Desktop/Done$ java INGNepal
```

Figure 1: Test 1

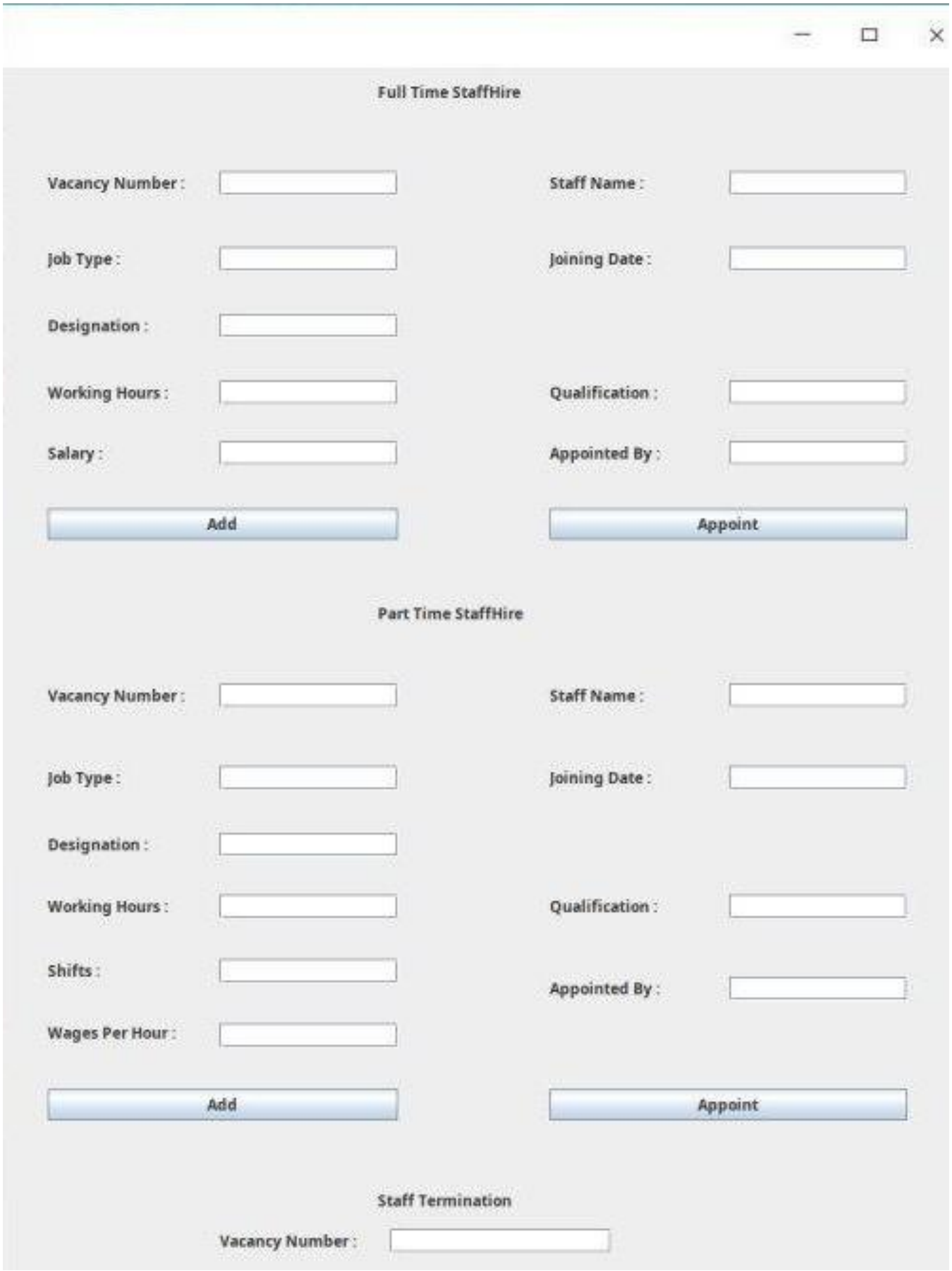
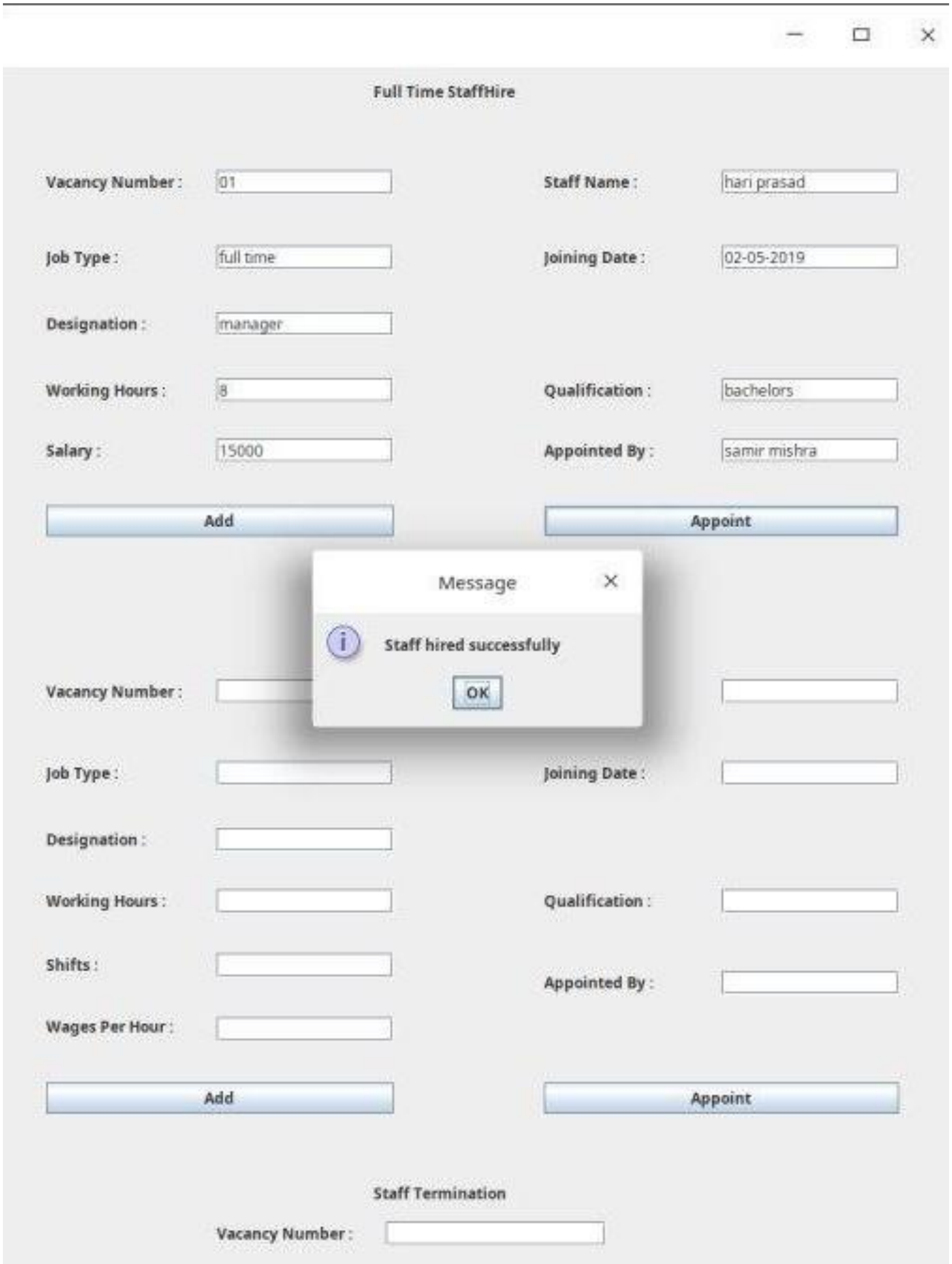


Figure 2: Test 1

Table 2: Appointing staff using GUI

Objective	To hire a staff using GUI.
Action	Filled form and appoint button was clicked.
Excepted result	Staff should be hired and a message should be shown.
Actual result	Staff was hired and a message was displayed.
Conclusion	Test successful.

Test 2:



Full Time StaffHire

Vacancy Number : 01

Staff Name : hari prasad

Job Type : full time

Joining Date : 02-05-2019

Designation : manager

Working Hours : 8

Qualification : bachelors

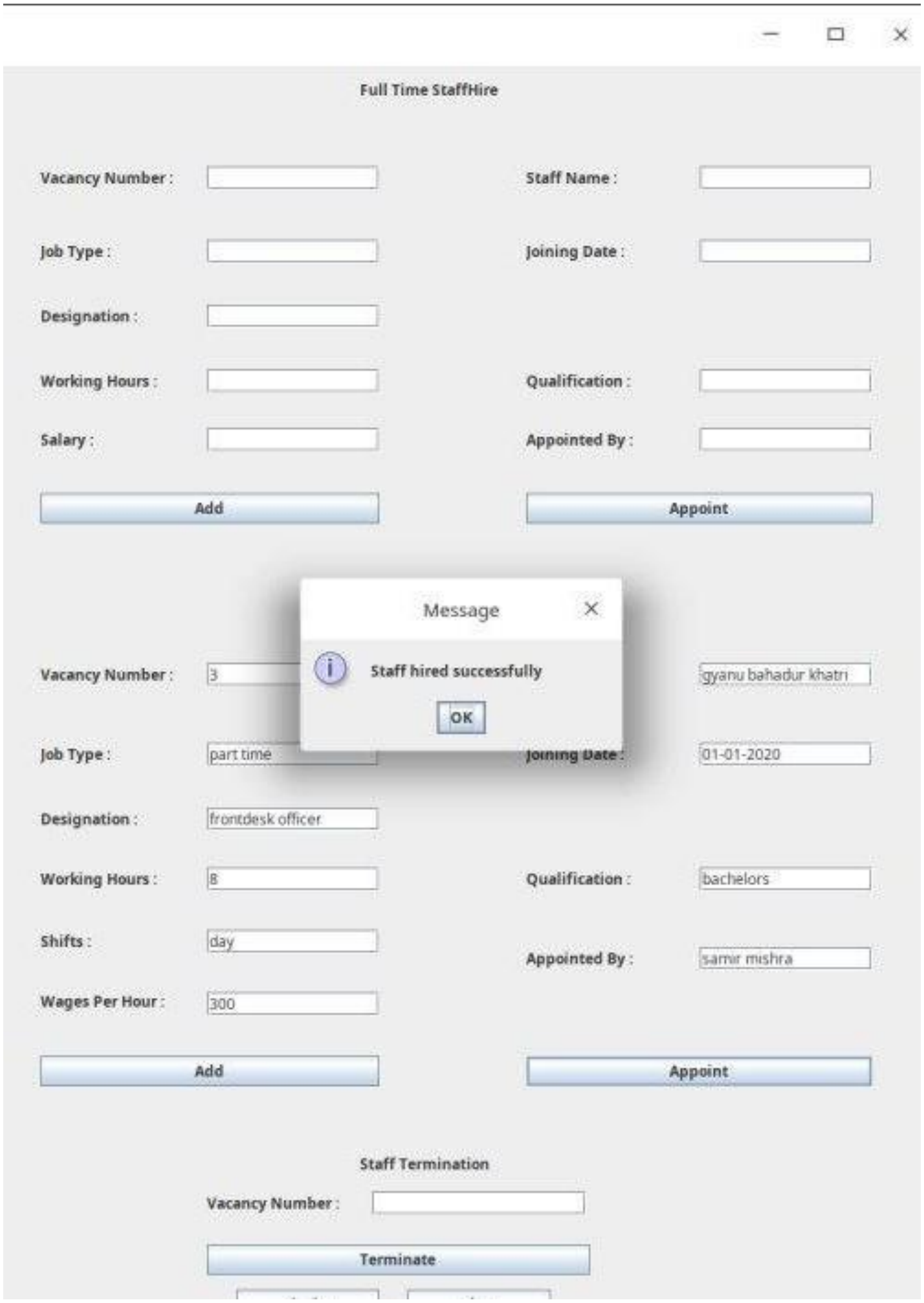
Salary : 15000

Appointed By : samir mishra

Message

Staff hired successfully

Figure 3 : Test 2



Staff hired successfully

Vacancy Number : 3

gyanu bahadur khatri

Job Type : part time

Joining Date : 01-01-2020

Designation : frontdesk officer

Working Hours : 8

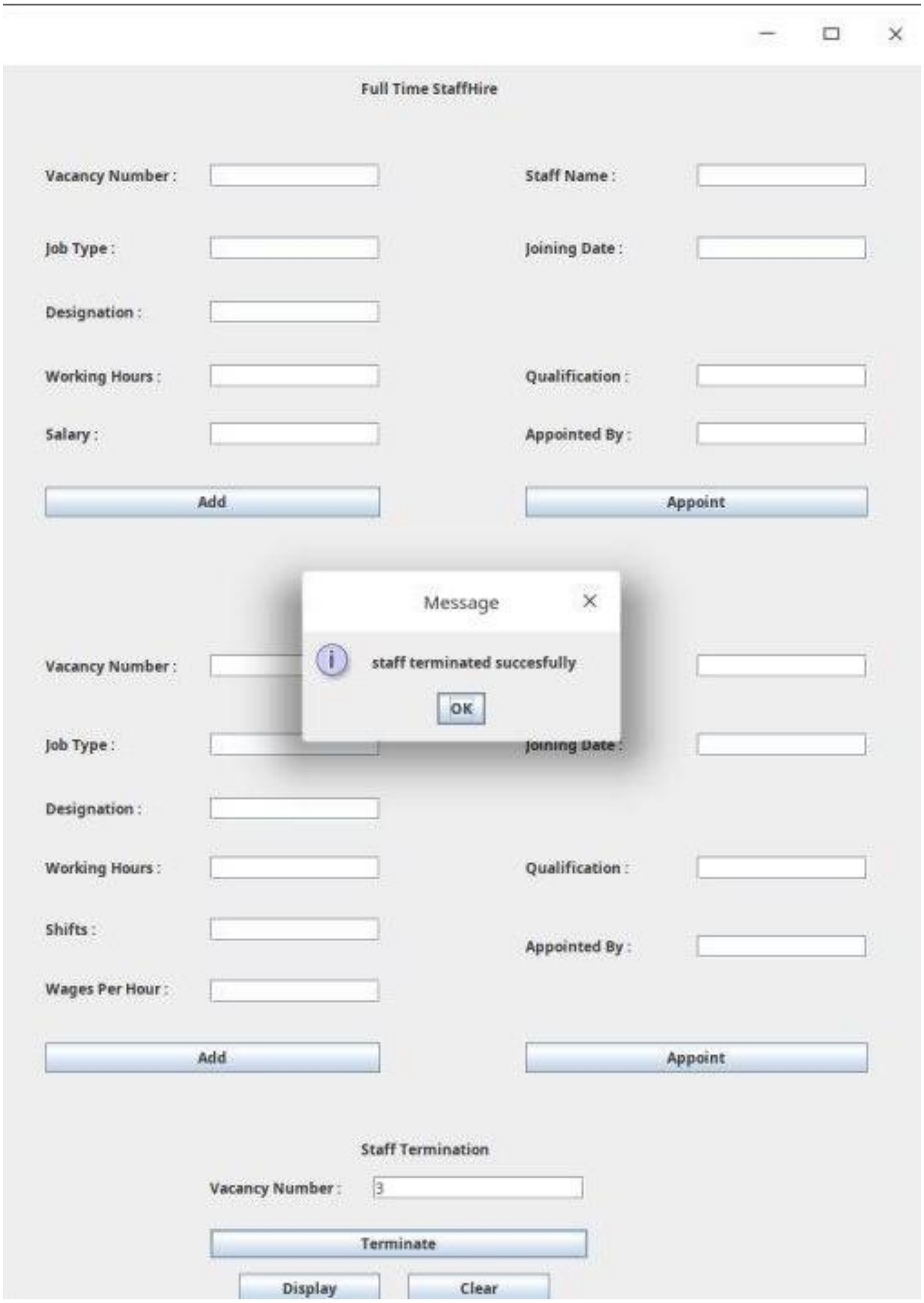
Qualification : bachelors

Shifts : day

Appointed By : samir mishra

Wages Per Hour : 300

Figure 4: Test 2



staff terminated succesfully

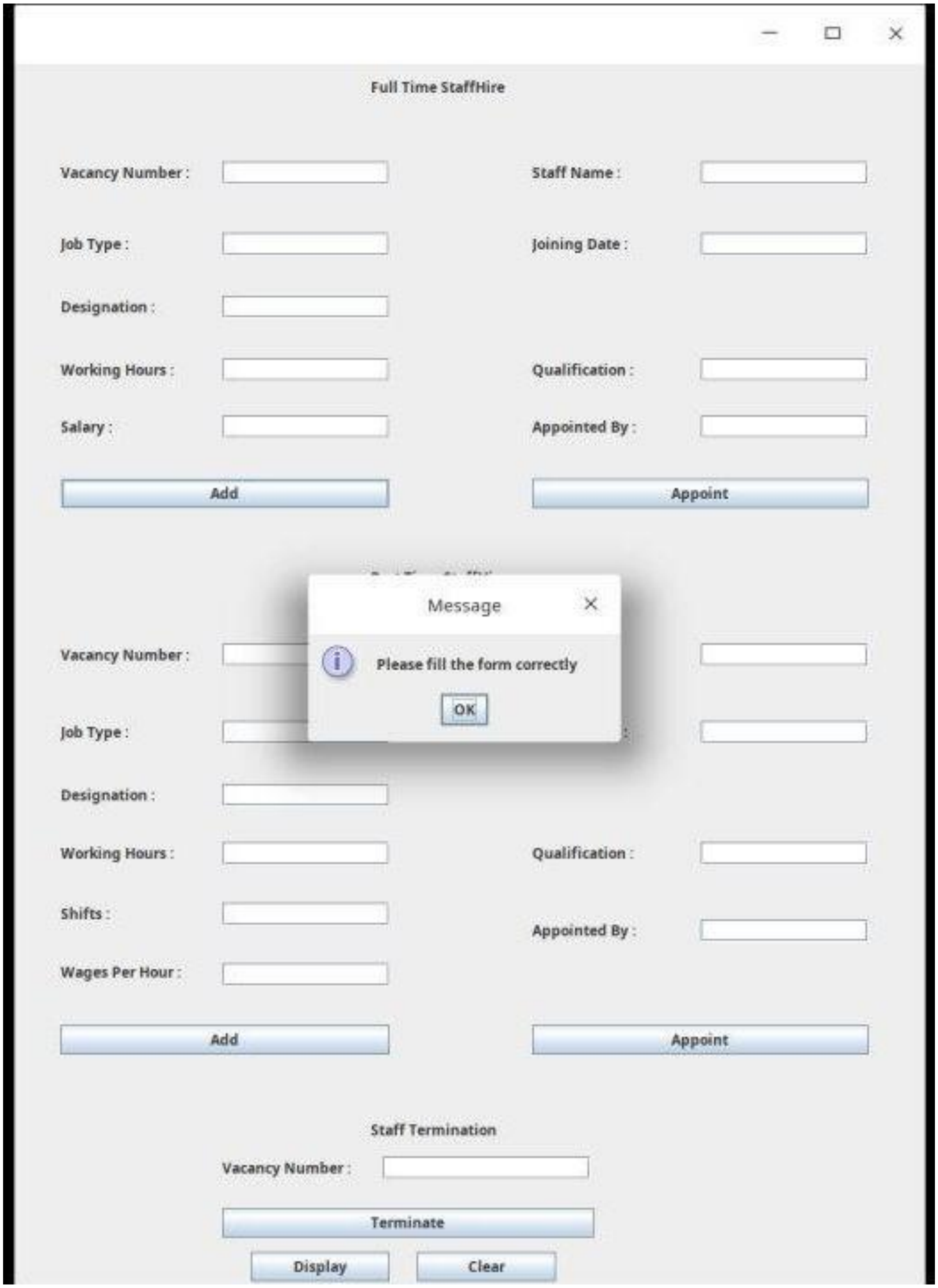
Vacancy Number : 3

Figure 5: Test 2

Test 3:

Table 3 : testing errors

Objective	To check errors.
Action	Hiring staff without filling form.
Excepted result	An error message should be displayed.
Actual result	Error message was displayed.
Conclusion	Test successful.

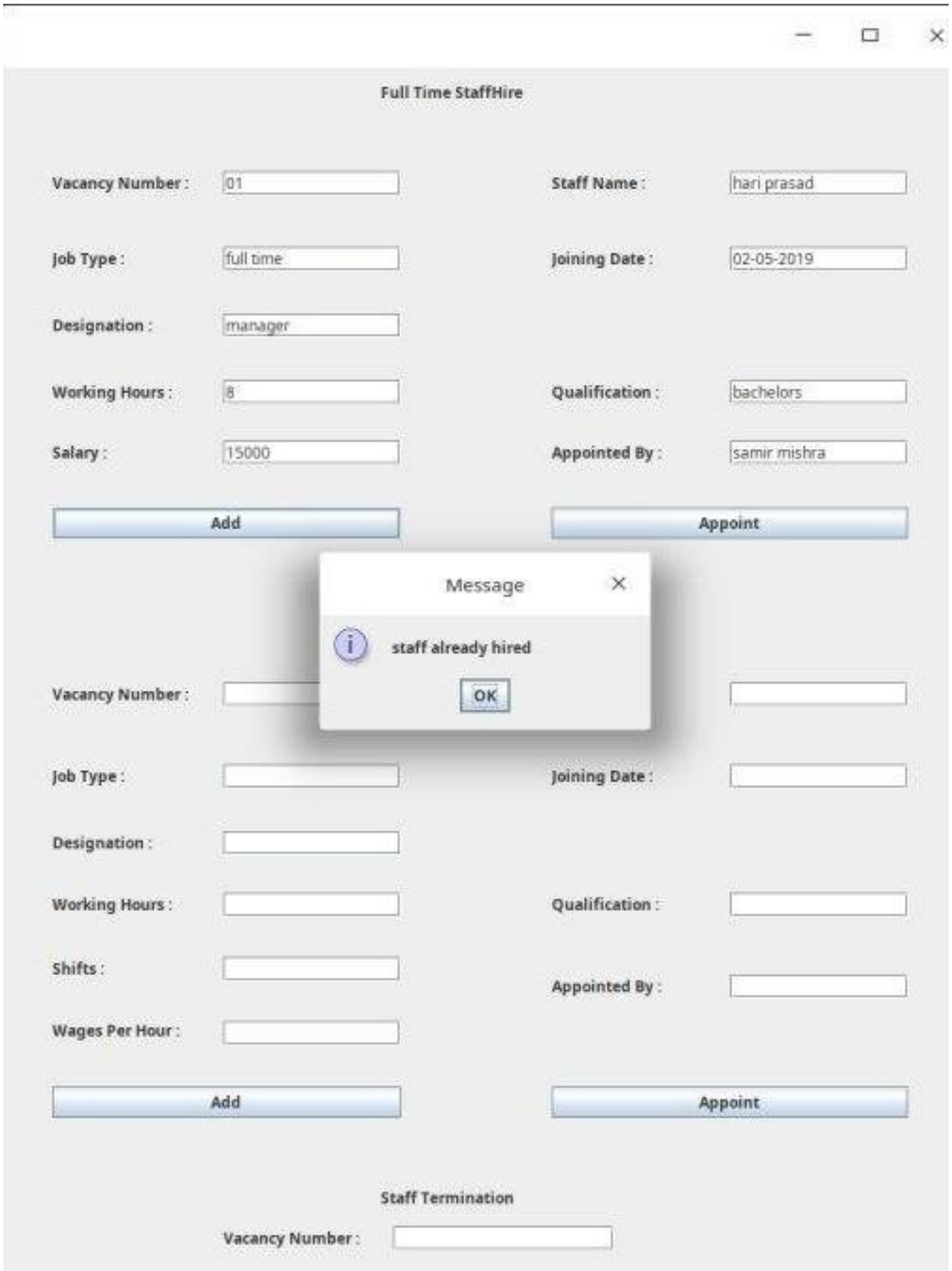


Please fill the form correctly

Figure 6: Test 3

Case 1: Empty form

Case 2: Hiring same staff


Full Time StaffHire

Vacancy Number : 01

Staff Name : hari prasad

Job Type : full time

Joining Date : 02-05-2019

Designation : manager

Working Hours : 8

Qualification : bachelors

Salary : 15000

Appointed By : samir mishra

Message

staff already hired

Vacancy Number :

Job Type :

Joining Date :

Designation :

Working Hours :

Qualification :

Shifts :

Appointed By :

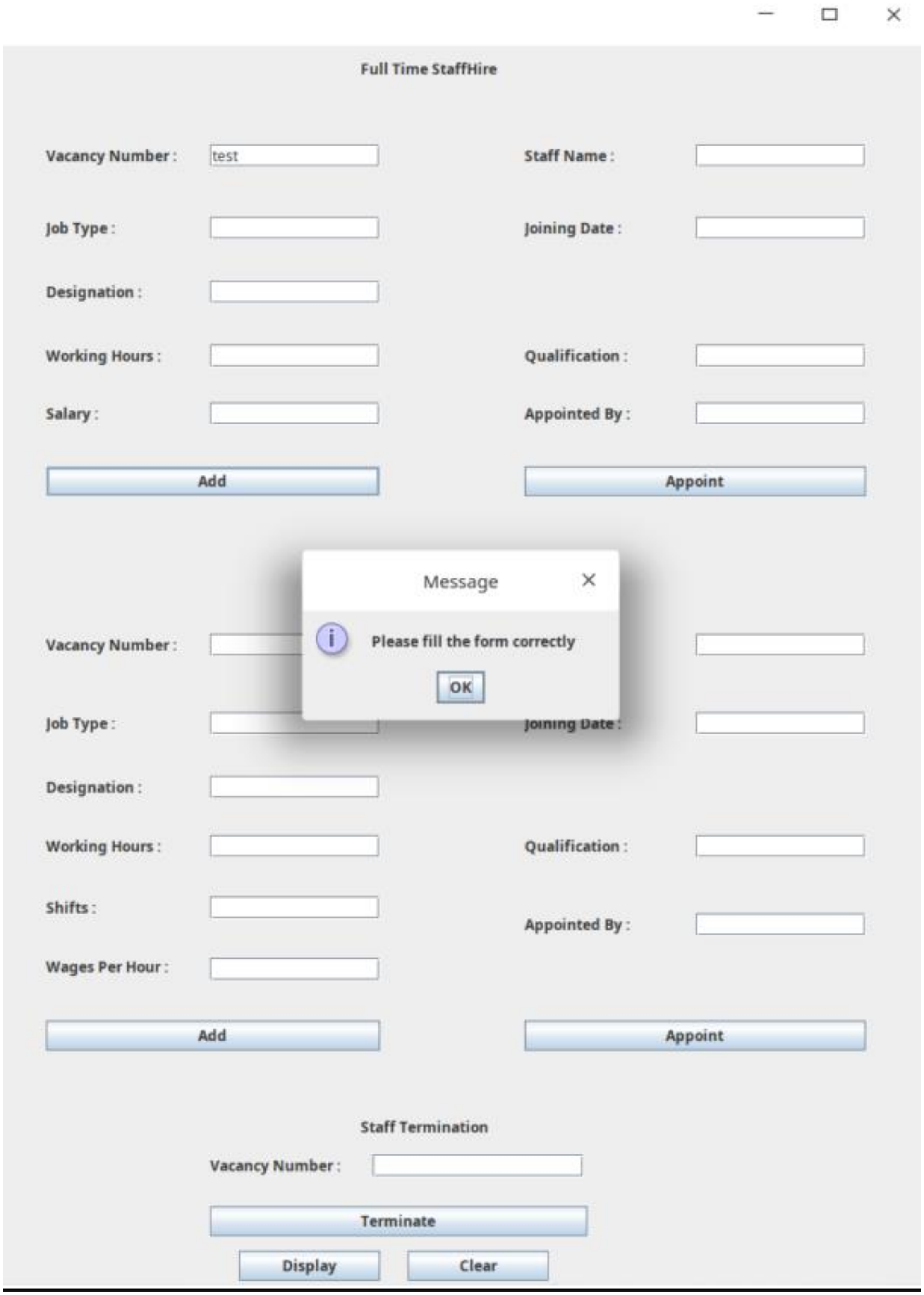
Wages Per Hour :

Staff Termination

Vacancy Number :

Figure 7: Test 3

Case 3: Invalid Characters



Full Time StaffHire

Vacancy Number : test

Staff Name :

Job Type :

Joining Date :

Designation :

Working Hours :

Qualification :

Salary :

Appointed By :

Message

Please fill the form correctly

Vacancy Number :

Job Type :

Joining Date :

Designation :

Working Hours :

Qualification :

Shifts :

Appointed By :

Wages Per Hour :

Staff Termination

Vacancy Number :

Figure 8: Test 3

Conclusion

Doing this coursework was very hard since there was no class which made very hard to complete this coursework. I was unable to get any help for this coursework; online class was like going to college where there is no teachers and students but anyhow, I was able to complete this coursework. In this coursework I learned about java swing package and get familiar with the GUI of java which was very fun to do. During coursework I faced many errors and bugs, making GUI was another challenge for me to set bounds values and to make GUI look nice was one of the most challenging work for me and the most challenging work for me in this coursework was to code for event handling and make that GUI functional.

Apendix

```
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JTextField;
import javax.swing.JLabel;
import javax.swing.JButton;
import javax.swing.JOptionPane;
import java.awt.event.ActionListener;
import java.util.ArrayList;
import java.awt.event.ActionEvent;

public class INGNepal {

	private JFrame frame;
	private JTextField textField;
	private JTextField textField_1;
	private JTextField textField_2;
	private JTextField textField_3;
	private JTextField textField_4;
	private JTextField textField_5;
	private JTextField textField_6;
	private JTextField textField_8;
	private JTextField textField_9;
	private JTextField textField_10;
	private JTextField textField_11;
	private JTextField textField_12;
	private JTextField textField_13;
	private JTextField textField_14;
	private JTextField textField_15;
	private JTextField textField_16;
private JTextField textField_17;
private JTextField textField_19;
private JTextField textField_20;
private JTextField textField_21;
FullTimeStaffHire fullTimeHire;
PartTimeStaffHire partTimeHire;
private ArrayList<StaffHire> arrList = new ArrayList<>();

/**
 * Launch the application.
 */
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                INGNepal window = new INGNepal();
                window.frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

/**
 * Create the application.
 */
public INGNepal() {
    initialize();
    }

    public void one() {
        JLabel lblFullTimeStaffhire = new JLabel("Full Time StaffHire");
        lblFullTimeStaffhire.setBounds(300, 12, 129, 15);
        frame.getContentPane().add(lblFullTimeStaffhire);

        JLabel lblVacancyNumber = new JLabel("Vacancy Number :");
        lblVacancyNumber.setBounds(39, 79, 141, 24);
        frame.getContentPane().add(lblVacancyNumber);

        textField = new JTextField();
        textField.setBounds(175, 82, 141, 19);
        frame.getContentPane().add(textField);
        textField.setColumns(10);

        JLabel lblJobTye = new JLabel("Job Type :");
        lblJobTye.setBounds(39, 144, 70, 15);
        frame.getContentPane().add(lblJobTye);

        textField_1 = new JTextField();
        textField_1.setColumns(10);
        textField_1.setBounds(175, 142, 141, 19);
        frame.getContentPane().add(textField_1);

        JLabel lblDesignation = new JLabel("Designation :");
        lblDesignation.setBounds(39, 197, 109, 15);
        frame.getContentPane().add(lblDesignation);
        textField_2 = new JTextField();
        textField_2.setColumns(10);
        textField_2.setBounds(175, 195, 141, 19);
        frame.getContentPane().add(textField_2);

        JLabel lblWorkingHours = new JLabel("Working Hours :");
        lblWorkingHours.setBounds(39, 250, 129, 15);
        frame.getContentPane().add(lblWorkingHours);

        textField_3 = new JTextField();
        textField_3.setColumns(10);
        textField_3.setBounds(175, 248, 141, 19);
        frame.getContentPane().add(textField_3);

        JLabel lblSalary = new JLabel("Salary :");
        lblSalary.setBounds(39, 298, 70, 15);
        frame.getContentPane().add(lblSalary);

        textField_4 = new JTextField();
        textField_4.setColumns(10);
        textField_4.setBounds(175, 296, 141, 19);
        frame.getContentPane().add(textField_4);

        JButton btnAdd = new JButton("Add"); //up add
        btnAdd.setBounds(39, 349, 277, 25);
        frame.getContentPane().add(btnAdd);
        btnAdd.addActionListener(new ActionListener() {

    @Override
```

```
   public void actionPerformed(ActionEvent arg) {
      int vacancy = 0,workingHour = 0,income = 0;
      String vacNo = textField.getText();
      String staffName = textField_5.getText();
      String designation = textField_2.getText();
      String jobType = textField_1.getText();
      String salary = textField_4.getText();
      String joinedDate = textField_6.getText();
      String workHours = textField_3.getText();
      String appointedBy = textField_9.getText();
      String qualification = textField_8.getText();


if(vacNo.trim().equals("")||staffName.trim().equals("")||designation.trim().equals("")||jobType.trim().equals("")||salary.trim().equals("")||joinedDate.trim().equals("")||workHours.trim().equals("")||appointedBy.trim().equals("")||qualification.trim().equals("")) {
         JOptionPane.showMessageDialog(frame,"Please fill the form correctly");
      }else {
         try {
            vacancy = Integer.parseInt(vacNo);
            workingHour = Integer.parseInt(workHours);
            income = Integer.parseInt(salary);
         }catch(NumberFormatException nfe) {
            JOptionPane.showMessageDialog(frame,"Please fill the form correctly");
            return;
         }
         for(StaffHire s : arrList) {
            if(s instanceof FullTimeStaffHire) {
               FullTimeStaffHire full = (FullTimeStaffHire) s;
               if(full.getVacancyNumber() == vacancy) {
```

```
                                JOptionPane.showMessageDialog(frame,"staff already hired");
                            }
                        }
                    }
                }
                fullTimeHire = new FullTimeStaffHire(vacancy, designation, jobType, income, workingHour);
                arrList.add(fullTimeHire);
            }
        });

                    JLabel lblStaffName = new JLabel("Staff Name :");
                    lblStaffName.setBounds(436, 84, 96, 15);
                    frame.getContentPane().add(lblStaffName);

                    textField_5 = new JTextField();
                    textField_5.setColumns(10);
                    textField_5.setBounds(578, 82, 141, 19);
                    frame.getContentPane().add(textField_5);

                    JLabel lblJoiningDate = new JLabel("Joining Date :");
                    lblJoiningDate.setBounds(436, 144, 96, 15);
                    frame.getContentPane().add(lblJoiningDate);

                    textField_6 = new JTextField();
                    textField_6.setColumns(10);
                    textField_6.setBounds(578, 142, 141, 19);
                    frame.getContentPane().add(textField_6);
                    JLabel lblQualification = new JLabel("Qualification :");
                    lblQualification.setBounds(436, 250, 109, 15);
                    frame.getContentPane().add(lblQualification);

                    textField_8 = new JTextField();
                    textField_8.setColumns(10);
                    textField_8.setBounds(578, 248, 141, 19);
                    frame.getContentPane().add(textField_8);

                    JLabel lblAppointedBy = new JLabel("Appointed By :");
                    lblAppointedBy.setBounds(436, 298, 129, 15);
                    frame.getContentPane().add(lblAppointedBy);

                    textField_9 = new JTextField();
                    textField_9.setColumns(10);
                    textField_9.setBounds(578, 296, 141, 19);
                    frame.getContentPane().add(textField_9);

                    JButton btnAppoint = new JButton("Appoint"); //up appoint
                    btnAppoint.setBounds(436, 349, 283, 25);
                    frame.getContentPane().add(btnAppoint);
                    btnAppoint.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent arg) {
                int vacancy = 0;
                String vacNo = textField.getText();
                String staffName = textField_5.getText();
                String designation = textField_2.getText();
```

```
        String jobType = textField_1.getText();
        String salary = textField_4.getText();
        String joinedDate = textField_6.getText();
        String workHours = textField_3.getText();
        String appointedBy = textField_9.getText();
        String qualification = textField_8.getText();


if(vacNo.trim().equals("")||staffName.trim().equals("")||designation.trim().equals("")||jobType.trim().equals("")||salary.trim().equals("")||joinedDate.trim().equals("")||workHours.trim().equals("")||appointedBy.trim().equals("")||qualification.trim().equals("")) {
            JOptionPane.showMessageDialog(frame,"Please fill the form correctly");
        }else {
            try {
                vacancy = Integer.parseInt(vacNo);
            }catch(NumberFormatException nfe) {
                JOptionPane.showMessageDialog(frame,"Please fill the form correctly");
                return;
            }
        }
        for(int i=0;i<arrList.size();i++) {
            StaffHire s = arrList.get(i);
            if(s instanceof FullTimeStaffHire) {
                FullTimeStaffHire full = (FullTimeStaffHire)s;
                if(full.getVacancyNumber()==vacancy) {
                    full.hireFullTimeStaff(staffName, joinedDate, qualification, appointedBy);
                    JOptionPane.showMessageDialog(frame,"Staff hired successfully");
                    break;
                }else {
                        JOptionPane.showMessageDialog(frame,"vacancy number not
found");
                    }
                }
            }
        }
    });
      }

      public void two() {
                JLabel lblPartlTimeStaffhire = new JLabel("Part Time StaffHire");
                lblPartlTimeStaffhire.setBounds(300, 425, 134, 15);
                frame.getContentPane().add(lblPartlTimeStaffhire);

                JLabel lblVacancyNumber_2 = new JLabel("Vacancy Number :");
                lblVacancyNumber_2.setBounds(39, 485, 141, 24);
                frame.getContentPane().add(lblVacancyNumber_2);

                textField_10 = new JTextField();
                textField_10.setColumns(10);
                textField_10.setBounds(175, 488, 141, 19);
                frame.getContentPane().add(textField_10);

                JLabel lblJobTye_1 = new JLabel("Job Type :");
                lblJobTye_1.setBounds(39, 555, 70, 15);
                frame.getContentPane().add(lblJobTye_1);

                textField_11 = new JTextField();
                textField_11.setColumns(10);
```

```
textField_11.setBounds(175, 553, 141, 19);
frame.getContentPane().add(textField_11);

JLabel lblDesignation_1 = new JLabel("Designation :");
lblDesignation_1.setBounds(39, 608, 109, 15);
frame.getContentPane().add(lblDesignation_1);

textField_12 = new JTextField();
textField_12.setColumns(10);
textField_12.setBounds(175, 606, 141, 19);
frame.getContentPane().add(textField_12);

JLabel lblWorkingHours_1 = new JLabel("Working Hours :");
lblWorkingHours_1.setBounds(39, 652, 129, 24);
frame.getContentPane().add(lblWorkingHours_1);

textField_13 = new JTextField();
textField_13.setColumns(10);
textField_13.setBounds(175, 655, 141, 19);
frame.getContentPane().add(textField_13);

JLabel lblShifts = new JLabel("Shifts :");
lblShifts.setBounds(39, 708, 70, 15);
frame.getContentPane().add(lblShifts);

textField_14 = new JTextField();
textField_14.setColumns(10);
textField_14.setBounds(175, 706, 141, 19);
frame.getContentPane().add(textField_14);
```

```
JLabel lblWorkingHours_1_1 = new JLabel("Wages Per Hour :");
lblWorkingHours_1_1.setBounds(39, 752, 129, 24);
frame.getContentPane().add(lblWorkingHours_1_1);

textField_15 = new JTextField();
textField_15.setColumns(10);
textField_15.setBounds(175, 757, 141, 19);
frame.getContentPane().add(textField_15);

JLabel lblStaffName_1 = new JLabel("Staff Name :");
lblStaffName_1.setBounds(436, 490, 96, 15);
frame.getContentPane().add(lblStaffName_1);

textField_16 = new JTextField();
textField_16.setColumns(10);
textField_16.setBounds(578, 488, 141, 19);
frame.getContentPane().add(textField_16);

JLabel lblJoiningDate_1 = new JLabel("Joining Date :");
lblJoiningDate_1.setBounds(436, 555, 96, 15);
frame.getContentPane().add(lblJoiningDate_1);

textField_17 = new JTextField();
textField_17.setColumns(10);
textField_17.setBounds(578, 553, 141, 19);
frame.getContentPane().add(textField_17);

JLabel lblQualification_1 = new JLabel("Qualification :");
```

```
		lblQualification_1.setBounds(436, 657, 109, 15);
		frame.getContentPane().add(lblQualification_1);

		textField_19 = new JTextField();
		textField_19.setColumns(10);
		textField_19.setBounds(578, 655, 141, 19);
		frame.getContentPane().add(textField_19);

		JLabel lblAppointedBy_1 = new JLabel("Appointed By :");
		lblAppointedBy_1.setBounds(436, 722, 129, 15);
		frame.getContentPane().add(lblAppointedBy_1);

		textField_20 = new JTextField();
		textField_20.setColumns(10);
		textField_20.setBounds(578, 720, 141, 19);
		frame.getContentPane().add(textField_20);

		JButton btnAdd_1 = new JButton("Add"); //down add
		btnAdd_1.setBounds(39, 809, 277, 25);
		frame.getContentPane().add(btnAdd_1);
		btnAdd_1.addActionListener(new ActionListener() {

	@Override
	public void actionPerformed(ActionEvent arg0) {
	   int vacancy,workingHours,wageshour;
	   String vacNo =textField_10.getText();
	   String designation =textField_12.getText();
	   String job =textField_11.getText();
	   String appointed =textField_20.getText();
```

```
                String shift =textField_14.getText();
                String wages =textField_15.getText();
                String name =textField_16.getText();
                String date =textField_17.getText();
                String work =textField_13.getText();
                String qualification =textField_19.getText();

                if(vacNo.trim().equals("") || designation.trim().equals("") || job.trim().equals("")
|| appointed.trim().equals("") || shift.trim().equals("") || wages.trim().equals("") ||
name.trim().equals("") || date.trim().equals("") || work.trim().equals("") ||
qualification.trim().equals("")) {
                    JOptionPane.showMessageDialog(frame,"Please fill the form correctly");
                }else {
                    try {
                        vacancy = Integer.parseInt(vacNo);
                        workingHours = Integer.parseInt(work);
                        wageshour = Integer.parseInt(wages);
                    }catch(NumberFormatException arg) {
                        JOptionPane.showMessageDialog(frame,"Please fill the form correctly");
                        return;
                    }
                    for(StaffHire s : arrList) {
                        if(s instanceof PartTimeStaffHire) {
                            PartTimeStaffHire part = (PartTimeStaffHire) s;
                            if(part.getVacancyNumber() == vacancy) {
                                JOptionPane.showMessageDialog(frame,"staff already hired");
                            }
                        }
                    }
                    partTimeHire = new PartTimeStaffHire(vacancy, designation, job,
workingHours, wageshour, shift);
                    arrList.add(partTimeHire);
                }
            }
        });

                      JButton btnAdd_1_1 = new JButton("Appoint"); //downn appoint
                      btnAdd_1_1.setBounds(436, 809, 283, 25);
                      frame.getContentPane().add(btnAdd_1_1);
                      btnAdd_1_1.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent arg0) {
                int vacancy=0;
                String vacNo =textField_10.getText();
                String designation =textField_12.getText();
                String job =textField_11.getText();
                String appointed =textField_20.getText();
                String shift =textField_14.getText();
                String wages =textField_15.getText();
                String name =textField_16.getText();
                String date =textField_17.getText();
                String work =textField_13.getText();
                String qualification =textField_19.getText();

                if(vacNo.trim().equals("") || designation.trim().equals("") || job.trim().equals("")
|| appointed.trim().equals("") || shift.trim().equals("") || wages.trim().equals("") ||
name.trim().equals("") || date.trim().equals("") || work.trim().equals("") ||
qualification.trim().equals("")) {
```

```
                JOptionPane.showMessageDialog(frame,"Please fill the form correctly");
            }else {
                try {
                    vacancy = Integer.parseInt(vacNo);
                }catch(NumberFormatException arg) {
                    JOptionPane.showMessageDialog(frame,"Please fill the form correctly");
                    return;
                }
            }
            for(int i=0;i<arrList.size();i++) {
                StaffHire s = arrList.get(i);
                if(s instanceof PartTimeStaffHire) {
                    PartTimeStaffHire part = (PartTimeStaffHire)s;
                    if(part.getVacancyNumber()==vacancy) {
                        part.hirePartTimeStaff(name, date, qualification, appointed);
                        JOptionPane.showMessageDialog(frame,"Staff hired successfully");
                        break;
                    }else {
                        JOptionPane.showMessageDialog(frame,"vacancy number not
found");
                    }
                }
            }


        }
    });
   }


   public void three() {
        JLabel lblStaffTermination = new JLabel("Staff Termination");
        lblStaffTermination.setBounds(300, 890, 134, 15);
        frame.getContentPane().add(lblStaffTermination);

        JLabel lblVacancyNumber_1_1_1 = new JLabel("Vacancy Number :");
        lblVacancyNumber_1_1_1.setBounds(175, 917, 141, 24);
        frame.getContentPane().add(lblVacancyNumber_1_1_1);

        textField_21 = new JTextField();
        textField_21.setColumns(10);
        textField_21.setBounds(310, 920, 175, 19);
        frame.getContentPane().add(textField_21);

        JButton btnAdd_1_1_1 = new JButton("Terminate");
        btnAdd_1_1_1.setBounds(175, 963, 313, 25);
        frame.getContentPane().add(btnAdd_1_1_1);
        btnAdd_1_1_1.addActionListener(new ActionListener() {

@Override
public void actionPerformed(ActionEvent arg0) {
    String vacNo = textField_21.getText();
    int vacancy = 0;
    if(vacNo.trim().equals("")) {
        JOptionPane.showMessageDialog(frame,"Please fill the form correctly");
    }else {
        try {
            vacancy = Integer.parseInt(vacNo);
        }catch(NumberFormatException nfe) {
```

```
                        JOptionPane.showMessageDialog(frame,"Please enter correct vacancy
number");
                        return;
                    }
                }
                boolean found = false;
                for(StaffHire staff : arrList) {
                    if(staff instanceof PartTimeStaffHire) {
                        PartTimeStaffHire part = (PartTimeStaffHire) staff;
                        if(staff.getVacancyNumber() == vacancy) {
                            part.terminateStaff();
                            arrList.remove(staff);
                            JOptionPane.showMessageDialog(frame,"staff terminated
succesfully");
                            found = true;
                        }
                    }
                }
                if(!found) {
                    JOptionPane.showMessageDialog(frame,"vacancy number is invalid");
                }


            }
        });


                      JButton btnDisplay = new JButton("Display");
                      btnDisplay.setBounds(199, 1000, 117, 25);
                      frame.getContentPane().add(btnDisplay);
                      btnDisplay.addActionListener(new ActionListener() {
```

```
        @Override
        public void actionPerformed(ActionEvent arg) {
            for(StaffHire staff : arrList) {
                if(staff instanceof FullTimeStaffHire) {
                    FullTimeStaffHire full = (FullTimeStaffHire)staff;
                    staff.display();
                }else {
                    PartTimeStaffHire part = (PartTimeStaffHire)staff;
                    staff.display();
                }
            }
        }
    });

                JButton btnClear = new JButton("Clear");
                btnClear.setBounds(339, 1000, 117, 25);
                frame.getContentPane().add(btnClear);
                btnClear.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent arg0) {
            textField.setText("");
                                  textField_1.setText("");
                                  textField_2.setText("");
                                  textField_3.setText("");
                                  textField_4.setText("");
                                  textField_5.setText("");
                                  textField_6.setText("");
                                  textField_8.setText("");
                                  textField_9.setText("");
                                  textField_10.setText("");
                                  textField_11.setText("");
                                  textField_12.setText("");
                                  textField_13.setText("");
                                  textField_14.setText("");
                                  textField_15.setText("");
                                  textField_16.setText("");
                                  textField_17.setText("");
                                  textField_19.setText("");
                                  textField_20.setText("");
                                  textField_21.setText("");
        }
    });
     }

     /**
        * Initialize the contents of the frame.
        */
     private void initialize() {
                frame = new JFrame();
                frame.setBounds(100, 100, 769, 1116);
                frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                frame.getContentPane().setLayout(null);

                one();
                two();
                three();
     }
 }
```

Bibliography

Oracle. (2018). *Package javax.swing*. Retrieved from oracle: https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html

Oracle. (2020). *Java Documentation*. Retrieved from Oracle: https://docs.oracle.com/en/java/